



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Análisis y caracterización del tráfico
HTC+MTC en una Smart City
Modelado de fuentes y Calidad de
Servicio

TESIS DOCTORAL

Autor

Ángel Gómez Sacristán

Director

Miguel Ángel Rodríguez Hernández

Valencia, España
Julio 2018

Resumen

Las ciudades constituyen un poderoso motor de crecimiento económico y social gracias a las oportunidades que ofrecen en contraposición al medio rural: mayor diversidad y calidad de empleo junto con mejores infraestructuras y servicios. Sin embargo, este crecimiento genera una demanda creciente de recursos por lo que su gestión sostenible se ha convertido en un elemento clave en la política de las administraciones en todo el mundo.

El concepto de “Ciudad Inteligente” surge para dar una respuesta global para el entorno urbano. Los principales organismos de estandarización están desarrollando un ecosistema de recomendaciones que definen entre otros aspectos, y relacionados con el ámbito tecnológico de la presente tesis, una arquitectura tecnológica consistente, métricas e indicadores de rendimiento, así como un modelo integrado de información.

En este complejo entorno, la red de comunicaciones de una Ciudad Inteligente ha de ser capaz de adaptarse a los incrementos, en muchos casos abruptos, del tráfico generado por miles de dispositivos de naturaleza, criticidad y comportamiento heterogéneo. Garantizar la calidad, disponibilidad y prestaciones de los servicios subyacentes cuando éstos compiten por recursos limitados, así como evaluar el impacto de la introducción de nuevos servicios “smart”, es un área de creciente interés.

Aspectos como la cobertura geográfica, el bajo consumo y coste de los dispositivos, junto con la aparición de nuevas tecnologías y novedosos patrones de tráfico no suficientemente caracterizados, generan nuevos retos que es necesario abordar. Sirva como ejemplo el lanzamiento en España de los primeros servicios NB-IoT (3GPP TR 45.820 Cellular system support for ultra-low complexity and low throughput Internet of Things) y el reto que supondrá el despliegue de las tecnologías 5G en entornos urbanos.

Esta tesis, que desarrolla el objetivo 05 del Proyecto de Investigación “MEC TIN2013-47272-C2-1-R - Plataforma de Servicios para Ciudades Inteligentes con Redes M2M Densas (PLASMA). Caracterización de servicios MTC/IoT y modelado de patrones de tráfico”, profundiza en los aspectos relacionados con la evaluación de prestaciones de diversos servicios bajo una arquitectura de referencia. Para ello se han caracterizado las diferentes fuentes de tráfico que intervienen en una ciudad inteligente y se ha definido una red modelo convergente de acuerdo con las recomendaciones ITU-T Y.2011/Y.2012/Q3.900 (Next Generation Networks – Frameworks and functional architecture models & testing).

En base a las conclusiones anteriores, se ha diseñado una herramienta denominada “SimulCity” que permite simular de manera flexible escenarios donde múltiples fuentes configurables de voz, vídeo y MTC comparten recursos en la red modelo definida. SimulCity permite la configuración de los mecanismos de gestión de tráfico (control de admisión en el acceso, clases de servicio, funciones policía, tránsito en la red MPLS) para evaluar prestaciones y dimensionar adecuadamente los parámetros de caudales y clases de servicio que garanticen el adecuado funcionamiento de las aplicaciones.

La herramienta Simulcity se ha utilizado para simular diferentes escenarios de servicios que están en proceso de implantación en la ciudad de València. Los resultados obtenidos han permitido realizar un dimensionado adecuado de los recursos de ancho de banda y clases de servicio necesarios con anterioridad a su despliegue.

Resum

Les ciutats constitueixen un poderós motor de creixement econòmic i social gràcies a les oportunitats que oferixen en contraposició al medi rural: major diversitat i qualitat d'ocupació junt amb millors infraestructures i servicis. No obstant això, este creixement genera una demanda creixent de recursos i la seua gestió sostenible s'ha convertit en un element clau en la política de les administracions en tot el món.

El concepte de “Ciutat Intel·ligent” sorgix per a donar una resposta global per a l'entorn urbà. Els principals organismes d'estandardització estan desenrotllant un ecosistema de recomanacions que definixen entre altres aspectes, i relacionats amb l'àmbit tecnològic de la present tesi, una arquitectura tecnològica consistent, mètriques i indicadors de rendiment, així com un model integrat d'informació.

En este complex entorn, la xarxa de comunicacions de una Ciutat Intel·ligent ha de ser capaç d'adaptar-se als increments, en molts casos abruptes, del tràfic generat per milers de dispositius de naturalesa, criticidad i comportament heterogeni. Garantir la qualitat, disponibilitat i prestacions dels servicis subjacents quan estos competixen per recursos limitats, així com avaluar l'impacte de la introducció de nous servicis “smart”, és una àrea de creixent interès.

Aspectes com la cobertura geogràfica, el baix consum i cost dels dispositius, junt amb l'aparició de noves tecnologies i nous patrons de tràfic no suficientment caracteritzats, generen nous reptes que és necessari abordar. Servisca com a exemple el llançament a Espanya dels primers servicis NB-IoT (3GPP TR 45.820 Cellular system support for ultra-low complexity and low throughput Internet of Things) i el repte que suposarà el desplaçament de les tecnologies 5G en entorns urbanes.

Esta tesi, que desenrotlla l'objectiu 05 del Projecte d'Investigació “MEC TIN2013-47272-C2-1-R - Plataforma de Servicis per a Ciutats Intel·ligents amb Xarxes M2M Denses (PLASMA) . Caracterització de servicis MTC/IoT i modelatge de patrons de tráfico”, aprofundix en els aspectes relacionats amb l'avaluació de prestacions de diversos servicis davall una arquitectura de referència. Per a això s'han caracteritzat les diferents fonts de tràfic que intervenen en una ciutat intel·ligent i s'ha definit una xarxa model convergent d'acord amb les recomanacions ITU-T Y.2011/Y.2012/Q3.900 (Next Generation Networks - Frameworks and functional architecture models & testing).

Basant-se en les conclusions anteriors, s'ha dissenyat una ferramenta denominada “SimulCity”; que permet simular de manera flexible escenaris on múltiples fonts configurables de veu, vídeo i MTC compartixen recursos en la xarxa model definida. SimulCity permet la configuració dels mecanismes de gestió de tràfic (control d'admissió en l'accés, classes de servici, funcions policia, trànsit en la xarxa MPLS) per a avaluar prestacions i dimensionar adequadament els paràmetres de cabals i classes de servici que garantisquen l'adequat funcionament de les aplicacions.

La ferramenta Simulcity s'ha utilitzat per a simular diferents escenaris de servicis que estan en procés d'implantació en la ciutat de València. Els resultats obtinguts han permès realitzar un dimensionat adequat dels recursos d'amplada de banda i classes de servici necessaris amb anterioritat al seu desplaçament.

Abstract

Cities are a powerful engine for economic and social growth thanks to the opportunities offered in contrast to the rural environment: greater diversity and quality of employment together with better infrastructure and services. However, this growth generates an increasing demand for resources, so that its sustainable management has become a key element in the policy of administrations around the world.

The concept of "Smart City" emerges to give a global response to the urban environment. The main standardization bodies are developing an ecosystem of recommendations that define, among other aspects, and related to the technological scope of this thesis, a consistent technological architecture, metrics and performance indicators, as well as an integrated model of information.

In this complex environment, the telecommunications network of a Smart City must be able to adapt to the often-abrupt increases in traffic generated by thousands of devices of a nature, criticality and heterogeneous behaviour. Ensuring the quality, availability and performance of underlying services when they compete for limited resources, as well as assessing the impact of the introduction of new smart services, is an area of growing interest.

Aspects such as geographical coverage, low consumption and cost of the devices, together with the appearance of new technologies and novel traffic patterns not sufficiently characterized, generate new challenges that need to be addressed. An example is the launch in Spain of the first NB-IoT services (3GPP TR 45.820 Cellular system support for ultra-low complexity and low throughput Internet of Things) and the challenge that will involve the deployment of 5G technologies in urban environments.

This thesis, which develops objective 05 of the Research Project "MEC TIN2013-47272-C2-1-R - Services Platform for Smart Cities with Dense M2M Networks (PLASMA). Characterization of MTC / IoT services and modelling of traffic patterns ", delves into the aspects related to the evaluation of the performance of various services under a reference architecture. For this, the different sources of traffic that intervene in an intelligent city have been characterized and a convergent model network has been defined in accordance with ITU-T recommendations Y.2011 / Y.2012 / Q3.900 (Next Generation Networks - Frameworks and functional architecture models & testing).

Based on the above conclusions, a tool called "SimulCity" has been designed that allows to simulate in a flexible way scenario where multiple configurable sources of voice, video and MTC share resources in the defined model network. SimulCity allows the configuration of the traffic management mechanisms (access admission control, service classes, police functions, traffic in the MPLS network) to evaluate performance and adequately dimension the parameters of flow rates and service classes that guarantee the adequate operation of the applications.

The Simulcity tool has been used to simulate different service scenarios that are in the process of being implemented in the city of Valencia. The results obtained have allowed an adequate sizing of the bandwidth resources and service classes necessary prior to their deployment.

Agradecimientos

A todos los que con su generosidad construyen un mundo mejor cada día.

A Margarita, mi compañera y amiga, porque tu sonrisa ilumina el cielo cada mañana.

A mi hijo David, por ser como es.

A mi familia y amigos, porque siempre estáis cerca.

A Miguel, mi Director de Tesis, sin tu ayuda esto no hubiera sido posible.

Y a todas las personas de las que he aprendido, que han sido muchas.

Índice

1	Introducción	1
1.1	Introducción	1
1.2	Motivación de la Tesis	3
1.3	Objetivos de la Tesis	4
1.4	Principales aportaciones.....	6
1.4.1	Artículos.....	6
1.4.2	Congresos.....	7
1.4.3	Participación en proyectos de investigación.....	9
1.4.4	Software	9
1.4.5	Participación en proyecto Valencia Ciudad Inteligente	11
1.5	Organización de la memoria	11
2	Ciudades Inteligentes	13
2.1	Introducción	13
2.2	Smart Cities o Ciudades Inteligentes	13
2.2.1	Concepto de Ciudad Inteligente	14
2.2.2	Atributos de las Ciudades Inteligentes	15
2.2.3	Requisitos de las Ciudades Inteligentes	20
2.2.4	Semántica de la Ciudad Inteligente	22
2.3	Arquitectura TIC en una Ciudad Inteligente	26
2.3.1	Capa de captación.....	33
2.3.2	Capa de Red (comunicaciones)	52
2.3.3	Capa de Datos y Soporte	58
2.3.4	Capa de Servicios, aplicaciones y contenidos	59
2.4	Plataformas de Ciudad	60
2.4.1	Requisitos funcionales de una Plataforma de Ciudad Inteligente	62
2.4.2	Modelo de capas de la Plataforma SSC.....	66
2.4.3	Plataforma de Ciudad del Ayuntamiento de València (VLCi).....	72
3	Fuentes de tráfico en una Smart City	79
3.1	Introducción	79
3.2	Modelo de red simplificado	81

3.2.1	Usuarios y Dispositivos conectados a la Red Corporativa Municipal	82
3.2.2	Acceso a dispositivos IoT a través de servicios móviles M2M VPN-IP	82
3.2.3	Acceso a servicios y dispositivos IoT a través de servicios VPN-IP	83
3.2.4	Acceso a servicios y dispositivos IoT a través de internet.....	83
3.2.5	Acceso a la Plataforma de Ciudad	84
3.3	Tipos de tráfico en las redes de telecomunicaciones	84
3.3.1	Tráfico de Poisson	85
3.3.2	Tráfico autosimilar	86
3.4	Clasificación de las Fuentes de tráfico	87
3.5	Human Type Communication (HTC).....	88
3.5.1	Características de las Fuentes de voz en redes TDM.....	89
3.5.2	Características de las Fuentes de voz en redes convergentes	89
3.5.3	Caracterización propuesta para una fuente de voz.....	94
3.5.4	Características de las fuentes de vídeo en redes NGN.....	96
3.5.5	Caracterización propuesta para una fuente de vídeo	100
3.6	Machine Type Communication (MTC).....	102
3.6.1	Aplicaciones y requisitos de servicio de redes MTC.....	103
3.6.2	Modelos de tráfico MTC.	105
3.6.3	Caracterización propuesta para una fuente MTC	108
3.7	Calidad de servicio (QoS)	111
3.7.1	Parámetros de medida de la calidad de servicio (QoS)	111
3.7.2	Ancho de banda disponible.....	115
3.7.3	Requisitos de calidad por tipo de servicio	115
3.8	Conclusiones	123
4	Herramienta de Simulación SimulCity.....	125
4.1	Introducción	125
4.2	OMNeT++.....	126
4.2.1	Visión general de OMNeT++	126
4.3	SimulCity	130
4.3.1	Fuentes de Tráfico implementadas en SimulCity	131
4.3.2.	Diseño del Elemento de Red e interfaz UNI Metro Ethernet de acuerdo con especificaciones servicio Macrolan de Telefónica	137
4.3.2	Módulos diseñados en base a la biblioteca INET.....	141
4.3.3	Módulos adicionales utilizados en SimulCity	145

4.3.3.	Implementación del escenario	147
4.3.4	Configuración del escenario.....	153
4.3.5	Recopilación de estadísticas.....	165
4.4	Evolución de SimulCity	170
5	Casos de uso de SimulCity en proyectos VLCi del Ayuntamiento de València.....	173
5.1	Introducción	173
5.2	Iniciativa Impulso VLCi	174
5.2.1	Proyecto Gestión de Residuos Urbanos	178
5.2.2	Arquitectura de la solución.....	179
5.2.3	Proyecto Gestión Alumbrado Público.....	187
5.2.4	Proyecto Aparcamiento Regulado para Personas con Movilidad Reducida, Servicios de Carga y Descarga y Paradas de Taxis.....	194
5.3	Actualización servicio de telefonía corporativa municipal	202
5.4	Simulación del escenario global.....	211
5.5	Conclusiones	215
6	Conclusiones y líneas de trabajo futuras.....	217
6.1	Conclusiones	217
6.1.1	Marco de referencia.....	217
6.1.2	Arquitectura de red, análisis y caracterización de tráfico.....	218
6.1.3	SimulCity	219
6.2	Futuras líneas de trabajo.....	220
6.2.1.	Caracterización de nuevas fuentes de tráfico y control de la calidad de servicio	220
6.2.2.	Evolución de la herramienta SimuCity	220

Índice de figuras

Figura 2.1 Ámbitos clave de una Ciudad Inteligente	16
Figura 2.2 Relación entre Ámbitos y Requisitos de Ciudad Inteligente.....	21
Figura 2.3 Anatomía de Ciudad © City Protocol Society (2014) All Rights Reserved	23
Figura 2.4 Arquitectura TIC de una Ciudad Inteligente	27
Figura 2.5 Taxonomía de la capa Infraestructuras TIC	28
Figura 2.6 Taxonomía de la capa Control, Calidad de Servicio y Seguridad TIC	29
Figura 2.7 Taxonomía de la Capa Servicios y Aplicaciones TIC.....	30
Figura 2.8 Visión sectorial de la capa de aplicaciones (ITU-T Y.Supl. 30(16)_F03) ...	31
Figura 2.9 Modelo de capas de la Arquitectura TIC (ITU-T Y. Supl.30(16)_F04) ...	32
Figura 2.10 Topología típica de una red de sensores para captación de datos	35
Figura 2.11 Arquitectura Zigbee (http://www.zigbee.org).....	41
Figura 2.12 La nueva dimensión que introduce IOT (ITU-T Y.2060(12) F01)	43
Figura 2.13 Descripción técnica de IoT (ITU-T Y.2060(12)F02).....	44
Figura 2.14 Arquitectura IoT (http://www.IoT-Analytics.com – 2014).....	48
Figura 2.15 Evolución de IoT en 3GPP	51
Figura 2.16 Capa de Comunicaciones	53
Figura 2.17 Tecnologías de Redes de Acceso	54
Figura 2.18 Arquitectura red de acceso xDSL	55
Figura 2.19 Arquitectura FTTH (www.telefonica.com).....	56
Figura 2.20 Arquitectura de Centro de Datos Gestionado.....	59
Figura 2.21 Modelo de capas de la Plataforma de Ciudad Inteligente	67
Figura 2.22 Modelo de arquitectura horizontal M2M ETSI.....	69
Figura 2.24 Descripción alto nivel de la plataforma THINKING CITY de Telefónica	74
Figura 3.1 Arquitectura básica de un modelo de red dedicado.....	81
Figura 3.2 Arquitectura de comunicaciones típica de una Smart City	82
Figura 3.3 Terminología asociada a los procesos de tráfico.....	84
Figura 3.4 Modelo ON-OFF VoIP	91
Figura 3.5 Densidad de probabilidad medida y evaluada de los tiempos de llamada de paquetes para una llamada de VoIP (GSM codec) sin filtrado, y una distribución exponencial.....	92

Figura 3.6 Densidad de probabilidad medida y evaluada de los tiempos de llamada de paquetes para una llamada de VoIP (G.711 codec) sin filtrado, y una distribución Gamma.	93
Figura 3.7 Densidad de probabilidad medida y evaluada de los tiempos de llamada de paquetes para una llamada de VoIP (G.729 codec) sin filtrado, y una distribución log normal.	94
Figura 3.8 Modelado de una fuente de voz.....	95
Figura 3.9 Comportamiento fuente de voz durante 3 minutos con un códec G.711	96
Figura 3.10 Estructura de un GOP, con cuadros I, B y P.....	97
Figura 3.11 Tasas binarias para diferentes tipos de fuentes de vídeo.	100
Figura 3.12 Modelado de una fuente de vídeo H.264.....	101
Figura 3.13 Fuente vídeo Telepresencia durante 3 minutos con códec H.264 HD+ ...	102
Figura 3.14 Modelado de una fuente MTC.....	109
Figura 3.15 Ancho de banda de la fuente MTC definida.....	110
Figura 3.16 Histograma de bloques enviados en el período de simulación.....	111
Figura 3.17 Jitter de una señal periódica.....	114
Figura 3.18 Estructura del campo DSCP dentro de la trama de IPv4.....	118
Figura 3.19 Implementación de la calidad de servicio en un router.....	120
Figura 4.1 Jerarquía de módulos en OMNeT++ (https://www.OMNeTpp.org/).....	126
Figura 4.2 El IDE de OMNeT++.....	128
Figura 4.2 Arquitectura de comunicaciones de una corporación municipal.	130
Figura 4.3 Diagrama de funcionamiento de una fuente de voz.....	133
Figura 4.5 Diagrama de funcionamiento de una fuente de vídeo.....	135
Figura 4.6 Diagrama de funcionamiento de una fuente de datos.....	137
Figura 4.7 Implementación de QoS (clasificado, marcado y policing) en OMNeT++	139
Figura 4.8 Implementación de la gestión de colas en OMNeT++.....	140
Figura 4.9 Descripción del algoritmo trTCM implementado.....	141
Figura 4.10 Arquitectura del módulo StandardHost.....	142
Figura 4.11 Arquitectura del puerto de Ethernet de un StandardHost.....	143
Figura 4.12 Arquitectura del módulo Router.....	143
Figura 4.13 Arquitectura de un puerto PPP de entrada de un router.....	144
Figura 4.14 Arquitectura del módulo Internet Cloud.....	145
Figura 4.15 Diagrama de funcionamiento del medidor de ancho de banda.....	146
Figura 4.16 Implementación en OMNeT++ del escenario global.....	148

Figura 4.17 Módulo Centro Corporativo.....	149
Figura 4.18 Implementación del módulo Centro Corporativo en OMNeT++.....	150
Figura 4.19 Implementación del módulo Fuente en OMNeT++.....	150
Figura 4.20 Nube MPLS.....	151
Figura 4.21 Cloud privada con los servidores remotos.....	152
Figura 4.22 Implementación del módulo Cloud en OMNeT++.....	152
Figura 4.23 Implementación del módulo Server en OMNeT++.....	153
Figura 4.24 Configuración del códec de voz.....	154
Figura 4.25 Configuración de las calidades de vídeo.....	155
Figura 4.26 Contenido del fichero DataSize.xml.....	156
Figura 4.27 Configuración de los perfiles de tráfico.....	157
Figura 4.28 Configuración del filtrado.....	158
Figura 4.29 Configuración general del escenario.....	159
Figura 4.30 Definición de una fuente de voz.....	160
Figura 4.31 Definición de una fuente de vídeo.....	161
Figura 4.32 Definición de una fuente de datos.....	161
Figura 4.33 Escalado del escenario.....	162
Figura 4.34 Configuración del medidor de ancho de banda.....	163
Figura 4.35 Configuración de los parámetros de QoS generales.....	164
Figura 4.36 Parámetros usados para la configuración de la nube MPLS.....	165
Figura 4.37 Recopilación de estadísticas.....	166
Figura 4.38 Módulo interno a un router con medidor de ancho de banda.....	166
Figura 4.39 Operador de vectores para cálculo el jitter a partir de vector de retardos	167
Figura 4.40 Dataset y gráficas empleadas en el modelo.....	168
Figura 4.41 Ancho de banda de 5 fuentes de VoIP (perfil de tráfico “low”).....	169
Figura 4.42 Retardo extremo a extremo 5 fuentes de VoIP (perfil de tráfico “low”) .	169
Figura 4.43 Ancho de banda total en el interfaz Metro-Ethernet.....	170
Figura 4.44 Nuevo interfaz SimulCity múltiples dependencias.....	171
Figura 4.45 Nuevo interfaz SimulCity.....	171
Figura 4.3 Nuevo interfaz SimulCity parámetros de control de la calidad de servicio	172
Figura 5.1 Proyectos Iniciativa Impulso VLCi.....	175
Figura 5.2 Plano parcial con ubicación de los contenedores.....	179

Figura 5.3 Arquitectura solución gestión inteligente de residuos urbanos.....	180
Figura 5.4 Configuración de perfiles utilizados en SimulCity.....	182
Figura 5.5 Configuración de caudal en el interfaz Metro Ethernet.....	183
Figura 5.6 Resultados de la simulación (entrada Plataforma Ciudad)	183
Figura 5.7 Resultados de la simulación (fuentes de tráfico)	184
Figura 5.8 Ancho de banda consumido por los sensores de residuos	185
Figura 5.9 Retardo interfaz UNI Metro Ethernet - Plataforma de Ciudad	185
Figura 5.10 Paquetes enviados / recibidos	186
Figura 5.11 Jitter extremo a extremo	186
Figura 5.12 Histograma número de paquetes - retardo	187
Figura 5.13 Arquitectura solución alumbrado.....	189
Figura 5.14 Configuración de perfiles utilizados en SimulCity para c. luminarias.....	191
Figura 5.15 Configuración de caudal en el interfaz Metro Ethernet	191
Figura 5.16 Ancho de banda Centros de Mando Luminarias.....	192
Figura 5.17 Configuración de caudal en el interfaz Metro Ethernet	192
Figura 5.18 Retardo interfaz UNI Metro Ethernet - Plataforma de Ciudad	193
Figura 5.19 Paquetes enviados / recibidos	193
Figura 5.20 Jitter extremo a extremo	194
Figura 5.21 Plano con ubicación de plazas PMR, CyD y paradas de taxi	195
Figura 5.22 Arquitectura solución PMR, CyD y Taxi	196
Figura 5.23 Configuración perfiles utilizados en SimulCity para PMP, CyD, TAXI.	199
Figura 5.24 Configuración de caudal en el interfaz Metro Ethernet	199
Figura 5.25 Ancho de banda PMP, CyD, TAXI	200
Figura 5.26 Ancho de banda interfaz UNI PMP, CyD, TAXI.....	200
Figura 5.27 Retardo interfaz UNI Metro Ethernet - Plataforma de Ciudad	201
Figura 5.28 Paquetes enviados / recibidos	201
Figura 5.29 Jitter extremo a extremo	202
Figura 5.30 Arquitectura municipal consolidada.....	203
Figura 5.31 Configuración de las fuentes de tráfico de voz y multimedia.....	205
Figura 5.32 configuración de los perfiles de tráfico de voz y multimedia	206
Figura 5.33 Configuración de caudal en el interfaz Metro Ethernet	206
Figura 5.34 Ancho de banda fuentes de voz	207

Figura 5.35 Ancho de banda fuente de vídeo	207
Figura 5.36 retardo y Jitter de las fuentes de voz	208
Figura 5.37 retardo y Jitter de las fuentes de vídeo	209
Figura 5.38 Histograma del retardo de paquetes de vídeo en la red.....	210
Figura 5.39 Ancho de banda interfaz UNI	210
Figura 5.40 Ancho de banda fuentes de voz.....	212
Figura 5.41 Ancho de banda fuentes de vídeo	212
Figura 5.42 Ancho de banda agregado de las fuentes de datos IoT	213
Figura 5.43 Ancho de banda interfaz UNI	213
Figura 5.44 Retardo tráfico de voz.....	214
Figura 5.45 Retardo tráfico de vídeo.....	214

Índice de tablas

Tabla 2.1. Palabras Clave por Categorías	15
Tabla 2.2. Componentes del Sistema Estructura Física	24
Tabla 2.3. Componentes del Sistema Interacciones	25
Tabla 2.4. Componentes del Sistema Sociedad.....	25
Tabla 2.5. Comparativa entre estándares de conectividad inalámbricos.....	40
Tabla 2.6 Categorías LTE para uso en MTC (IoT).....	52
Tabla 3.1 Características del tráfico de voz	89
Tabla 3.2 Tipos de Codecs más comunes para VoIP	90
Tabla 3.3 Parámetros estadísticos en una red 3G para un códec GSM	92
Tabla 3.4 Distribución de los tiempos entre llegadas de paquetes para códec GSM	92
Tabla 3.5 Parámetros estadísticos en una red 3G para un códec G.711 ley A	93
Tabla 3.6 Distribución tiempos entre llegadas de paquetes en un códec G.711 ley A..	93
Tabla 3.7 Parámetros estadísticos en una red 3G para un códec G.729.....	94
Tabla 3.8 Distribución de tiempos entre llegadas de paquetes para un códec G.729....	94
Tabla 3.9 Parámetros para la caracterización de una fuente de voz.....	95
Tabla 3.10 Características del flujo de tráfico IPTV.....	96
Tabla 3.11 Funciones de autocorrelación para el tráfico de vídeo	98
Tabla 3.12 Parámetros del tráfico de vídeo.....	98
Tabla 3.13 Tasa binaria H.264 -bajo movimiento - 25 fps.....	101
Tabla 3.14 Parámetros para la configuración de una fuente de vídeo.....	101
Tabla 3.15 Modelos de tráfico del 3GPP TR 43.868	105
Tabla 3.16 Configuración de los modelos de tráfico TR 43.868	106
Tabla 3.17 Modelos de tráfico MTC 3GPP (TS-37.868).....	107
Tabla 3.18 Parámetros definidos para la configuración de una fuente MTC.....	109
Tabla 3.19 Parámetros de configuración de una fuente MTC arbitraria.	109
Tabla 3.20 Valores máximos de retardo por servicio (ITU-T E.800)	113
Tabla 3.21 SLA Retardo (round trip) servicio Macrolan	113
Tabla 3.22 Jitter máximo (<i>SLO</i>) para el servicio Macrolan	114
Tabla 3.23 SLA pérdida de paquetes servicio Macrolan.....	115
Tabla 3.24 Relación entre retardo y calidad (MOS)	116

Tabla 3.25 Correspondencia entre DSCP y tráfico equivalente en Macrolan	119
Tabla 4.1 Variables usadas en el modelado de una fuente de voz en OMNeT++	132
Tabla 4.2 Variables usadas en el modelado de una fuente de vídeo en OMNeT++	134
Tabla 4.3 Variables usadas en el modelado de una fuente de datos en OMNeT++	136
Tabla 4.4 Configuración por defecto de las colas	140
Tabla 4.5 Variables usadas en el medidor de ancho de banda	146
Tabla 4.6 Códecs de voz implementados	153
Tabla 4.7 Descripción de los elementos configurables en SimulCity	172
Tabla 5.1 Proyectos en el ámbito del Entorno Inteligente.....	176
Tabla 5.2 Proyectos en el ámbito de la Movilidad Inteligente	177
Tabla 5.3 Características de conectividad del dispositivo IoTsens.....	180
Tabla 5.4 Características de conectividad del dispositivo Controlador de Luminaria	189
Tabla 5.5 Características de conectividad del dispositivo de Libelium.....	197
Tabla 5.6 Características del tráfico de voz.....	203
Tabla 5.7 Valores máximos de QoS en fuentes de VoIP.....	208
Tabla 5.8 Valores máximos de QoS en fuentes de vídeo	208
Tabla 5.9 Resumen de fuentes por proyecto	211
Tabla 5.10 Caudales (CIR y EIR)	211

Abreviaturas y acrónimos

3G	Tercera generación de transmisión de voz y datos a través de telefonía móvil me diante UMTS (Universal Mobile Telecommunications System o servicio universal de telecomunicaciones móviles)
3GPP	Proyecto Asociación de Tercera Generación (3rd Generation Partnership Project)
4G	Cuarta generación de tecnologías de telefonía móvil
AAA	Autenticación, autorización y contabilidad (accounting)
ACELP	Algebraic Code Excited Linear Prediction
AENOR	Asociación Española de Normalización
AF	Assured Forwarding
API	Application Programming Interface
API REST	Protocolo de intercambio y manipulación de datos en los servicios de internet
APN	Access Point Name, puerta de enlace entre una red móvil GSM, GPRS, 3G o 4G y otra red informática, con frecuencia la Internet pública.
AS NGN	Servidores de aplicaciones del operador
ASV	Servidores de Aplicaciones Verticales
BDA	Base de Datos analítica
BHCA	Intentos de llamadas de hora ocupada
BI	Business Intelligence
BSI	British Standards Institution
CAN	Red de Control de Zona (Control Area Network)
CBS	Tamaño de ráfaga garantizado (Committed Burst Size)
CEP	Gestor de eventos
CIR	Caudal promedio garantizado (Committed Information Rate)
CKAN	Portal de datos abiertos
CRC	Código para detección de errores (cyclic redundancy check)
CS	Conmutación de circuitos
CS	Class Selector
DF	Default Forwarding
DSCP	Differentiated Services Code Point (QoS)

DSL	Líneas digitales de abonado
DSLAM	Multiplexor localizado en la central telefónica que proporciona a los abonados acceso a los servicios DSL sobre cable de par trenzado de cobre (Digital Subscriber Line Access Multiplexer).
DWDM	Multiplexación densa por longitud de onda (Dense Wavelength Division Multiplexing)
EF	Expedited Forwarding
ERP	Sistemas de planificación de recursos empresariales (enterprise resource planning)
ESB	Carga de datos en tiempo real
ETL	Extract, Transform and Load
ETLS	Herramientas de carga de información e integración con sistemas externos
ETSI	Organización de estándares europea
FCAPS	Gestión de fallos, de la configuración, de la contabilidad, del rendimiento y de la seguridad
FG-SCC	Grupo Temático sobre Ciudades sostenibles e inteligentes
FPS	Tramas por segundo
FTTB	Fibra hasta el edificio
FTTC	Fibra hasta la acera
FTTD	Fibra hasta el puesto de trabajo
FTTH	Fibra hasta el hogar
FTTLA	Fibra hasta el último amplificador
FTTN	Fibra hasta el vecindario
FTTx	Término genérico para designar cualquier acceso de banda ancha sobre fibra óptica que sustituya total o parcialmente el cobre del bucle de acceso
GIS	Sistema de información geográfica
GOP	Group of Pictures
GPON	Gigabit Passive Optical Network
GPRS	Servicio general de paquetes vía radio (General Packet Radio Service)
GPS	Sistema de posicionamiento global (Global Positioning System)
GSM	Sistema global para las comunicaciones móviles (Global System for Mobile communications)
H.264	Estándar de vídeo

Hadoop	Framework de software que soporta aplicaciones distribuidas bajo una licencia libre
HD	High Definition
HDFS	Sistema de ficheros distribuido que almacena datos en equipos de consumo
HQ	High Quality
HTC	Comunicación humana (Human Type Communication)
IDM	Identity Manager
IMS	Conjunto de especificaciones que describen la arquitectura de las redes NGN (IP Multimedia Subsystem)
IoT	Internet de las Cosas (Internet of Things)
IP	Protocolo de Internet (Internet Protocol)
ISM	Industrial, Scientific and Medical radiofrecuency
ISO	Organización encargada de la estandarización de protocolos (International Organization for Standardization)
ITU	Unión Internacional de Telecomunicaciones
JMX	Tecnología que define una arquitectura de gestión, la API, los patrones de diseño, y los servicios para la monitorización/administración de aplicaciones basadas en Java
JSON	JavaScript Object Notation
KPI's	Indicadores que nos ayudan a identificar el rendimiento de una Ciudad Inteligente (Key Performance Indicator)
LC_LTE	LTE de bajo coste
LDAP	Protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.
LiFi	Sistemas de comunicaciones inalámbricas rápidos y de bajo costo (light fidelity)
LoRa-WAN	Especificación para redes de baja potencia y área amplia
LPWAN	Red de cobertura amplia de bajo consumo (Low-Power Wide-Area Network)
LTE	Estándar para comunicaciones inalámbricas de transmisión de datos de alta velocidad para teléfonos móviles y terminales de datos (Long Term Evolution)
M2M	Intercambio de información en forma de datos entre máquinas (Machine to Machine)

MAC	Capa de control de acceso al medio
MGW	Media Gateway
MMPP	Markov-modulated Poisson process
MongoDB	Base de datos orientada a documentos
MOS	Mean Opinion Score
MPEG	Estándar de vídeo
MPLS	Conmutación de etiquetas multiprotocolo1 (Multiprotocol Label Switching)
MQTT	Message Queue Telemetry Transport
MT	Móvil Terminado Infrecuente
MTC	Forma de comunicación de datos entre dispositivos que no requieren la intervención humana (Machine-type Communication)
MySQL	Base de Datos
NbIoT	Redes móviles de banda estrecha diseñadas para IoT (Narrowband IoT)
NGN	Red de Siguiete Generación (Next Generation Networking)
NGSi	Next Generation Services Interface
OA&M	Operación, Administración y Mantenimiento
oData	Open Data Protocol
OLT	Elemento activo situado en sitio central de equipamiento (Optical line terminal)
OneM2M	Estándar de M2M
ONT	Terminación de red óptica (Optical Network Termination)
OTN	Optical Transport Network
PaaS	Virtualización. Plataforma como Servicio
PAP/PDP	Policy Administration Point/Policy Decision Point
PBS	Tamaño máximo del pico de la ráfaga (Peak Burst Size)
PEP	Policy Enforcement Point
PHY	Capa física de radio
PIR	Tasa de información de pico (Peak Information Rate)
Plataforma SSC	Smart sustainable cities
PLC	Power Line Communications

PON	Redes ópticas pasivas (Passive optical network)
PoS	Puntos de Venta
PQ	Priority Queueing
PS	Conmutación de paquetes
PSTN	Red telefónica pública conmutada
QoE	Calidad de la Experiencia (Quality of Experience)
QoS	Calidad de Servicio (Quality of Service)
RCM	Red Corporativa Municipal
RDF	Marco de descripción de recursos (Resource Description Framework)
RED	Random Early Detection
Red ad-hoc	Conjunto de redes donde todos los nodos tienen el mismo estado dentro de la red y son libres de asociarse con cualquier otro dispositivo de red
RFC	Request for Comments
RFID	Identificador por radiofrecuencia (Radio Frequency Identification)
RPO	Objetivo de Punto de Recuperación
RSVP	Resource Reservation Protocol
RTO	Objetivo de Tiempo de Recuperación
RTP	Real-time Protocol
SCADA	Software para ordenadores que permite controlar y supervisar procesos industriales a distancia (Supervisory Control And Data Acquisition)
SD	Standard Definition
SDH	Jerarquía Digital Síncrona
SDK	Software Development Kit
SG20	Grupo de estudio para la estandarización de las tecnologías IoT
Sigfox	Solución de conectividad celular mundial para el Internet of Things pensada para comunicaciones de baja velocidad que permite reducir los precios y el consumo de energía para los dispositivos conectados.
SLA	Acuerdo de nivel de servicio (Service Level Agreement)
SLC	Controladores de luminaria
SNMP	Protocolo estándar de monitorización
SRR	Shared Round Robin
SSC	Ciudad Inteligente (Smart Sustainable City)
STH	Histórico a corto plazo

Sw	Software
TCP	Protocolo de control de transmisión
TDM	Time Division Multiplexing
TE legacy	Terminal clásico
TE NGN	Terminal NGN
TIC	Tecnologías de la Información y de las Comunicaciones
UDP	Protocolo del nivel de transporte basado en el intercambio de datagramas (User Datagram Protocol)
UE	Equipamiento de usuario
UNI	Interfaz de red del usuario (User Network Interface)
VLAN	Virtual Local Area Network
VoIP	Voz sobre protocolo IP (Voice over IP)
VPN	Red privada virtual
VPN-IP	Servicio de Telefónica
WAN	Red de área amplia (Wide Area Network)
WDM	Multiplexación por división de longitud de onda (Wavelength Division Multiplexing)
WFQ	Weighted Fair Queue
Wi-Fi	Tecnología que permite la interconexión inalámbrica de dispositivos electrónicos
WPAN	Red inalámbrica de área personal
WRED	Weighted Random Early Detection
WRR	Weighted Round Robin
WSN	Redes de sensores inalámbricos (Wireless sensor networks)
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

Capítulo 1

Introducción

1.1 Introducción

La Agenda 2030 de Naciones Unidas para el Desarrollo Sostenible [1] incluye entre sus objetivos “Lograr que las ciudades y los asentamientos humanos sean inclusivos, seguros, resilientes y sostenibles”. En esta línea, la Nueva Agenda Urbana HABITAT III [2] y específicamente el documento de políticas 9: “Servicios Urbanos y la tecnología” [3] recogen la necesidad de una integración de las comunicaciones en los servicios e infraestructuras urbanas.

El atractivo y las oportunidades que ofrecen las ciudades ha provocado que una gran parte de la población mundial se desplace a las áreas urbanas con la esperanza de mejorar su calidad de vida generando una demanda creciente de recursos, energía, agua, sanidad y educación entre otros. Actualmente las ciudades generan más del 70% de las emisiones de gases de efecto invernadero y entre el 60-80 % del consumo global de energía. Asumiendo que aproximadamente un 70% de la población mundial vivirá en ciudades en el 2050, su gestión sostenible se ha convertido en un elemento clave en la política de las administraciones en todo el mundo.

El concepto de Ciudad Inteligente o Smart City surge como respuesta global a los múltiples retos de este sistema urbano de gran complejidad a los que deben de enfrentarse los gestores municipales con el fin de mejorar la eficiencia, la sostenibilidad y ofrecer más y mejores servicios a los ciudadanos. [4].

Las Tecnologías de la Información y de las Comunicaciones (TIC) desempeñan un papel crucial para soslayar estos retos y aprovechar nuevas oportunidades, incrementando la

eficiencia y la innovación, facilitando la integración de los servicios municipales, la cooperación entre sectores y como ayuda en la toma de decisiones en las áreas clave del desarrollo urbano: economía, gobernanza, entorno y sociedad.

Esta gestión eficiente de los recursos y servicios disponibles requiere no solamente dotar de “inteligencia” determinadas infraestructuras físicas y lógicas, sino también recolectar y gestionar en tiempo real un volumen de información cada vez más elevado. Como resultado, información de muy diversa índole ha de ser procesada e interpretada, no solamente para monitorizar el funcionamiento de la ciudad, sino también para identificar sinergias y mejorar la eficiencia.

Dotar de inteligencia a las infraestructuras físicas existentes requiere un esfuerzo considerable. Energía, infraestructura urbana, transporte y movilidad, saneamiento, gestión de residuos, seguridad, salud y educación son solamente ejemplos que requieren un tratamiento específico, pero bajo una perspectiva global. El proyecto “SmartSantander” [5] constituye un ejemplo de soporte y gestión integral de aplicaciones y servicios en una ciudad inteligente.

Esta tesis, se centra en construir una herramienta que permita la evaluación de prestaciones y el impacto de la implantación de nuevas soluciones sobre la infraestructura de red existente, definiendo para ello de la arquitectura tecnológica de una Smart City y los dispositivos que la componen.

El desarrollo del presente trabajo ha supuesto el análisis de diferentes fuentes de datos, el modelado de su comportamiento y el análisis del impacto de su integración en la arquitectura.

Para la confección de esta tesis se han considerado especialmente los trabajos de la Unión Internacional de Telecomunicaciones a través de las recomendaciones y especificaciones técnicas del Grupo Temático sobre Ciudades sostenibles e inteligentes (FG-SCC) y de los Grupos de Estudio SG12 - Performance, QoS and QoE, y SG20 - IoT y sus aplicaciones, incluidas las ciudades y comunidades inteligentes (C+CI).

Además, en esta tesis se recoge la aplicación de los temas descritos con anterioridad al proyecto Smart City de Valencia (Valencia Ciudad Inteligente). Concretamente se analizará el impacto de algunos proyectos recogidos en la Iniciativa “Impulso VLCi” del Ajuntament de València correspondiente a la convocatoria de concesión de ayudas para el desarrollo del Programa de Ciudades Inteligentes de la Agenda Digital para España [6].

1.2 Motivación de la Tesis

En las estrategias de desarrollo de las “Ciudades Inteligentes” [7], las redes de telecomunicación avanzadas juegan un papel primordial. Para conseguir una gestión eficiente y sostenible de los recursos globales, es necesario analizar conjuntamente el ecosistema de comunicaciones formado por un gran número de dispositivos (personas, infraestructuras, sensores-actuadores) de características heterogéneas y por la propia red que permita su conectividad con la calidad, seguridad y disponibilidad esperada a mínimos costes.

En este escenario de “Smart Cities and the Internet of Everything”, la Red de Acceso Convergente de Siguiete Generación (Next Generation Network, NGN) [8, 9], tanto fija como móvil, juega un papel fundamental pues ha de ser capaz de adaptarse a los incrementos, en muchos casos abruptos, del tráfico generado por miles de dispositivos de naturaleza, criticidad y comportamiento heterogéneo. Tráficos isócronos como el generado por la voz, de tasa variable para servicios de vídeo, asíncrono para tráfico de alarmas, best-effort para servicios no críticos, etc. con diferentes necesidades de latencia, pérdida de paquetes y jitter, compiten por un ancho de banda limitado.

Existen múltiples tecnologías disponibles tanto fijas como inalámbricas para responder a este requerimiento y un importante esfuerzo normalizador para adaptar las redes a estos nuevos perfiles de tráfico [10, 11]. En apartados posteriores se describirán muy someramente las principales tecnologías en redes cableadas e inalámbricas aplicables a entornos de área amplia como LoRaWAN, Sigfox, OnRamp wireless, Weightless -N & -P etc. que operan en banda ISM y soluciones basadas en las recomendaciones del 3GPP como IoT de banda estrecha (NB-IoT) en bandas licenciadas.

Este ecosistema es dinámico y altamente impredecible dado que las necesidades de calidad, ancho de banda y disponibilidad crecen y varían tanto espacial como temporalmente. Diferentes áreas en la ciudad, financieras o residenciales muestran patrones de comportamiento muy diferentes y con gran variabilidad a lo largo de la jornada. Además, la información crítica ha de estar disponible y ser prioritaria sobre otra menos relevante [12]. Conseguir que la red sea capaz de adaptarse a los incrementos, en muchos casos abruptos, de tráfico manteniendo la eficiencia y la calidad de los servicios subyacentes, es una tarea compleja y que conlleva un importante impacto en costes.

A diferencia de las redes de telecomunicaciones clásicas, servicios y aplicaciones de muy diferente naturaleza comparten recursos de una infraestructura común. Con estas premisas, la disponibilidad de los recursos de red puede cambiar dinámicamente en el tiempo y producirse congestión, degradando la calidad de servicio (QoS) comprometida en las recomendaciones internacionales [13, 14]. Las redes con conexiones oportunistas como, por ejemplo, las redes móviles o las redes ad-hoc, son especialmente vulnerables a estos cambios.

La categorización y estandarización de este tipo de tráfico es un área en constante desarrollo. 3GPP en TS 22.368 [15] define categorías y funcionalidades de fuentes Machine

Type Communications (MTC), no así su patrón de comportamiento. ITU-T en su recomendación Q.3925 [16] define diferentes tipos de fuentes para la comprobación de los parámetros de calidad de servicio en NGN, pero de los siete tipos de fuentes MTC descritos, deja tres para estudios posteriores.

Gran parte de la bibliografía existente y artículos de investigación se centran en caracterizar flujos de tráfico de fuentes agregadas MTC, asumiendo cierta “homogeneidad” en su comportamiento y un número “elevado” de fuentes de acuerdo con las premisas clásicas de las redes de sensores (WSN). Este modelado de tráfico trata la información agregada de dispositivos homogéneos MTC en un único flujo, realizando un análisis similar al que se utiliza en redes troncales para enlaces de alta velocidad [17].

Garantizar la calidad, disponibilidad y prestaciones de los servicios cuando éstos compiten por recursos limitados, como es el ancho de banda disponible, es un área de creciente interés, especialmente en entornos dinámicos como el descrito. Las MTC y las Human Type Communications (HTC) con códec avanzados siguen patrones de tráfico novedosos, algunos de ellos escasamente caracterizados.

Describir analíticamente el comportamiento de un escenario de estas características es una tarea compleja especialmente si el entorno es dinámico. Sin embargo, simular escenarios con fuentes de tráfico configurables y evaluar su respuesta ante diferentes variaciones de tráfico o la introducción de nuevas fuentes de datos, permitirá evaluar el impacto en términos de Calidad de Servicio, y optimizar los recursos dado que existe una relación directa entre consumo de ancho de banda y costes.

Por lo tanto, diseñar un entorno que permita simular el ecosistema de fuentes de tráfico – red convergente en una Smart City, aportará flexibilidad y facilitará un ahorro en costes al simplificar las tareas de diseño, evaluación de prestaciones y crecimiento de la red.

1.3 Objetivos de la Tesis

El trabajo desarrollado en la presente tesis, se puede encuadrar en el área de la investigación aplicada, dirigido y enfocado principalmente a diferentes escenarios (aplicación en casos de uso).

El **objetivo general** de la Tesis es el análisis y la caracterización del tráfico en una Smart-City mediante el modelado del perfil de tráfico de dispositivos reales y el análisis de la calidad de servicio proporcionada por una red de acceso convergente NGN (Next Generation Network) en el interfaz UNI (User Network Interface) cuando un número arbitrario de fuentes compiten por los recursos disponibles.

- Un **primer objetivo** específico consiste en la identificación de escenarios y servicios en entornos convergentes HTC (Human Type Communications) + MTC (Machine Type Communications), caracterizando los diferentes tipos de fuentes de tráfico y su comportamiento. Para ello se han definido: fuentes HTC comunicaciones humano-humano para las comunicaciones de baja latencia y tiempo

real como voz, multimedia, y fuentes MTC comunicaciones donde intervine una máquina para comunicaciones con requerimientos heterogéneos como alarmas, sensores, acceso a fuentes de información, etc.

- El **segundo objetivo** específico consiste en simular los mecanismos de calidad de servicio de una red de acceso convergente, gestión de tráfico y control de admisión, emulando un servicio real de operadora basado en accesos Metro-Ethernet y núcleo de red MPLS.
- El **tercer objetivo** específico ha sido diseñar una herramienta de simulación diseñada ad-hoc que caracterice el comportamiento agregado de un número arbitrario de fuentes HTC/MTC IP heterogéneas conectadas a un interfaz UNI configurable con tres clases de servicio. Los resultados como el ancho de banda disponible, el retardo, el jitter y la pérdida de paquetes por clase de servicio y fuente, permitirá construir escenarios más realistas en entornos dinámicos, evaluar las prestaciones del sistema, el impacto de la introducción de nuevas soluciones en el ámbito urbano y garantizar los objetivos de calidad extremo-extremo recogidos en las recomendaciones internacionales.

De manera complementaria, y fruto de esta investigación se alcanzarán los siguientes objetivos:

1. Definir conceptualmente los atributos y requisitos que sirven de base al desarrollo de la arquitectura TIC de una Smart City.
2. Diseñar una biblioteca de modelos de fuentes de tráfico HTC y MTC para los servicios y dispositivos más comunes, VoIP, vídeo H.264 y diferentes clases de sensores.
3. Describir y justificar los requisitos de calidad de servicio en el plano de las aplicaciones.
4. Simulación del escenario resultado del proyecto “*Impulso VLCi*” del Ayuntamiento de València.

Esta investigación se ha abordado en tres fases de acuerdo con el modelo conceptual de la arquitectura:

- Primera fase, en el Dominio de Usuario, donde se caracterizarán las fuentes HTC y MTC más comunes, modelo de tráfico, comportamiento y agregación de fuentes homogéneas.
- Segunda fase, en el Dominio de la Red (capa de transporte en NGN), se tratarán de simular los mecanismos de Control de Admisión en el Acceso, ancho de banda disponible, clases de servicio, funciones policía, gestión de colas y tránsito de tráfico por la red MPLS.

- Tercera fase, en el Dominio de las Aplicaciones (capa de servicio en NGN) donde se tratarán de evaluar parámetros relacionados con la Calidad de Servicio extremo a extremo y la gestión de flujos de tráfico agregado. En esta etapa, se estudiarán las prestaciones de un escenario real IoT con diferentes dispositivos. Para ello se de utilizar la Plataforma Smart-City del Ayuntamiento de Valencia.

Esta actividad investigadora está incluida en el objetivo 05 del Proyecto “MEC TIN2013-47272-C2-1-R - Plataforma de Servicios para Ciudades Inteligentes con Redes M2M Densas (PLASMA) Caracterización de servicios MTC/IoT y modelado de patrones de tráfico”, proyecto donde el autor de esta tesis forma parte del equipo de trabajo.

1.4 Principales aportaciones

1.4.1 Artículos.

Clave: Revista Indexada		
Título: Evaluation of Quality of Service in Smart-Hospital Communications		
Autores (por orden de firma): Angel Gomez-Sacristan, Miguel A. Rodriguez-Hernandez, and Victor Sempere		
Revista: Journal of Medical Imaging and Health Informatics		
Editorial: American Scientific Publishers		
Volumen: 5 No.8	Páginas: 1864-1869	Año: 2015
DOI: 10.1166/JMIHI.2015.1660		JCR. IF:0.877

Clave: Revista Indexada		
Título: Virtual Laboratory for QoS Study in Next Generation Networks with Metro Ethernet Access		
Autores (por orden de firma): Angel Gomez-Sacristan, M. Ángel Rodríguez, V.M. Sempere		
Revista: IEEE Transactions on Education		
Editorial: IEEE		
Volumen: 59, No.3	Páginas: 187-193	Año: 2016
DOI: 10.1109/TE.2015.2498120		JCR. IF:1.727

Clave: Revista Indexada		
Título: 2D biological representations with reduced speckle obtained from two perpendicular ultrasonic arrays		
Autores (por orden de firma): Miguel A. Rodriguez-Hernandez, Angel Gomez-Sacristan, and Victor Sempere		
Revista: Journal of Technology and Health Care (THC)		
Editorial: IOS Press		
Volumen: 24 No. S2	Páginas: 561-568	Año: 2016
DOI: 10.3233/THC-161182		CR. IF: 0.724

1.4.2 Congresos

Clave: Internacional
Título: Communications for Sanitary Teleservices in a Smart City
Autores (por orden de firma): Ángel Gómez-Sacristán, Miguel A. Rodríguez-Hernández, and Franklin G. Jiménez-Peralta.
Congreso: Global Medical Engineering Physics Exchanges & Pan American Health Care Exchanges (GMEPE/PAHCE 2018)
ISBN: 978-1-5386-5474-3, ISSN: 2327-8161. IEEE CATALOG NUMBER CFP1818G-PRT. Páginas 188-191
Fecha de celebración: 19-24 marzo 2018

Clave: Internacional
Título: Simulation of the Communications System for a National Radiation-Dose Databank.
Autores (por orden de firma): Miguel A. Rodríguez-Hernández, and Ángel Gómez-Sacristán.
Congreso: Global Medical Engineering Physics Exchanges & Pan American Health Care Exchanges (GMEPE/PAHCE 2018)
ISBN: 978-1-5386-5474-3, ISSN: 2327-8161. IEEE CATALOG NUMBER CFP1818G-PRT. Páginas 184-187
Fecha de celebración: 19-24 marzo 2018

Clave: Internacional
Título: Simulation of a Communication System in Castile-La Mancha for a National Radiation Dose Data Bank
Autores (por orden de firma): Franklin G. Jiménez-Peralta, Ángel Gómez-Sacristán, and Miguel A. Rodríguez-Hernández
Congreso: Proceedings of the Workshop on Innovation on Information and Communication Technologies (ITACA-WIICT 2017)
ISB/ISSN: 978-84-697-7327-7 Páginas: 12-19
Fecha de celebración: 16 junio 2017

Clave: Internacional
Título: Simulation of Telecom Traffic Sources in Hospitals
Autores (por orden de firma): Yang LI, Ángel Gómez Sacristán, Miguel A. Rodríguez Hernández
Congreso: Workshop on Innovation on Information and Communication Technologies (ITACA-WIICT 2016)
ISB/ISSN: 978-84-617-7199-8. Páginas: 50-59
Fecha de celebración: 17 junio 2016

Clave: Internacional
Título: Telecom Services Design in Smart-Hospital Communications
Autores (por orden de firma): Ángel Gómez Sacristán, Miguel A. Rodríguez Hernández, Victor Miguel Sempere Payá
Congreso: Global Medical Engineering Physics Exchanges & Pan American Health Care Exchanges (GMEPE / PAHCE 2016)
ISB/ISSN: 978-1-5090-2485-8 IEEE Catalog Number: CFP1618G-PRT. Páginas: 89-94
Fecha de celebración: 4-9 abril 2016

Clave: Internacional
Título: Simulation of All-IP Access Network with Configurable Traffic Sources
Autores (por orden de firma): David Gómez, Victor M. Sempere, Miguel A. Rodríguez, Ángel Gómez
Congreso: Workshop on Innovation on Information and Communication Technologies (ITACA-WIICT 2014)
ISBN/ISSN: 978-84-697-1166-8 Páginas: 101-110
Fecha de celebración: 4 julio 2014

1.4.3 Participación en proyectos de investigación

Clave: Proyecto de Investigación competitivo
Título del proyecto/contrato: TIN2013-47272-C2-1-R Plataforma De Servicios para Ciudades Inteligentes con redes M2M Densas (PLASMA)
Entidad convocante/financiadora: Ministerio Economía y Competitividad
Fecha de inicio: 1-1-2014 Fecha de finalización: 31-12-2017
Investigador principal: D. Vicent Pla Boscá y D. Luis Guijarro Coloma
Importe de la subvención: 135.103€ Nº total de investigadores del proyecto: 12

1.4.4 Software

Se han desarrollado una aplicación basada en el entorno de simulación OMNET++ que permite el modelado y simulación de entornos HTC+ MTC en una Smart City. Esta herramienta permite evaluar diferentes parámetros de calidad extremo-extremo recogidos en las recomendaciones internacionales, ancho de banda, pérdida de paquetes, retardo extremo a extremo y jitter.

1.4.4.1 Modelado de las fuentes de tráfico

Cada fuente puede definirse con un modelo de tráfico (perfil) característico y un comportamiento (actividad) que generalmente es una variable aleatoria con una distribución definida [18, 19].

Los parámetros utilizados para definir las fuentes HTC de voz son el códec de telefonía IP seleccionado, y el número de fuentes homogéneas. Los parámetros que definen el comportamiento son el tiempo de conversación y el tiempo de inactividad. Los

parámetros utilizados para definir las fuentes HTC de vídeo son: el códec de vídeo utilizado (H.264 por defecto), la calidad de imagen (SD: 320x180 píxeles, HQ: 640x360 píxeles, HD: 1280x720 píxeles y HD+ con: 1920x1080 píxeles), el número de fotogramas por segundo, el grado de movimiento (bajo, medio, alto) y el número de fuentes homogéneas. Los parámetros que definen el comportamiento son las variables aleatorias tiempo activo y tiempo de inactividad.

Las fuentes MTC, se pueden caracterizar por el tiempo de actividad y la distribución de los datos generados en un período. Las fuentes de tráfico MTC se han modelado como un flujo arbitrario de información compuesto por "s" ventanas de transmisión o sesiones distribuidas aleatoriamente en el período de simulación. En cada ventana, el origen envía varios bloques de información de tamaño variable que simular imágenes y de datos o recursos. El número de archivos, imágenes y su tamaño se pueden modelar utilizando distribuciones aleatorias.

1.4.4.2 *Simulación y evaluación de los mecanismos de Calidad de Servicio (QoS) en la red de Acceso*

Las funciones básicas de calidad de servicio en la red de acceso (Access Network) incluyen la clasificación de paquetes, el marcado y la función policía, así como la gestión de colas para cada tipo de tráfico [20-22].

Con el fin de obtener una simulación lo más similar posible a un entorno real, se ha utilizado como referencia un servicio IP basado en la tecnología Metro Ethernet ofrecido por un operador español [23] y el Cisco Catalyst 3650 Switch-Router [24]. En este servicio se definen tres categorías de tráfico: Multimedia, Oro y Plata, siendo Multimedia del tráfico con la máxima prioridad y Plata el de mínima.

El tráfico se clasifica, se filtra y se marca en el router. Las reglas de filtrado dependen de varios criterios, por ejemplo: dirección IP (origen/destino), el puerto UDP (origen/destino), etc. Una vez que el paquete está clasificado y marcado, será procesado por el bloque de QoS, donde se implementan los mecanismos de calidad asociados a cada tipo de tráfico. (CIR - Caudal promedio garantizado de entrada, CBS – Tamaño de ráfaga garantizado, PIR- Tasa de información de pico, y PBS – Tamaño máximo del pico de la ráfaga).

La simulación del Core Network, retardo y pérdida de paquetes se realiza aplicando a cada paquete una variable aleatoria exponencial negativa acotada con los valores de nivel de servicio (SLA) máximos y mínimos definidos por el operador en el contrato del servicio.

1.4.4.3 *Evaluación del tráfico y calidad de servicio extremo a extremo. Dimensionado de caudales, servicios y aplicaciones verticales*

En el dominio de las Aplicaciones confluye el tráfico generado por las diferentes redes de acceso de la Smart-City. Es en este dominio, ubicado normalmente en servidores Cloud donde residen los Servidores de Aplicaciones Verticales (ASV), parte del plano

de control de IMS y el Context Broker de Servicios M2M. En este punto, se analiza el comportamiento y prestaciones relacionadas con la gestión de la calidad de servicio extremo a extremo por cada clase de servicio y/o servicio subyacente: retardo total, jitter y pérdida de paquetes.

1.4.5 Participación en proyecto Valencia Ciudad Inteligente

- Elaboración de la Norma AENOR PNE 178201 “Ciudades inteligentes. Definición, atributos y requisitos”.
- Certification Platinum ISO 37120 "Sustainable development of communities, Indicators for city services and quality of life”.
- Elaboración memoria “Solicitud Ayuda II edición Ciudades Inteligentes Agenda Digital España (C059/15-AE)”.
- Colaboración elaboración norma ITU L.1603 y apoyo a la Certificación de la Ciudad de Valencia.
- Elaboración en colaboración con la Entidad Pública Red.es de la documentación técnica de la Iniciativa “Impulso VLCi” del Ayuntamiento de Valencia.
- Elaboración memoria Solicitud Ayuda C052/17-SP concesión de ayudas para el desarrollo del plan nacional de territorios inteligentes para la agenda digital para España (pilotos de edificios inteligentes).

1.5 Organización de la memoria

La memoria de la tesis está estructurada de la siguiente manera:

- El segundo capítulo, introduce tres aspectos de las Smart Cities que el autor considera relevantes, el primero de índole conceptual [atributos, requisitos y semántica]. El segundo, la arquitectura física y lógica [redes de acceso convergente cableadas e inalámbricas (NbIoT), Internet of Things (IoT), redes de transporte y aplicaciones sectoriales]. Y, por último, las Plataformas de Ciudad [AENOR 178104, Plataforma Thinking City].
- El tercer capítulo, describe los escenarios de conectividad, la arquitectura genérica, el modelado de fuentes HTC / MTC y aspectos sobre calidad de servicio (QoS).
- El cuarto capítulo, describe la herramienta de simulación (SimulCity) que se ha diseñado ad hoc para evaluar el impacto en diferentes parámetros de calidad de servicio. Se describirá el entorno de desarrollo, los módulos de simulación de fuentes de tráfico, funcionales de calidad de servicio, cloud, así como los módulos medidores de parámetros relevantes, como el ancho de banda, la pérdida de paquetes, el jitter, etc.

- El quinto capítulo, detalla la aplicación de SimulCity al caso de uso de varios proyectos IoT de la Iniciativa “Impulso VLCi” del Ayuntamiento de Valencia, así como la evolución de la red de comunicaciones de voz municipal a un entorno de comunicaciones unificadas basado en telefonía IP. Se describen los proyectos que configuran la iniciativa, los componentes del sistema y arquitectura, el modelado de fuentes, la simulación del escenario y por último un dimensionado de caudales.
- El sexto capítulo, resume las conclusiones alcanzadas en esta tesis, así como posibles líneas futuras de investigación.
- Por último, se listan las referencias citadas en el documento.

Capítulo 2

Ciudades Inteligentes

2.1 Introducción

En este capítulo se describen brevemente los principales elementos sobre la que esta tesis se soporta:

- La visión conceptual de una Ciudad Inteligente (Smart Sustainable City o SSC): atributos, requisitos y semántica.
- La Arquitectura TIC de una Ciudad Inteligente: fuentes HTC y MTC (IoT), redes de acceso convergentes fijas y móviles; redes de transporte.
- La Plataforma de Ciudad Inteligente adoptada por el Ayuntamiento de València (Plataforma VLCi) que está basada en la solución *THINKING CITY* desarrollada por Telefónica.

2.2 Smart Cities o Ciudades Inteligentes

Como ya se apuntó en la introducción, las ciudades constituyen un poderoso motor de crecimiento económico y social gracias a las oportunidades que ofrecen respecto al medio rural: mayor diversidad y calidad de empleo y mejores infraestructuras y servicios. Este atractivo ha provocado que una gran parte de la población mundial se desplace a las áreas urbanas con la esperanza de mejorar su calidad de vida.

La necesidad de desarrollar un modelo sostenible de crecimiento y gestión ha favorecido la proliferación de ciudades que se definen a sí mismo “inteligentes” en uno u otro

sentido. Existen más de 100 definiciones de Ciudad Inteligente y una literatura muy extensa publicada por organismos gubernamentales nacionales e internacionales, organismos de estandarización y grupos de interés que abordan aspectos muy diversos de este tema.

Destacan especialmente los trabajos de la British Standards Institution (BSI) en cuanto al modelo conceptual [26, 27], la Unión Internacional de Telecomunicaciones a través del Grupo Temático sobre ciudades sostenibles e inteligentes (ITU-T FG-SSC) y su Comisión de Estudio 20 del UIT-T – Internet de las Cosas y Ciudades y Comunidades Inteligentes en los aspectos de normalización de la arquitectura TIC de una Ciudad Inteligente [28-34], redes de sensores [35, 36], interoperabilidad entre aplicaciones [37], edificios inteligentes [38], gestión [39], datos abiertos [40], requerimientos del entorno residencial en una Smart City [41] e indicadores clave relacionados con diferentes aspectos [42-45].

Finalmente, reseñar la contribución del City Protocol Agreement [46] en materia de arquitectura, de la entidad Pública Red.es [47] sobre metodología de Ciudades Inteligentes, ISO/TC 268 (UNE-ISO 37120:2015) [48] sobre desarrollo sostenible en las ciudades. Indicadores para los servicios urbanos y calidad de vida y por último la recomendación AENOR AEN/CTN 178 PNE 178201:2016 [49] “Ciudades inteligentes. Definición, atributos y requisitos”, Asociación Española de Normalización (AENOR).

2.2.1 Concepto de Ciudad Inteligente

Existe una gran cantidad de definiciones, indicadores y métricas relativas a Ciudades Inteligentes recogidas en la literatura como se ha descrito en las referencias anteriores, pero muchas de ellas poseen enfoques sectoriales o no incluyen determinados atributos que se consideran clave.

La definición de Ciudad Inteligente, sostenible y abierta a los ciudadanos ha de:

- Reflejar de manera unívoca qué es y para qué sirve.
- Ser general para incluir a todos los agentes involucrados.
- Flexible para adaptarse a los cambios.
- Describir los atributos relevantes que la caracterizan y que la hacen diferente de las demás.

El conjunto de atributos que caracterizan a una Ciudad Inteligente constituye la pieza básica que se utilizará no solamente en el diseño de una definición estructurada del término Ciudad Inteligente, sino también en el establecimiento de relaciones con las infraestructuras TIC, métricas, políticas e indicadores de semántica de la Ciudad Inteligente.

De las definiciones identificadas en la literatura, se han agrupado en categorías las palabras clave que poseen un sentido similar, como, por ejemplo: Tecnología, TIC y

Sistemas. En la Tabla 2.1 se describen las ocho categorías identificadas, el número de palabras clave que agrupan y el porcentaje relativo de ocurrencia agregado.

Tabla 2.1. Palabras Clave por Categorías

Categorías	Palabras Clave	% ocurrencia
TIC, comunicaciones, inteligencia e información	11	26%
Infraestructura y servicios	9	17%
Medio ambiente y sostenibilidad	7	17%
Las personas, los ciudadanos, la sociedad	9	12%
Gobernanza, gestión y administración	7	10%
Economía y finanzas	4	8%
Calidad y estilo de vida	1	6%
Movilidad	2	4%
Total	50	100%

Puede observarse el papel primordial de las TIC en una Ciudad Inteligente. La sostenibilidad, en aspectos relacionados con las infraestructuras físicas y lógicas, el medio ambiente, la sociedad y la economía. La mejora de la calidad de vida de los ciudadanos, no solamente en aspectos económicos, sino también fomentando el bienestar, la cultura y la cohesión social. La gobernanza, para promover la mejora y la eficiencia de los servicios públicos. Es espíritu de mejora en la ambición de mejorar sus estándares económicos, sociales y medioambientales utilizando para ello las tecnologías de la información y de las comunicaciones.

2.2.1.1 Definición de Ciudad Inteligente

Mediante un ejercicio de síntesis de las categorías identificadas en el apartado anterior se ha construido la siguiente definición de Ciudad Inteligente:

Una Ciudad Inteligente **es** una ciudad justa y equitativa centrada en el ciudadano **que** mejora continuamente su sostenibilidad y resiliencia **aprovechando** el conocimiento y los recursos disponibles, especialmente las Tecnologías de la Información y la Comunicación (TIC), **para** mejorar la calidad de vida, la eficiencia de los servicios urbanos, la innovación y la competitividad sin comprometer las necesidades futuras en aspectos económicos, de gobernanza, sociales y medioambientales.

2.2.2 Atributos de las Ciudades Inteligentes

Los atributos de una Ciudad Inteligente se derivan de la definición e identifican las características propias e inherentes que permiten distinguirla de las demás. De acuerdo con este enfoque, se definen seis ámbitos clave según se muestra en la Figura 2.1: economía, gobernanza, entorno, movilidad, personas y estilo de vida.

El ámbito ***Economía Inteligente (Smart Economy)*** agrupa los atributos relacionados con la capacidad de la ciudad para desarrollarse en términos de empleo, crecimiento económico y financiero.

El ámbito ***Gobernanza Inteligente (Smart Governance)*** agrupa atributos relacionados con la capacidad de administrar eficientemente los recursos, ejecutar políticas y conjugar adecuadamente los intereses de ciudadanos, organizaciones sociales, empresas y administraciones.

El ámbito ***Entorno Inteligente (Smart Environment)*** agrupa atributos relacionados con el medio natural, las estructuras físicas de energía, agua y el urbanismo, así como la capacidad de ser sostenible en su funcionamiento en el presente y en el futuro.

El ámbito ***Movilidad Inteligente (Smart Mobility)*** agrupa atributos relacionados con el transporte y la logística. Por ejemplo, sistemas integrados e interconectados de transporte público eficientes y sostenibles o sistemas de información en tiempo real para los ciudadanos.

El ámbito ***Sociedad Inteligente (Smart People)*** agrupa atributos relacionados con el capital humano de la ciudad, educación, la inclusión social y la colaboración ciudadana.

El ámbito ***Bienestar Inteligente (Smart Living)*** agrupa atributos relacionados con la mejora de la calidad de vida de los ciudadanos y su estilo de vida en aspectos físicos y materiales (salud, seguridad, vivienda, ingresos) y en aspectos sociales (cultura, familia, asociaciones, desarrollo personal) para favorecer la innovación, la educación, la cohesión social y la colaboración ciudadana.



Figura 2.1 Ámbitos clave de una Ciudad Inteligente

2.2.2.1 *Economía Inteligente*

La Economía Inteligente persigue un incremento productivo en la ciudad mediante la eficiencia y la creación de nuevos servicios y modelos de negocio que fomenten la innovación y la eficiencia. Una economía inteligente también implica una interconexión local y global de bienes, servicios y conocimientos.

Una ciudad económicamente inteligente se caracteriza por fomentar la eficiencia, la innovación, la sostenibilidad económica y medioambiental y fomentar nuevos modelos de negocio basados en la economía del conocimiento.

- *Eficiencia.* Una ciudad eficiente y productiva consigue optimizar al máximo sus recursos para conseguir mayores objetivos, o utiliza el menor número de recursos para satisfacer las necesidades de sus ciudadanos.
- *Innovación.* Una ciudad innovadora orienta sus políticas y estrategias a fomentar, coordinar, impulsar y dinamizar el emprendimiento y la innovación para contribuir a la mejora del capital humano y de la economía de la comunidad.
- *Sostenibilidad económica.* Una ciudad sostenible es capaz de satisfacer las necesidades actuales sin comprometer los recursos y posibilidades futuras aplicando políticas orientadas a garantizar el crecimiento preservando el entorno.
- *Nuevos modelos de negocio.* Una ciudad económicamente inteligente fomenta la competitividad mediante la potenciación de los negocios existentes y el fomento de nuevos modelos de negocio basados en la economía digital para aumentar la inteligencia competitiva de su tejido empresarial y financiero y atraer inversión, riqueza externa y capital humano.

2.2.2.2 Gobernanza Inteligente

Por Gobernanza Inteligente se entiende la gestión global de los servicios, integrando gobierno, organizaciones públicas y privadas para que la ciudad pueda funcionar con eficiencia y eficacia como un único organismo. La herramienta principal para lograrlo son las TIC, que permiten una gestión inteligente de los procesos y la interoperabilidad entre actores para compartir información no solamente dentro de la ciudad, sino también con otros actores nacionales e internacionales.

Una ciudad con gobernanza inteligente incluye mecanismos para garantizar un entorno justo y equitativo, transparente, una gestión óptima e integrada de los servicios (Gobierno Electrónico) y la protección de la información entendida como un derecho fundamental de la ciudadanía.

- *Buen Gobierno.* Una ciudad con gobernanza inteligente consigue que ciudadanos, organizaciones sociales, empresas y administraciones trabajen conjuntamente para la generación de valor en la ciudad en términos de competitividad y prosperidad global. Existe un adecuado equilibrio entre mecanismos regulatorios, procesos y políticas para salvaguardar los principios de eficiencia, calidad, colaboración y participación con la formalización de nuevos sistemas, métodos, protocolos y normativas.
- *Transparencia:* Una ciudad transparente utiliza de forma coherente los medios y recursos necesarios para garantizar el derecho a saber y el acceso público a la información, facilitando la reutilización de la misma y la participación de los

ciudadanos. El acceso de los datos públicos es una forma de mejora de la democracia al permitir que los ciudadanos estén mejor informados, y tengan un mayor control sobre las acciones del gobierno.

- *Gobierno Electrónico.* La gestión global de los servicios permite identificar sinergias, optimizar recursos y ofrecer más y mejores servicios a los ciudadanos. Las TIC se configuran como plataforma de captación y procesamiento de información para permitir conocer el funcionamiento de la ciudad en términos de consumo de recursos, servicios y calidad de vida. Las TIC como herramienta de infraestructura lógica para conseguir que una Ciudad Inteligente opere eficientemente, permiten compartir conocimiento y predecir vulnerabilidades. Tecnologías como Open Data, Big-Data, Internet of Things (IoT), accesibilidad, seguridad de la infraestructura son esenciales en una Ciudad Inteligente.
- *Protección de la Información.* Una ciudad segura debe de preservar el derecho fundamental de los ciudadanos a la protección de la información, disponer y decidir sobre las informaciones que se refieran a él y a la confidencialidad de ésta.

2.2.2.3 *Entorno Inteligente*

Por Entorno Inteligente se entiende la gestión eficiente del conjunto de elementos físicos (estructuras) que permiten a los ciudadanos obtener los recursos que necesitan o que permiten la gestión de los mismos dentro de la ciudad. Las estructuras más representativas son: el entorno natural (clima, orografía, recursos naturales, etc.). Las infraestructuras (redes de telecomunicación, el ciclo del agua, la energía, el ciclo de los materiales) y el urbanismo (edificios, manzanas, distritos, etc.).

Un Entorno Inteligente incluye mecanismos para garantizar un desarrollo armónico de las infraestructuras, orientado a satisfacer las necesidades de los ciudadanos, garantiza la preservación del medio ambiente para futuras generaciones y el acceso y disfrute del espacio urbano a la sociedad.

- *Infraestructuras eficientes.* Comprenden los elementos necesarios para la prestación de los servicios y mejora de la vida de los ciudadanos. La gestión eficiente de las infraestructuras, como el ciclo del agua, la energía, las telecomunicaciones, el urbanismo y la vivienda son ejemplos representativos.
- *Sostenibilidad medioambiental.* Una ciudad sostenible es capaz de obtener los recursos necesarios del entorno y mantener los adecuados niveles de calidad medioambiental sin comprometer los recursos y posibilidades futuras. La gestión eficiente de residuos, el control y monitorización del medio ambiente urbano y la utilización de energías limpias son ejemplos representativos.

2.2.2.4 *Movilidad Inteligente*

Por Movilidad Inteligente se entiende la gestión segura, eficiente y sostenible de los sistemas logísticos y de transporte para facilitar a los ciudadanos el acceso, uso y disfrute del espacio urbano, promoviendo la movilidad eficiente de las personas y el acceso a todos los servicios, especialmente a los ciudadanos con discapacidad.

La Movilidad Inteligente prioriza el acceso a opciones de transporte accesible y ecológico, ahorra tiempo a los ciudadanos en sus desplazamientos, reduce costes y disminuye emisiones contaminantes.

- *Infraestructuras viarias inteligentes.* Comprenden los elementos necesarios para la prestación segura y eficiente y accesible de los servicios de movilidad incluyendo la interconexión entre los diferentes sistemas de transporte: autobuses, vehículos, metro, tren, bicicletas o a pie.
- *Transporte y tráfico inteligente.* Comprende la gestión eficiente de los recursos proporcionados por las infraestructuras viarias para minimizar los tiempos de desplazamiento de los usuarios y ofrecer mejores servicios e información en tiempo real a los ciudadanos.
- *Infraestructuras y conectividad TIC.* Las TIC en una Ciudad Inteligente configuran la infraestructura “lógica” de información. Su papel es crucial debido a su capacidad para actuar como una plataforma de captación, manejo de información y gestión global del conocimiento, facilitando a los actores interesados una mejor comprensión sobre el funcionamiento de la ciudad en términos de estilos de vida, servicios y consumo de recursos. Las TIC poseen la capacidad de proporcionar soluciones sostenibles y económicamente viables para las ciudades en multitud de áreas que afectan al entorno (gestión eficiente del agua, residuos, energía, etc.) y a la sociedad (participación ciudadana, open data, etc.) La información obtenida permitirá definir políticas que mejoren la calidad de vida de la sociedad en su conjunto.

2.2.2.5 *Sociedad Inteligente*

Por Sociedad Inteligente se entiende la gestión eficiente de las interacciones que realizan personas, familias, organizaciones, empresas entre sí y con el entorno con el objetivo de potenciar el capital humano y social de la ciudad teniendo como foco al individuo. Estas interacciones configuran las actividades inmateriales que la sociedad realiza y constituyen un activo intangible de la ciudad.

Una Sociedad Inteligente está caracterizada por potenciar la educación y la creatividad de los ciudadanos, el respeto a la pluralidad, la inclusión social de los miembros más desfavorecidos y la participación activa de los ciudadanos en las decisiones que les conciernen.

- *Educación Inteligente:* Entendida como la potenciación de las competencias y

habilidades de los ciudadanos para incrementar el capital humano de la ciudad. Aspectos representativos son la formación en el uso de las herramientas TIC y la formación on-line.

- *Inclusión Social*: Persigue garantizar a los ciudadanos el acceso a los recursos sociales, culturales y económicos independientemente de sus recursos y capacidades para conseguir la integración social plena de los colectivos más vulnerables.
- *Participación Ciudadana*: Es uno de los elementos básicos de la acción municipal y complemento de la democracia representativa. Persigue la participación de los ciudadanos en los asuntos públicos y en la toma de decisiones que les competen. También persigue reforzar el espíritu de ciudadanía, el fomento de asociaciones vecinales, y un mayor conocimiento por parte de la ciudad de las demandas y expectativas de los ciudadanos con el fin de dar respuestas apropiadas a sus necesidades.

2.2.2.6 Bienestar Inteligente

Por Bienestar Inteligente se entiende cómo los ciudadanos o habitantes de una ciudad perciben su propia calidad de vida. Es un concepto amplio y multidimensional que aborda diferentes aspectos relacionados con el individuo, y es el motivo fundamental del crecimiento de las ciudades.

Una Ciudad Inteligente está caracterizada por ser una ciudad deseable para vivir y trabajar que fomenta iniciativas que inciden en el bienestar de los ciudadanos en todos sus aspectos: bienestar ambiental como son el equipamiento urbano, la vivienda o la seguridad, y bienestar social como es el fomento de la cultura, el ocio y el consumo responsable.

- *Bienestar ambiental*: El bienestar ambiental comprende el conjunto de recursos que la Ciudad Inteligente pone a disposición de los ciudadanos para mejorar sus condiciones de vida como son las infraestructuras públicas, el equipamiento urbano, la seguridad física y protección frente a emergencias, la vivienda, el patrimonio histórico, etc.
- *Bienestar social*: El bienestar social comprende el conjunto de recursos que la Ciudad Inteligente pone a disposición de los ciudadanos para mejorar su satisfacción como individuos, familias o colectivos. Salud, ocio, cultura, asociaciones, educación y consumo son ejemplos de este sub-ámbito.

2.2.3 Requisitos de las Ciudades Inteligentes

Mientras que los atributos definen los ámbitos clave de una Ciudad Inteligente, los requisitos son la base para la construcción de indicadores (KPI's) de Ciudad Inteligente que permitan medir el grado de evolución de la ciudad en los ámbitos clave definidos.

Estos requisitos han de ser transversales dado que en general abarcan varios ámbitos clave de manera simultánea y han de permitir su comparación a lo largo del tiempo y con otras ciudades.

Existe una gran cantidad de indicadores disponibles promovidos por diferentes organismos internacionales, nacionales, regionales, locales, académicos y de empresas, parte de ellos reflejados en apartados anteriores. En muchos casos son sectoriales o están ligados a la evaluación del rendimiento de servicios municipales específicos. Casi todos ellos abarcan los ámbitos de economía, medio ambiente y, en cierta medida gobernanza. Los aspectos sociales y los relacionados con la sostenibilidad son tratados de manera más heterogénea. Indicadores relacionados con las TIC normalmente tienen un tratamiento heterogéneo y normalmente están desagregados entre infraestructuras TIC, Servicios TIC, áreas de Gobernanza y Sociedad.

La práctica totalidad de los indicadores analizados entraría en uno de los seis requisitos definidos para Ciudad Inteligente que se proponen. En la Figura 2.2 se describe la relación entre requisitos y atributos de Ciudad Inteligente, y específicamente un requisito relacionado con las TIC por la relevancia que éste posee. Cada uno de estos requisitos, a su vez está formado por varias subcategorías específicas que permitan establecer métricas e indicadores de avance en cada una de las áreas de actuación definidas con grado de detalle deseado.



Figura 2.2 Relación entre Ámbitos y Requisitos de Ciudad Inteligente

Tecnologías de la Información y de las Comunicaciones: recoge todos los indicadores relacionados con el tratamiento de la información, conexión a Internet, proveedores de telefonía, acceso a la red, etc. así como todos aquellos indicadores TIC que son transversales a todos los ámbitos de la ciudad, por ejemplo, la privacidad y la seguridad de las comunicaciones. Los indicadores de infraestructura física de comunicaciones se tratan en los requisitos de Infraestructura física.

Sostenibilidad Medioambiental: Cubre el entorno natural, la energía y las fuentes naturales para garantizar el crecimiento y la búsqueda de maneras más eficientes de utilizar la energía. El índice de sostenibilidad medioambiental está compuesto de cuatro subindicadores: la calidad del aire, agua y suelo, las emisiones de CO₂, la energía y la contaminación.

Productividad: Cubre diferentes aspectos relacionados con el desempeño de las variables económicas. El indicador de productividad está compuesto por 9 indicadores como la inversión de capital, el empleo, la inflación, el comercio, el ahorro, la exportación e importación, los ingresos y consumo en el hogar, la innovación y la economía del conocimiento.

Calidad de Vida: Recoge los aspectos relacionados con el bienestar de la ciudadanía y normalmente es una combinación de cuatro subindicadores: educación, salud, seguridad y espacio público. Respecto a la educación se incluye la alfabetización, grado de acceso a la educación primaria, secundaria y superior. En cuanto a la salud incluye variables de esperanza de vida, las tasas de mortalidad de menores de cinco años y al nivel de nutrición.

Igualdad e Inclusión social: Recoge aspectos relacionados con la igualdad de derechos, justicia e inclusión social. Este índice cubre los aspectos de igualdad social y de género, la transparencia, la participación pública y la gobernanza.

Infraestructura física: Recoge las infraestructuras físicas existentes en la ciudad. Existen dos grupos de indicadores de infraestructura: uno para la infraestructura de servicios comunitarios y el otro para la vivienda. El indicador de infraestructura incluye: conexión a los servicios (agua corriente, alcantarillado, electricidad y TIC), gestión de residuos, infraestructura de conocimientos, infraestructura de salud y transporte y de infraestructura vial. El indicador de vivienda incluye el espacio urbano y las características de habitabilidad.

2.2.4 Semántica de la Ciudad Inteligente

Una ciudad es un sistema en continua evolución con múltiples y complejas iteraciones entre las personas y su entorno cuya gestión aborda aspectos muy diversos y de complejidad creciente. Afrontar estos retos para conseguir una ciudad eficiente, sostenible y que responda a las necesidades de los ciudadanos, refuerza la necesidad de generar soluciones conjuntas y compartir conocimiento.

Definir una semántica de ciudad que permita establecer una estructura común de información que sea válida a lo largo del tiempo, e independiente del tamaño de la ciudad, cultura y peculiaridades, permitirá establecer una sólida base de conocimiento para definir de una manera organizada:

- Métricas e indicadores de ciudad que permitan identificar fortalezas y debilidades.
- Identificar oportunidades y amenazas para poner en marcha estrategias de innovación y anticipación para mejorar la competitividad.
- Priorizar iniciativas de transportación de manera consistente entre necesidades recursos y estrategia.
- Mejorar la comunicación entre los actores proveedores de servicios, los ciudadanos y la administración.
- Canalizar nuevas demandas ciudadanas identificando nuevas necesidades.

La semántica definida en City Protocol [26, 27] proporciona una taxonomía válida para cualquier ciudad, y establece una descripción, identificación, nomenclatura y clasificación jerárquica de los sistemas, subsistemas e iteraciones que existen entre ellos. Se ha diseñado como una herramienta común que puede aplicarse para facilitar iniciativas relacionadas con aspectos claves de la organización de las ciudades – gobernanza, evaluación y transformación -. Anatomía de Ciudad (City Anatomy) podría ser adoptada no solamente en las iniciativas de evaluación y estandarización como una descripción normalizada de los sistemas e iteraciones en la ciudad, sino también en la descripción de nuevos procesos y exportación de iniciativas a otras ciudades.

La Figura 2.3 describe con detalle la arquitectura propuesta como una integración holística de tres niveles que configuran el ecosistema ciudad: la infraestructura física (estructuras), la gente que ocupa y utiliza el espacio físico (sociedad), y las interacciones entre ambos.

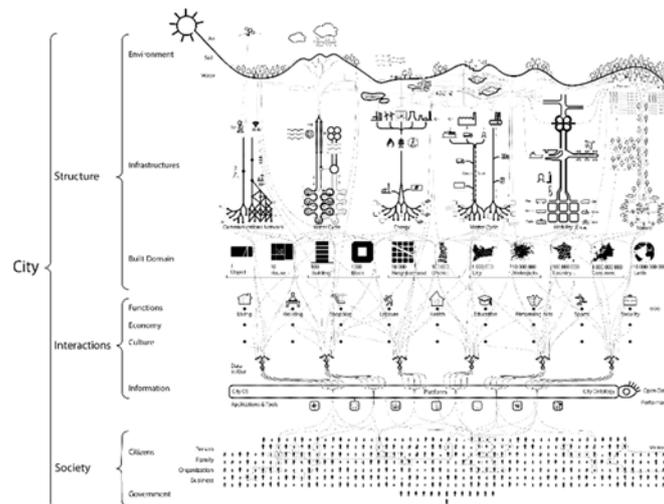


Figura 2.3 Anatomía de Ciudad © City Protocol Society (2014) All Rights Reserved

2.2.4.1 Estructuras

El sistema estructuras engloba a los elementos físicos no humanos existentes en la ciudad, y que a su vez se subdivide en tres subsistemas: entorno, infraestructuras y territorio, cada uno de ellos con sus propios indicadores.

El entorno, describe las condiciones naturales existentes en la ciudad y donde ésta se desarrolla. Está compuesto por la naturaleza: animales y plantas más los tres elementos básicos de la biodiversidad: aire, tierra y agua.

Las infraestructuras engloban al conjunto de elementos físicos que permiten a los ciudadanos obtener los recursos que necesitan o que permiten su gestión dentro de la ciudad. Las infraestructuras más representativas son las redes de telecomunicación, el ciclo del agua, la energía, el ciclo de los materiales, la movilidad y la naturaleza entendida como información sobre los elementos vivos no humanos a todas las escalas.

El territorio define las estructuras públicas y privadas donde los ciudadanos viven (casas, edificio, manzana, barrio, etc.) y el espacio público que las rodea (calles, plazas)

Estos tres subsistemas permiten establecer una relación entre elementos físicos con redes de información ayudando a entender la ciudad como un conjunto de sistemas e interacciones que procesan y almacenan información.

En la Tabla 2.2 especifican los elementos que componen la Estructura Física.

Tabla 2.2. Componentes del Sistema Estructura Física

Estructura																
Entorno					Infraestructuras					Territorio						
Asentamiento	Biodiversidad	Aire	Suelo	Agua	Comunicaciones	Ciclo del Agua	Energía	Ciclo de la Materia	Movilidad	Naturaleza	Vivienda	Edificios	Barrios	Ciudad	Espacio Público	Espacio Urbanizable

2.2.4.2 Interacciones

Las interacciones son las actividades que la sociedad efectúa con las estructuras y que pueden analizarse y medirse como flujos de información. Las interacciones podrían asimilarse a las funciones metabólicas del cuerpo humano, y que está compuesto por cuatro capas: Las funciones urbanas que incluyen educación, trabajo, salud, artes, etc, La economía que incluye las actividades de innovación, y funcionamiento de la ciudad. La cultura, donde se incluyen los activos no materiales de la ciudad, valores, tradiciones, etc.

y por último la información, que incluye indicadores de prestaciones de la ciudad, procesos y modelos de funcionamiento, así como su gestión y apertura a terceros.

En la Tabla 2.3 se especifican los elementos que componen el Sistema Interacciones.

Tabla 2.3. Componentes del Sistema Interacciones

Interacciones																				
Funciones							Economía				Cultura	Información								
Vida	Trabajo	Comercio	Ocio	Salud	Educación	Cultura	Deportes	Seguridad	Producción de riqueza	Distribución de riqueza	Comercio	Inversiones	Competitividad	Iniciativa Empresarial	Actividades	Equipamientos	Aplicaciones	Open Data	Interoperabilidad	Resiliencia

2.2.4.3 Sociedad

El sistema *Sociedad* se refiere a los individuos y a la manera en la que éstos se organizan y relacionan, así como a su gobierno.

Se definen dos subsistemas: ciudadanos y gobierno. El subsistema ciudadanos engloba a las personas en sentido amplio como individuos: ciudadanos o visitantes y su estructura social: familias, organizaciones privadas, negocios, etc. El subsistema gobierno se refiere a la estructura político -administrativa de la ciudad y es el responsable de las políticas y regulaciones que facilitan el desarrollo de la ciudad. En este caso el término gobernanza se utiliza para propósitos de evaluación de la efectividad del gobierno.

En la Tabla 2.4 se especifican los elementos que componen el Sistema Sociedad

Tabla 2.4. Componentes del Sistema Sociedad

Sociedad	
Ciudadanía	Administración
Familia	Liderazgo
Organizaciones	Estrategias
Negocios	Leyes y regulaciones
Participación ciudadana	Impulso social

2.3 Arquitectura TIC en una Ciudad Inteligente

La arquitectura TIC es un conjunto complejo de componentes, bloques funcionales, hardware, redes, servicios y aplicaciones que interactúan entre sí y cuya influencia es capital para el correcto desarrollo de la ciudad Inteligente. Dentro de la Anatomía de la Ciudad anteriormente descrita, las redes de telecomunicación y el hardware pertenecen a la infraestructura física, pero las aplicaciones y los servicios TIC son la base del correcto funcionamiento del resto de sistemas, Sociedad y Relaciones. Esta arquitectura ha de cubrir áreas geográficas diferentes, utilizar múltiples tecnologías, ser válida para diferentes tipos de ciudades y operar a lo largo del tiempo.

La evolución tecnológica, la introducción paulatina de las redes convergentes y los nuevos paradigmas como el Internet de las Cosas (IoT) o los Servicios Cloud, requieren una arquitectura TIC que sea capaz de cubrir la totalidad de las necesidades presentes y futuras necesarias para conseguir el objetivo de Ciudad Inteligente.

Los principios básicos que ha de cumplir la arquitectura TIC es que esté basada en estándares, sea flexible, escalable y tolerante a fallos. También ha de proveer seguridad, protección de la información (privacidad) y un conjunto de servicios avanzados. Existen diferentes visiones para abordar cómo describir, identificar y clasificar los sistemas y subsistemas TIC y sus relaciones, en general orientadas a describir cómo se resuelven necesidades específicas a través de la arquitectura TIC.

- *La visión funcional:* se enfoca hacia aspectos relacionados con actividades específicas de la Ciudad y cómo estas actividades utilizan la arquitectura TIC. Por ejemplo, la gestión inteligente de la energía o la movilidad.
- *La visión de la implementación:* se enfoca hacia cómo se implementa, opera, administra y mantiene la arquitectura. También aplica a la descripción de los servicios, seguridad lógica, fabricantes, gestión de la información y usabilidad de la arquitectura por parte de los ciudadanos.
- *La visión física:* se enfoca en la ubicación, tipo, potencia, ancho de banda, almacenamiento y demás características de los elementos de la arquitectura tanto de computación como de comunicaciones.
- *La visión de negocio:* se enfoca hacia la sostenibilidad económica de la arquitectura TIC en base a inversiones y retornos para la ciudad.

Representar estos aspectos en una arquitectura unificada TIC es complejo por la gran variedad y heterogeneidad de los componentes que la constituyen y la dificultad de su desarrollo e implementación. Todos estos componentes necesitan ser modelados para garantizar que la arquitectura TIC de una Ciudad Inteligente sea completa, lógica y que cumpla los objetivos de negocio requeridos.

En la Figura 2.4 se propone una arquitectura funcional basada en capas para facilitar el establecimiento de una clasificación jerárquica de los diferentes elementos de la estructura TIC y sus relaciones. Se ha definido un modelo funcional donde cada capa posee sus propias funciones y establece relaciones con las capas adyacentes o con el entorno exterior a través de un modelo cliente-proveedor.

Este modelo permite a los usuarios: personas, máquinas inteligentes y procesos, establecer relaciones de comunicación entre sí (comunicaciones humano – humano, humano – máquina y máquina – máquina), con la infraestructura física no TIC formada por los elementos físicos susceptibles de ser gestionados por la infraestructura TIC y finalmente, el “medio natural” que define las características físicas propias de la ciudad.

A continuación, se describe con mayor detalle la funcionalidad de cada capa.

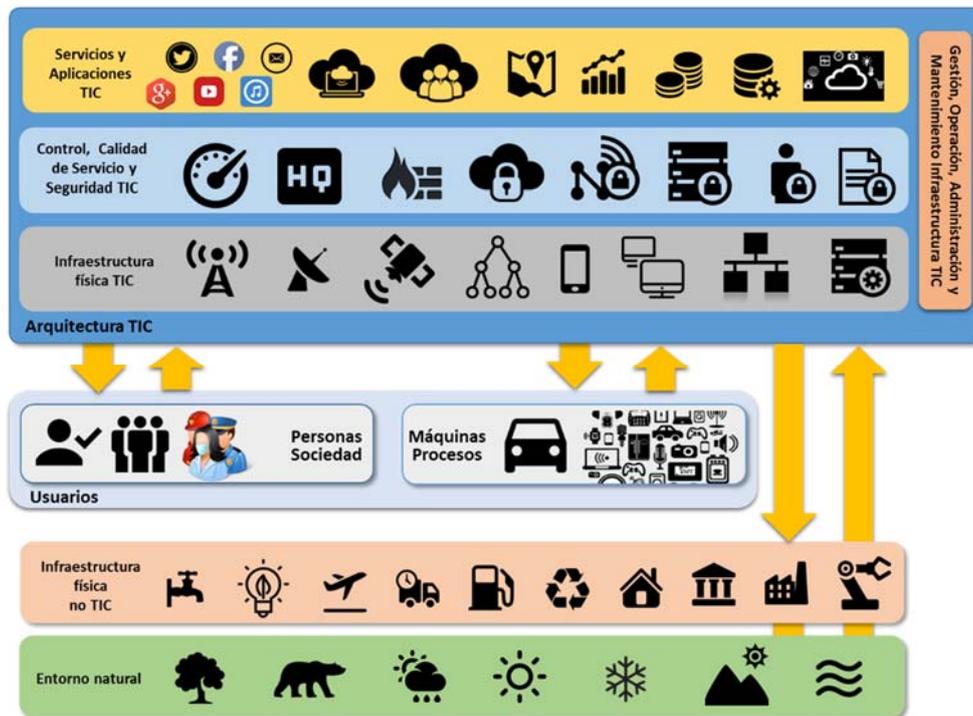


Figura 2.4 Arquitectura TIC de una Ciudad Inteligente

Entorno natural. Se refiere a los aspectos medioambientales que caracterizan la ciudad y que son relevantes para el despliegue de las infraestructuras físicas. Orografía y climatología son ejemplos de esta capa.

Infraestructuras físicas no TIC. Esta capa contiene todos los elementos físicos que se utilizan para el funcionamiento de la ciudad, que permiten a los ciudadanos obtener los recursos que necesitan y que pueden ser gestionados por la infraestructura TIC. Edificios, carreteras, suministro de energía, agua o gestión de residuos son ejemplos de esta capa.

Usuarios. Esta capa representa al elemento más relevante de la arquitectura TIC de la Ciudad Inteligente: el individuo, las máquinas-procesos y sus relaciones tanto con otras personas como con el entorno. Esta capa utiliza los recursos necesarios de la estructura TIC para conseguir los objetivos perseguidos. A través de interfaces normalizados se pueden definir modelos de relación como, por ejemplo, las comunicaciones entre humanos (telefonía, redes sociales), de humanos con el entorno (acceso a bases de datos, lectura de sensores), y máquinas entre sí (sistemas automáticos de adquisición de datos y control).

Estructura TIC. Contiene la totalidad de elementos que proporcionan la “inteligencia” necesaria a las infraestructuras físicas no TIC y que permiten la relación con personas y aplicaciones en términos de comunicación HTC + MTC y sus posibles combinaciones.

- **Infraestructuras físicas TIC.** Esta capa contiene todos los elementos físicos que se han construido para interactuar con las infraestructuras físicas TIC y no TIC y facilitar su operación, administración y gestión. Servidores, redes, terminales, sensores etc. son ejemplos de esta capa. En la Figura 2.5 se especifica hasta el primer nivel, los elementos que componen esta capa.

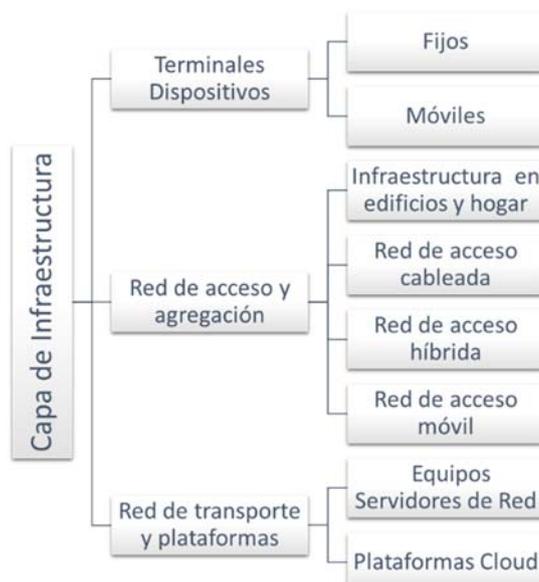


Figura 2.5 Taxonomía de la capa Infraestructuras TIC

- Control, Calidad de Servicio y Seguridad TIC.** Esta capa contiene los mecanismos que permiten el correcto funcionamiento de los servicios y aplicaciones de Ciudad Inteligente y que se ofrecen a la sociedad o a la infraestructura física (TIC y no TIC). Esta capa proporciona el control del uso de recursos de la infraestructura, garantiza la calidad de las comunicaciones, la movilidad y la seguridad de las mismas (seguridad de la infraestructura física, seguridad de la infraestructura lógica, control de acceso a los recursos y privacidad de la información). En la Figura 2.6 se especifica hasta el primer nivel, los elementos que componen esta capa.

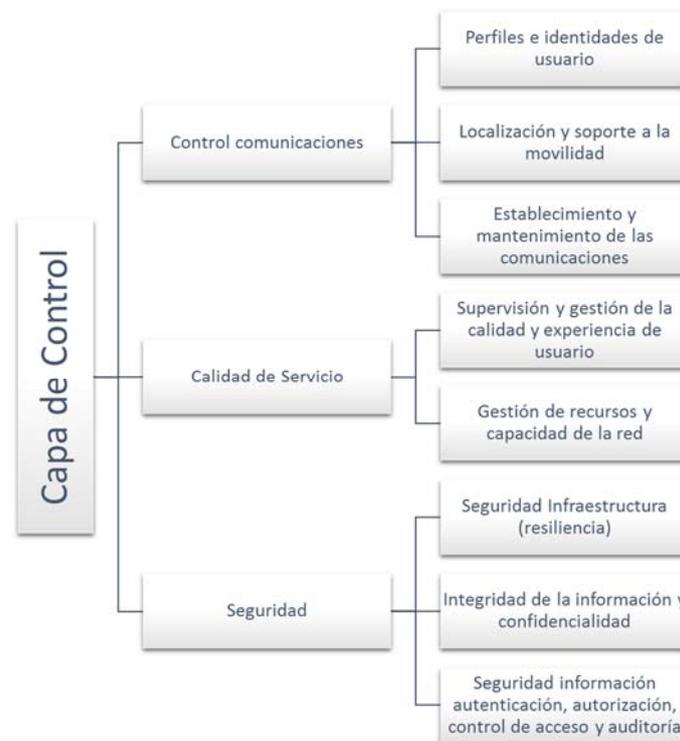


Figura 2.6 Taxonomía de la capa Control, Calidad de Servicio y Seguridad TIC

- Servicios y Aplicaciones TIC.** Esta capa contiene los servicios y aplicaciones de Ciudad Inteligente que se ofrecen a la sociedad o a la infraestructura física. Estos servicios están soportados por la infraestructura TIC y son piezas básicas para construir las aplicaciones de Ciudad Inteligente. Servicios de transporte y movilidad, administración electrónica, negocio electrónico, seguridad y

emergencia, salud inteligente, turismo, educación, edificios/urbanismo inteligente, gestión de residuos, energía y agua son ejemplos de esta capa. En la Figura 2.7 se especifica hasta el primer nivel, los elementos que componen esta capa.

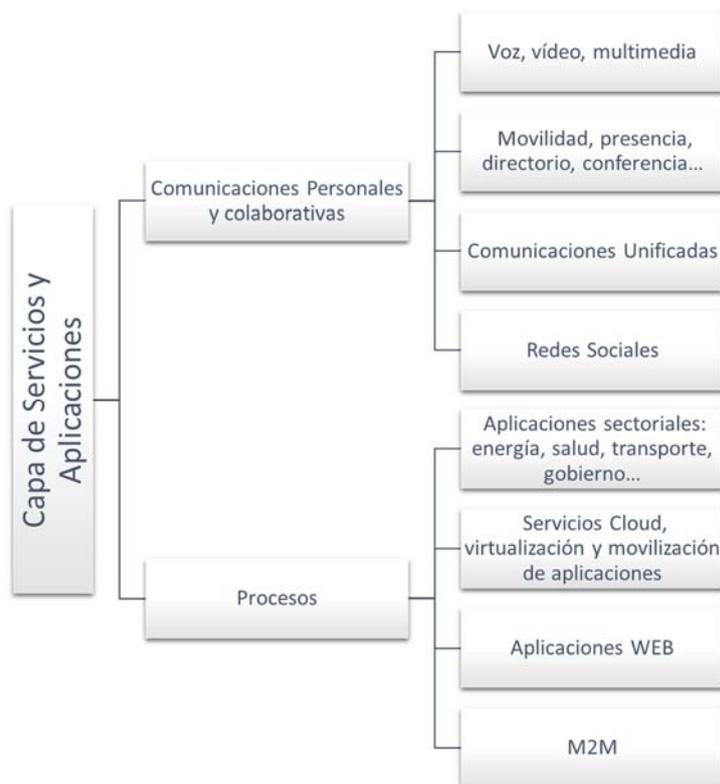


Figura 2.7 Taxonomía de la Capa Servicios y Aplicaciones TIC

Esta visión transversal de la capa de Servicios y Aplicaciones ha de ser necesariamente complementada por una visión “sectorial” que refleje agregaciones de aplicaciones por categorías, por ejemplo, seguridad y emergencias, movilidad, gestión de residuos, etc. Esta visión sectorial es necesaria dado que normalmente refleja organizaciones funcionalmente separadas y flujos de información heterogéneos. La Figura 2.8 se muestra esta visión sectorial de la Capa de Aplicaciones.

- **Gestión, Operación, Administración y Mantenimiento** de la Infraestructura TIC es el elemento transversal a las capas TIC. Comprende el conjunto de recursos necesarios para garantizar la disponibilidad, interoperabilidad, adecuada

atención a las incidencias y la correcta configuración de los elementos que componen la infraestructura.

Por último, una visión de la arquitectura desde la perspectiva física también es necesaria para identificar los componentes existentes y su relación entre ellos. En la Figura 2.9 se refleja este enfoque, más cercano a los operadores de redes e infraestructuras de telecomunicación.

Esta arquitectura desarrolla la taxonomía de las figuras 2.5, 2.6 y 2.7 mediante un modelo de capas (ver Figura 2.9):



Figura 2.8 Visión sectorial de la capa de aplicaciones (ITU-T Y.Supl. 30(16)_F03)

- **Capa de captación:** Esta capa permite la conexión de un nodo terminal a una red capilar. Los terminales (sensor, transductor, actuador, cámara, lector RFID, símbolos de código de barras, rastreador GPS, etc.) son capaces de detectar el mundo físico. Proporcionan la habilidad y la inteligencia de la "detección del ambiente" para supervisar y controlar la infraestructura física dentro de la ciudad. La red capilar (incluyendo SCADA, red de sensores, WPAN, video vigilancia, RFID, redes relacionadas con GPS, etc.) conecta varios terminales a la capa de red, proporcionando la información del mundo físico a través de los

terminales.

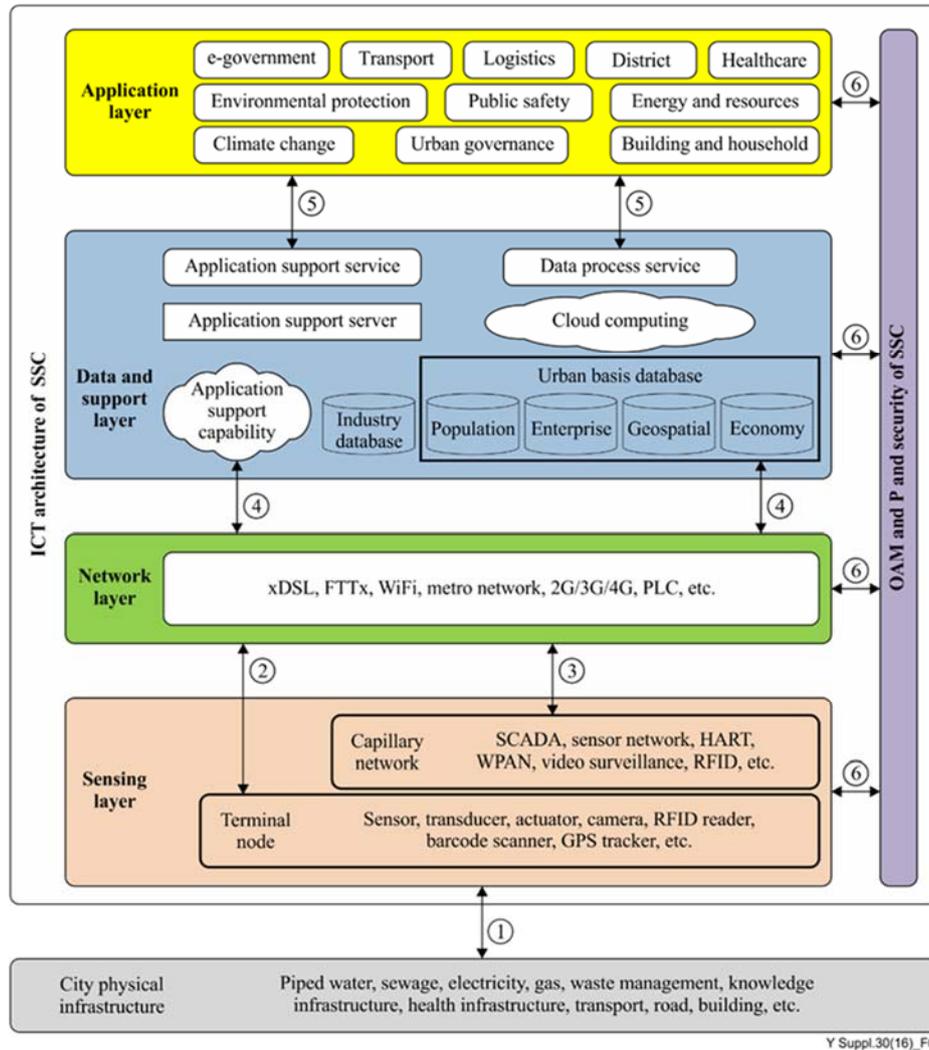


Figura 2.9 Modelo de capas de la Arquitectura TIC (ITU-T Y. Suppl.30(16)_F04)

- Capa de red: incluye las redes proporcionadas por los operadores de telecomunicaciones, así como otras redes de entorno metropolitano construidas por la propia administración de la ciudad para autoconsumo. Su propósito principal es asegurar las capacidades de soporte de varias aplicaciones y servicios a nivel de ciudad.
- Capa de datos y soporte: contiene los repositorios de datos e información

necesarios para el consumo de capas superiores y está compuesta por los centros de datos y almacenamiento, local o en la nube.

- **Capa de aplicación:** Esta capa incluye las aplicaciones que administran y entregan los servicios de Ciudad Inteligente.

En esta capa pueden a su vez distinguirse:

- *Capa de usuarios:* organiza a los usuarios finales del servicio en grupos tanto desde el lado de la demanda como de la oferta.
 - *Capa de presentación:* contiene las interfaces de usuario (web, aplicaciones, comandos de voz, etc.) que se encuentran entre los usuarios finales y los servicios de ciudad.
 - *Capas de aplicaciones software:* Contiene todas las aplicaciones de software correspondientes que realizan los servicios de ciudad.
 - *Capa de negocio:* Consiste en los procesos de negocio, que se encuentran detrás de cada ejecución del servicio de la ciudad.
- **Capa de Operación, Supervisión, Mantenimiento, Protección y Seguridad:** Esta capa proporciona el funcionamiento, la administración, el mantenimiento, el aprovisionamiento y la función de seguridad para los sistemas TIC de la ciudad.

2.3.1 *Capa de captación*

Esta capa está compuesta por terminales, sensores, transductores, actuadores y dispositivos que interconectan el mundo físico, personas o infraestructuras, con la capa superior (red de acceso) y entre sí, proporcionando conectividad uni o bidireccional HTC / MTC.

Existe una gran variedad de dispositivos de todo tipo, alámbricos, inalámbricos; de seguridad, iluminación, presencia, tiempo, transporte, movimiento o posición instalados en la infraestructura física. Estos dispositivos que combinan sensores, computación y comunicación permiten conocer y estudiar variables tan heterogéneas como el clima, la congestión de las carreteras, la contaminación atmosférica, o los asistentes a eventos deportivos, culturales y de entretenimiento.

En esta capa se incluye la llamada “red capilar”, arquitecturas ad-hoc, redes de sensores, etc. que persiguen mejorar la eficiencia, los costes de transmisión o el consumo de energía agrupando dispositivos y ofreciendo un único interfaz a la red de acceso.

2.3.1.1 *Arquitectura de las redes de sensores inalámbricos*

El coste de las comunicaciones y del hardware WAN específico junto con la limitación del consumo de energía, ha sido una de las causas de la proliferación de protocolos avanzados de redes malladas donde los dispositivos se comunican entre sí formando una red de conectividad que se conecta con la red de acceso a través de pasarelas (Gateways) específicos.

Esta conectividad mallada permite construir soluciones más robustas y simplificar el hardware. Aunque las capacidades de cualquier dispositivo son mínimas, la composición de cientos de dispositivos ofrece nuevas posibilidades tecnológicas. La potencia de las redes de sensores inalámbricos reside en la capacidad de desplegar un gran número de nodos pequeños que se montan y configuran a sí mismos. Los escenarios de uso de estos dispositivos van desde el seguimiento en tiempo real o la monitorización de las condiciones ambientales, de la salud de las estructuras o equipos, entre otros. Aunque a menudo se denomina redes de sensores inalámbricos, también pueden controlar actuadores que extienden el control al mundo físico.

El concepto de redes de sensores inalámbricos se basa en una simple ecuación:

$$\text{Sensor} + \text{CPU} + \text{interfaz radio} = \text{miles de aplicaciones potenciales}$$

Sin embargo, en realidad la combinación de sensores, radios y CPU en una red de sensores inalámbricos requiere una comprensión detallada de las capacidades y limitaciones de cada uno de los componentes de hardware subyacentes, así como una comprensión detallada de las tecnologías de redes modernas y la teoría de sistemas distribuidos. Cada nodo individual debe estar diseñado para proporcionar el conjunto de primitivas necesarias para sintetizar la red interconectada que surgirá a medida que se desplieguen, cumpliendo con los estrictos requisitos de tamaño, coste y consumo de energía. Para hacer realidad la visión de la red de sensores inalámbricos, se debe desarrollar una arquitectura que sintetice las aplicaciones previstas de las capacidades de hardware subyacentes.

2.3.1.2 Clases de aplicaciones de red de sensores

Las tres clases de aplicación que se consideran más relevantes son: recopilación de datos ambientales, monitorización de seguridad y seguimiento. Se cree que la mayoría de las implementaciones de redes de sensores inalámbricos caerán dentro de una de estas categorías.

2.3.1.2.1 Recopilación de datos ambientales

Una aplicación de recolección de datos ambientales consiste en obtener varias lecturas de sensores de un conjunto de puntos en un entorno durante un período de tiempo con el fin de detectar tendencias e interdependencias. La idea es recopilar datos de cientos de puntos distribuidos por toda la zona y luego analizar los datos en servidores externos.

A nivel de red, la aplicación de recogida de datos medioambientales se caracteriza por tener un gran número de nodos que detectan y transmiten continuamente datos a un conjunto de estaciones base que almacenan los datos utilizando métodos tradicionales. Estas redes generalmente requieren velocidades de datos muy bajas y una vida útil extremadamente larga. En el escenario de uso típico, los nodos se distribuirán uniformemente en un entorno externo. Esta distancia entre nodos adyacentes será mínima, pero la distancia a través de toda la red será significativa (ver Figura 2.10).

Las aplicaciones de recolección de datos ambientales usan típicamente topologías de

enrutamiento basadas en árboles en las que cada árbol de enrutamiento está conectado con nodos de alta capacidad que reciben los datos. Los datos se transmiten periódicamente de los nodos secundarios a los nodos primarios hasta que llegue a la puerta de enlace (sumidero). Con este modelo, cada nodo es responsable de reenviar los datos de todos sus descendientes. Una vez configurada la red, cada nodo muestrea periódicamente sus sensores y transmite sus datos por el árbol de enrutamiento y de vuelta a la estación base.

Para muchos escenarios, el intervalo entre estas transmisiones puede ser del orden de minutos. Periodos típicos de reporte pueden variar entre 1 minuto y varias horas o días dado que los parámetros ambientales típicos que se están supervisando, como la temperatura, la intensidad de la luz y la humedad, no cambian lo suficientemente rápido como para requerir mayores tasas de notificación.

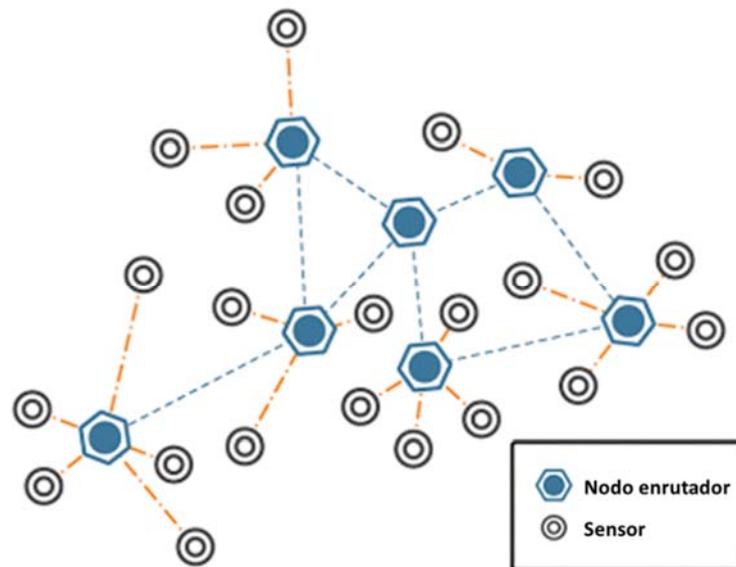


Figura 2.10 Topología típica de una red de sensores para captación de datos

Además de grandes intervalos de muestreo, las aplicaciones de monitorización ambiental no tienen requisitos de latencia estrictos. Las muestras de datos pueden retrasarse dentro de la red durante períodos de tiempo moderados sin afectar significativamente el rendimiento de la aplicación. En general, los datos se recogen para el análisis futuro, no para la operación en tiempo real.

Las características más importantes de los requisitos de monitorización ambiental son larga vida útil, sincronización precisa, bajas velocidades de datos y topologías relativamente estáticas. Además, no es esencial que los datos se transmitan en tiempo real de

vuelta al punto de recogida central. Las transmisiones de datos pueden retrasarse dentro de la red siempre que sea necesario para mejorar la eficiencia de la red.

2.3.1.3 *Monitorización de la seguridad*

Las redes de monitorización de seguridad están compuestas de nodos que se colocan en ubicaciones fijas en un entorno que monitoriza continuamente uno o más sensores para detectar una anomalía. Una diferencia clave entre la monitorización de seguridad y la monitorización ambiental es que las redes de seguridad no están recopilando datos. Esto tiene un impacto significativo en la arquitectura. Cada nodo tiene que comprobar frecuentemente el estado de sus sensores, pero sólo tiene que transmitir un informe de datos cuando hay una violación de seguridad. La comunicación inmediata y confiable de los mensajes de alarma es el requisito fundamental del sistema. Además, es esencial confirmar que cada nodo está todavía presente y funcionando. Para las aplicaciones de monitorización de seguridad, la red debe configurarse de manera que los nodos sean responsables de confirmar el estado de los demás. La topología óptima de una red de monitorización de seguridad será bastante diferente de la de una red de recopilación de datos.

Es razonable asumir que cada sensor debe ser revisado en función de la aplicación para que el coste energético de realizar esta comprobación sea mínimo. La mayoría del consumo de energía en una red de seguridad se gasta en cumplir con los estrictos requisitos de latencia asociados con la señalización de la alarma cuando se produce una violación de seguridad.

Una vez detectada, una infracción de seguridad debe comunicarse inmediatamente a la estación base. La latencia de la comunicación de datos a través de la red con la estación base tiene un impacto crítico en el rendimiento de la aplicación. Los usuarios exigen que las situaciones de alarma se denuncien en cuestión de segundos después de la detección. Esto significa que los nodos de la red deben ser capaces de responder rápidamente a las peticiones de sus vecinos para enviar datos.

Adicionalmente, la información sobre la incidencia se convierte en un elemento crítico para la toma de decisiones. En este caso, imágenes o vídeo permiten identificar la naturaleza de la incidencia y servir como herramienta de decisión para ejecutar las acciones más adecuadas a la naturaleza de la incidencia. A diferencia de los sistemas de recuperación de datos ambientales, la arquitectura árbol-rama puede no ser la más adecuada por el importante volumen de información a intercambiar entre nodos y por el tiempo de propagación hasta el nodo raíz.

Una primera aproximación pueden ser sensores 3G autónomos equipados con cámaras de vídeo que permitan el desarrollo de nuevas aplicaciones de seguridad y vigilancia, pero no cumplen algunos requerimientos como la supervisión mutua y el consumo de energía.

2.3.1.3.1 Escenarios de seguimiento de objetos

Esto indica el seguimiento de un objeto marcado a través de una región de espacio supervisada por una red de sensores. Hay muchas situaciones donde es necesario rastrear la ubicación de activos o personal valioso. Los sistemas actuales de control de inventario intentan rastrear los objetos registrando el último punto de control que pasó un objeto. Con redes de sensores inalámbricos, los objetos pueden rastrearse simplemente marcándolos con un sensor. Este sensor será rastreado a medida que se mueva a través de una red que se despliegue en el entorno definido. En lugar de detectar datos ambientales, estos nodos se desplegarán para detectar los mensajes de radio de los nodos conectados a varios objetos. Los nodos se pueden utilizar como etiquetas activas que anuncian la presencia, y en muchos casos la ubicación de un dispositivo. Se puede utilizar una base de datos para registrar la ubicación de objetos rastreados en relación con el conjunto de nodos en ubicaciones conocidas. Con este sistema se hace posible solicitar la ubicación exacta del objeto y no sólo la última ubicación donde fue escaneado por última vez.

Hay dos maneras principales de realizar el seguimiento de localización cuando los dispositivos sensores se encuentran en una gran área como una ciudad. El más extenso es utilizar un módulo GPS para obtener la información enviada por los satélites y extraer toda la información posible (latitud, longitud, velocidad, dirección). Sin embargo, esta metodología no es efectiva cuando se requieren escenarios móviles en los que los nodos pueden cambiar de un entorno exterior a uno interior, como ir a edificios, garajes y túneles.

Para estos casos puede utilizarse, si el dispositivo posee conectividad con una red móvil un método alternativo que consiste en tomar la información enviada por las celdas de la red móvil y buscar su ubicación en una base de datos previamente guardada.

Tanto las tecnologías GPS como GPRS/HxxPA/3G/4G son complementarias entre sí como un sistema en el que ambas tecnologías permiten el seguimiento de la posición dentro de los edificios, garajes e incluso dentro de los túneles (por ejemplo, el sistema de ferrocarril subterráneo), manteniendo la exactitud de la información de los satélites GPS alcance el dispositivo sensor. Hay muchas tecnologías para obtener información de localización.

2.3.1.3.2 Redes híbridas

Los escenarios reales generalmente contienen aspectos de las tres categorías. Por ejemplo, en una red de sensores diseñada para rastrear vehículos la red puede conmutar entre ser una red de vigilancia de alarmas y una red de recogida de datos. Durante un largo período de inactividad cuando no hay vehículos presentes, la red realizará simplemente una función de monitorización de alarmas. Cada nodo monitoreará sus sensores mientras espera detectar un vehículo. Una vez que se detecta un evento de alarma, toda o parte de la red cambiará a un modo de red de recolección de datos y reportará periódicamente las lecturas del sensor a una estación base, en la que estaría en curso el seguimiento de los vehículos. Como resultado de este comportamiento de la red multimodal, es importante

desarrollar una única arquitectura que pueda manejar los tres escenarios de aplicación antes mencionados.

2.3.1.4 Infraestructura de la red radio

El subsistema de radio es el sistema más importante en un sensor inalámbrico, ya que es el principal consumidor de energía en los tres escenarios de aplicación destacados. Los modernos transceptores de corto alcance de bajo consumo consumen entre 15 y 300 milivatios de potencia al enviar y recibir y consumen aproximadamente la misma cantidad de energía cuando están en modo de recepción o transmisión. Esta energía se consume si la radio está encendida, esté o no recibiendo datos reales.

2.3.1.4.1 Rango de transmisión

El alcance de transmisión de un sistema inalámbrico es controlado por varios factores clave. El factor más intuitivo es el de la potencia de transmisión, cuanto más energía mayor alcance, pero la relación entre la potencia de salida y la cobertura es un polinomio de grado 3 y 4 (sin propagación en la línea de visión directa). Así que, para transmitir al doble de distancia en un entorno interior, se necesita emitir de 8 a 16 veces más energía.

Otros factores en la determinación del alcance incluyen la sensibilidad del receptor, la ganancia, la eficiencia de la antena y el mecanismo de codificación del canal. En general, los nodos de red de sensores inalámbricos no pueden explotar antenas direccionales de alta ganancia porque requieren una alineación especial y no son útiles en las topologías de red ad hoc. Las antenas omnidireccionales permiten que los nodos se comuniquen eficazmente en todas las direcciones.

Las sensibilidades típicas del receptor están entre -85 y -110 dBm. Los aumentos de la cobertura de transmisión pueden alcanzarse aumentando la sensibilidad o aumentando la potencia de transmisión. Al transmitir a 0 dBm, una sensibilidad del receptor de -85 dBm resultará en un rango de espacio libre al aire libre de 25-50 metros, mientras que una sensibilidad de -110 dBm resultará en un rango de 100 a 200 metros. El uso de una radio con una sensibilidad de -100 dBm en lugar de una radio con -85 dBm le permitirá disminuir la potencia de transmisión en un factor de 30 pero a costa de encarecer el hardware y disminuir la vida de las baterías si el terminal es autónomo.

2.3.1.4.2 Velocidades binarias

A diferencia de muchas redes de datos, las redes de sensores inalámbricos no requieren velocidades binarias elevadas, 10-100 Kbps de ancho de banda de red es suficiente para muchas aplicaciones. El consumo de ancho de banda tiene un impacto más significativo en el consumo de energía del nodo y su vida útil en caso de un nodo con batería. En estos casos es esencial que la radio permanezca apagada tanto como sea posible. Al aumentar la velocidad de bits sin aumentar la cantidad de datos que se transmiten, el ciclo de trabajo de radio se reduce.

Durante los últimos años se ha experimentado una explosión en la tecnología de los

sensores que han permitido un aumento vertiginoso tanto en su capacidad de proceso como de comunicación y autonomía. En la Tabla 2.5 se muestra un ejemplo de varias plataformas.

2.3.1.5 Interfaces

Los interfaces representan la descripción de los mecanismos que permiten a la red actuar con los sensores, generalmente se dividen en sensores analógicos y digitales. Los sensores analógicos proporcionan generalmente una tensión analógica que corresponde a los fenómenos físicos que están midiendo. Generalmente, éstos producen una forma de onda continua que debe ser digitalizada y luego analizada. Los sensores analógicos a menudo requieren calibración externa y linealización porque suele ser común que el sensor tenga una respuesta no lineal a los estímulos. El controlador debe entonces compensar para producir una lectura en unidades significativas. Dependiendo de las características del sensor, éste puede ser un proceso complejo. En muchos casos, la traducción puede depender de otros factores externos tales como temperatura, presión o tensión de entrada. Una segunda dificultad en la interfaz con sensores analógicos es la de escala. Cada sensor tendrá diferentes escalas de tiempo y voltaje. El voltaje de salida contendrá generalmente un offset DC combinado con una señal que varía en el tiempo. Dependiendo de la relación entre la señal y el componente de continua, puede ser necesario un conjunto de amplificadores y filtros para que coincida con la salida del sensor al rango y fidelidad del ADC que se utiliza para capturarlo.

Se han desarrollado sensores digitales para eliminar muchas de estas dificultades. Internamente contienen un sensor analógico, pero proporcionan una interfaz digital. Toda la compensación y la linealización requeridas se realizan internamente. La salida es una lectura digital en una escala apropiada. La interfaz con estos sensores es a través de uno de un puñado de protocolos estándar de comunicación chip a chip.

2.3.1.6 Subcapa de red capilar

Además de la naturaleza de los sensores que se ha descrito en párrafos anteriores, su conexión directa a la red de acceso implica en general costes elevados y un consumo de energía inasumible para este tipo de dispositivos.

Para paliar esta ineficiencia se ha definido dentro de esta capa una subcapa llamada “red capilar” que permite la interconexión de dispositivos entre sí a cortas distancias y proveen funciones de red en este ámbito. Existen múltiples tecnologías de conectividad y cobertura generalmente limitada donde uno o más nodos (Gateways) proveen la conectividad con la red de acceso. Zigbee, Bluetooth, WI-Fi LP son ejemplos de estas tecnologías. En la Tabla 2.5 se describen las características básicas de soluciones baja capacidad.

2.3.1.6.1 ZigBee (IEEE 802.15.4)

ZigBee es un estándar mundial de conectividad inalámbrica, centrado en la estandarización y la interoperabilidad de dispositivos heterogéneos. Es un estándar de comunicaciones para corto alcance, bajo coste y mínimo consumo de batería.

Tabla 2.5. Comparativa entre estándares de conectividad inalámbricos

Standard	ZigBee 802.15.4.	Bluetooth 802.15.1 (v5.0)	Wi-Fi 802.11b/n/ac	GPRS/GSM LTE/4G
Aplicación	Monitorización y Control	Sustituye cableado con periféricos	Web, vídeo e-mail	WAN, voz y datos
Duración baterías (días)	100-1.000+	1-7	0.1-5	1-7
Nodos por subred	256/65k+	7+	30+	10.000+
Ancho de banda	20-250 kb/s	4 Mb/s	900 Mb/s	64-128
Rango (m)	1-75+	1-10+	1-100	1 000+
Atributos clave	Económico, bajo consumo	Económico, corto alcance	Velocidad y flexibilidad	Calidad, alcance

ZigBee se basa en una capa física de radio (PHY) y de control de acceso al medio (MAC) definidas por el estándar IEEE 802.15.4. En contraste con estándares como Bluetooth y Wi-Fi que abordan velocidades de datos medias y altas para servicios de voz, datos y video, ZigBee está diseñado para satisfacer necesidades típicas de sensores y dispositivos de control que no necesiten un gran ancho de banda, pero sí baja latencia y un consumo de energía mínimo. Los estándares como Bluetooth y Wi-Fi, debido fundamentalmente a la complejidad de los protocolos, no son adecuados para aplicaciones de bajo consumo de energía. Para paliar estas deficiencias se han desarrollado actualizaciones del estándar (Bluetooth Low Energy (v4.0 – v5.0)) y Wi-Fi de baja energía (IEEE 802.11ah) o LiFi.

Zigbee actualmente constituye una solución económicamente atractiva en sistemas de control y automatización: seguridad inalámbrica, termostatos a distancia, iluminación remota, lectura automatizada de medidores, atención médica personal, monitorización avanzada, control remoto universal, iluminación, etc. La arquitectura flexible de Zigbee permite diseñar topologías en estrella, malladas e híbridas como puede observarse en la Figura 2.11.

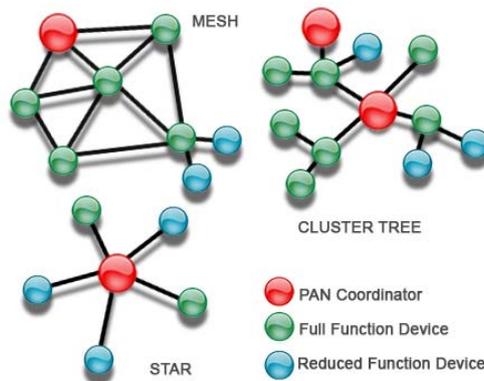


Figura 2.11 Arquitectura Zigbee (<http://www.zigbee.org>)

2.3.1.6.2 Bluetooth Low Energy

Bluetooth low energy o smart bluetooth es una tecnología de comunicación inalámbrica diseñada para dispositivos de bajo coste, bajo consumo energético y comunicación de corto rango. Bluetooth LE permite la comunicación entre dispositivos alimentados con baterías de pequeñas dimensiones tales como relojes, teclados inalámbricos y sensores que se conectan a dispositivos como ordenadores personales o teléfonos móviles. Bluetooth LE funciona en la banda de 2.4 GHz con una tasa de datos de 1 Mbps. La distancia de enlace entre dispositivos es de 5-10 m. Bluetooth LE está pensado para operar junto a Bluetooth y permite hacer a los dispositivos más pequeños y más eficientes energéticamente, aprovechando la existencia de la comunicación Bluetooth con un consumo de energía más bajo.

2.3.1.7 Internet of Things (IoT)

Gracias a la reducción del coste por conexión y el consiguiente rápido crecimiento en el número de conexiones máquina a máquina (MTC), se espera que el número de objetos conectados alcance los 50 mil millones en 2020 (2.7% de las cosas en el mundo). Si a esto añadimos que se espera que los precios de la conectividad se reduzcan a una tasa media de un 25% durante los próximos años, es probable un enorme crecimiento de objetos conectados según los costes de conectividad bajen. Por último, los desarrollos específicos en redes móviles para dispositivos (NB-IoT) que permiten reducir costes y complejidad en los dispositivos contribuirán decisivamente a este crecimiento.

Internet de las cosas (IoT) es una red constituida por todo tipo de dispositivos, sensores y terminales que pueden formar parte de una casa inteligente, un edificio, una empresa y una ciudad que se conectan entre sí principalmente a través de Internet.

IoT constituye el sistema nervioso de Ciudad Inteligente, apoyando la interconexión y el flujo de información entre el entorno, las máquinas, las infraestructuras y las personas.

A diferencia de la comunicación MTC, que conecta sensores y otros dispositivos a sistemas de información a través de redes cableadas o inalámbricas, IoT también se refiere a la conexión de tales sistemas y sensores a Internet en general, así como también al uso de tecnologías generales de Internet. El ecosistema de IoT no será muy diferente al de Internet actual, permitiendo que las cosas y los objetos del mundo real se conecten, se comuniquen e interactúen entre sí de la misma manera que lo hacen los seres humanos a través de la web.

IoT no es un nuevo tipo de red de Internet, es una extensión de Internet existente, ya no será Internet solo para las personas, los medios y el contenido, sino que también incluirá todos los activos del mundo real como criaturas inteligentes que intercambian información, interactúan con las personas, respaldan los procesos comerciales de las empresas y crean conocimiento.

IoT no solamente trata de la tecnología, la supervisión remota y el control sino también de dónde y cómo ésta se utiliza tanto en entornos abiertos como cerrados, por ejemplo, la automatización industrial. En este tipo de entornos surge también el concepto de "Intranet of Things" para determinar soluciones que aplican a entornos cerrados y propietarios.

Otro aspecto relevante es la necesidad de gestionar un muy elevado número de dispositivos y las limitaciones que presenta el protocolo IPV4 en cuanto a direccionamiento. El despliegue de IoT favorecerá una rápida implantación del protocolo IPv6.

En resumen, una definición de IoT puede ser la propuesta por Adrian MCEwen y Hakim Cassimally en su libro "Diseño de Internet de las cosas" [50]

$$IoT = \text{Objeto físico} + \text{Controlador, sensor y actuador} + \text{Internet}$$

IoT junto con otros desarrollos emergentes de Internet, como Internet de la Energía, los Medios, las Personas, los Servicios y las Empresas / Empresas, son la columna vertebral de la economía digital. Las previsiones sobre el desarrollo de IoT muestran que habrá 16 mil millones de dispositivos conectados para el año 2020, que promediarán hasta seis dispositivos por persona en la tierra y muchos más por persona en las sociedades digitales. Dispositivos como teléfonos inteligentes y comunicaciones de máquina a máquina (M2M) (o de una cosa a otra) serán los principales impulsores de su desarrollo.

La primera consecuencia directa del IoT es la generación de enormes cantidades de datos, donde cada objeto conectado puede ser generador y consumidor de información al relacionarse con otro objeto, con un servidor o con un humano.

La segunda consecuencia será el procesamiento de este enorme volumen de información y su transformación en conocimiento.

La tercera consecuencia será la correlación de la información entre diferentes sistemas y sectores de la industria para generar nuevas aplicaciones. Por ejemplo, la detección de

un incendio en un túnel debería generar mensajes a otros sistemas, la red de semáforos para impedir la circulación de vehículos particulares, favorecer el acceso a los servicios de bomberos y emergencias sanitarias, así como la desconexión de sistemas eléctricos y aviso a los centros hospitalarios más cercanos.

La última y no menos importante consecuencia será la necesidad de normalización de la información para que pueda ser gestionada de manera eficiente por diferentes sistemas.

El IoT permitirá que las personas y las cosas se conecten "en cualquier momento, en cualquier lugar, con cualquier cosa y cualquier persona", idealmente utilizando cualquier ruta / red y cualquier servicio. Esta es la visión de la UIT sobre el IoT.

2.3.1.7.1 Arquitectura de referencia IoT

Desde la perspectiva de la normalización técnica [51], IoT puede concebirse como una infraestructura global de la sociedad de la información, que permite ofrecer servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación (TIC) presentes y futuras.

Aprovechando las capacidades de identificación, adquisición de datos, procesamiento y comunicación, IoT utiliza plenamente las "objetos" para ofrecer servicios a todos los tipos de aplicaciones, garantizando a su vez el cumplimiento de los requisitos de seguridad y privacidad.

Como se muestra en la Figura 2.12, IoT añade la dimensión "Comunicación con cualquier objeto" a las tecnologías de la información y la comunicación (TIC), que ya ofrecen la comunicación "en todo INSTANTE" y "en cualquier LUGAR".

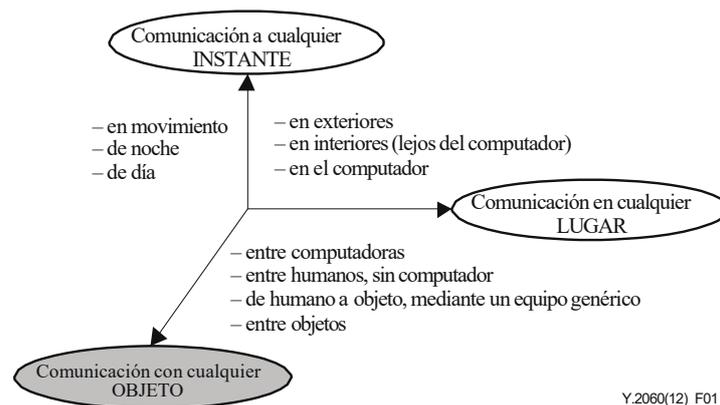


Figura 2.12 La nueva dimensión que introduce IOT (ITU-T Y.2060(12) F01)

En el contexto de IoT, los objetos son objetos del mundo físico (objetos físicos) o del mundo de la información (mundo virtual) que se pueden identificar e integrar en redes de comunicación. Los objetos tienen información conexas, que puede ser estática y dinámica.

Los objetos físicos existen en el mundo físico y es posible detectarlos, actuar sobre ellos y conectarlos. Ejemplos de objetos físicos son el entorno que nos rodea, los robots industriales, los bienes y los equipos eléctricos.

Los objetos virtuales existen en el mundo de la información y se pueden almacenar, procesar y acceder a las mismas. Ejemplos de objetos virtuales son el contenido multimedia y el software de aplicaciones.

2.3.1.7.2 Descripción técnica de IoT

La Figura 2.13 muestra una descripción técnica de IoT donde un objeto físico puede estar representado en el mundo de la información por una o varios objetos virtuales (correspondencia), pero el objeto virtual también puede existir sin tener asociado ningún objeto físico.

Un dispositivo es una pieza de equipo con capacidades obligatorias de comunicación y capacidades opcionales de detección, accionamiento, adquisición, almacenamiento y procesamiento de datos. Los dispositivos recaban diversos tipos de información y la suministran a las redes de la información y la comunicación para su posterior procesamiento. Algunos dispositivos también ejecutan operaciones en función de la información recibida de las redes de comunicaciones.

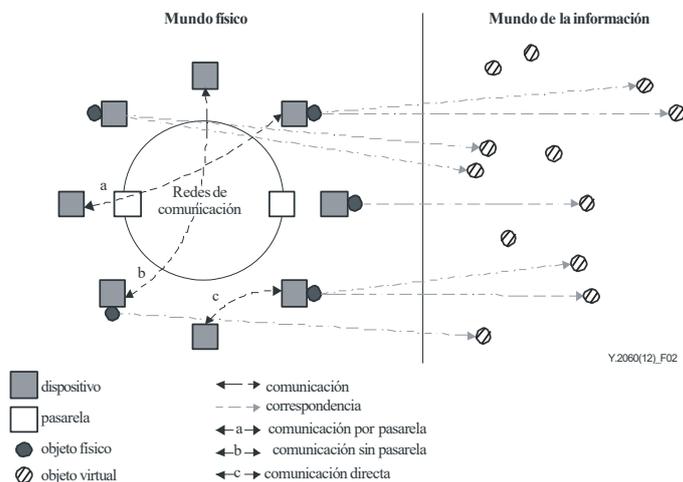


Figura 2.13 Descripción técnica de IoT (ITU-T Y.2060(12)F02)

Los dispositivos se comunican con otros dispositivos: a través de la red de comunicaciones por medio de una pasarela (caso a), por medio de una red sin pasarela (caso b) o directamente, esto es, sin utiliza la red de comunicación (caso c). También son posibles otras combinaciones de los casos a y c, y de los casos b y c; por ejemplo, los dispositivos pueden comunicarse con otros utilizando una red local (es decir, una red que ofrece conectividad local entre dispositivos y entre dispositivos y pasarelas, como una red ad-hoc) (caso c) y luego se comunican por medio de la red de comunicación a través de pasarela de red local (caso a).

Las redes de comunicaciones transfieren datos adquiridos por los dispositivos a aplicaciones y otros dispositivos, así como instrucciones de las aplicaciones a los dispositivos. Las redes de comunicación ofrecen capacidades de transferencia de datos fiables y eficientes. La infraestructura de red IoT puede crearse mediante redes existentes, como las redes convencionales basadas en TCP/IP, y/o redes evolutivas, tales como las redes de la próxima generación (NGN) [8].

El requisito mínimo que han de cumplir los dispositivos en IoT es que dispongan de capacidades de comunicación. Los dispositivos se clasifican en dispositivos de transporte de datos, de adquisición de datos, de detección y accionamiento y dispositivos genéricos, como se describe a continuación:

- Dispositivo de transporte de datos: Dispositivo anexo a un objeto físico para conectar indirectamente el objeto físico con las redes de comunicación.
- Dispositivo de adquisición de datos: Dispositivo de lectura/escritura con capacidad para interactuar con objetos físicos. La interacción puede suceder indirectamente a través de dispositivos de transporte de datos, o directamente a través de dispositivos de transporte de datos unidos a los objetos físicos. En el primer caso, el dispositivo de adquisición de datos lee la información sobre el dispositivo de transporte de datos y pueden también escribir información que suministran las redes de comunicación sobre el dispositivo de transporte de datos.
- Dispositivo de detección y accionamiento: Detecta o mide información de su entorno y la convierte en señales electrónicas digitales. También puede convertir señales electrónicas digitales procedentes de las redes de información en operaciones. Por lo general, los dispositivos de detección y accionamiento forman redes locales que se comunican entre sí utilizando tecnologías de comunicación alámbricas o inalámbricas y utilizan pasarelas para conectarse con las redes de comunicación.
- Dispositivo genérico: Dispositivo que cuenta con capacidades de procesamiento y comunicación y puede comunicarse con las redes de comunicación mediante tecnologías alámbricas e inalámbricas. Los dispositivos generales incluyen equipos y aplicaciones para diferentes dominios de aplicación IoT, tales como máquinas industriales, electrodomésticos y teléfonos inteligentes.

2.3.1.7.3 Características fundamentales y requisitos de alto nivel de IoT

Las características fundamentales de IoT son las siguientes:

- Interconectividad: En el contexto de IoT, todo puede estar interconectado con la infraestructura mundial de la información y la comunicación.
- Servicios relacionados con objetos: IoT es capaz de suministrar servicios relacionados con los objetos dentro de las restricciones de objetos, como protección de la privacidad y coherencia semántica entre los objetos físicos y sus correspondientes objetos virtuales. Para ofrecer servicios relacionados con objetos dentro de las restricciones de objetos, las tecnologías en el mundo físico y en el de la información evolucionarán.
- Heterogeneidad: Los dispositivos en IoT son heterogéneos dado que se basan en diferentes plataformas hardware y redes. Pueden interactuar con otros dispositivos o plataformas de servicios a través de redes diferentes.
- Cambios dinámicos: El estado de los dispositivos varía dinámicamente, por ejemplo, del modo reposo al activo, conectado y/o desconectado, así como el contexto del dispositivo, como la ubicación y velocidad. Además, el número de dispositivos también puede cambiar dinámicamente.
- Escala enorme: El número de dispositivos que ha de gestionarse y que se comunican entre sí puede ser incluso un orden de magnitud mayor que el número de dispositivos conectados actualmente a Internet. El porcentaje de comunicación que requerirán estos dispositivos será muchísimo mayor que el de la comunicación entre humanos. Será incluso más esencial la gestión de los datos generados y su interpretación para fines de aplicación, aspectos éstos que guardan relación con la semántica de datos y la manipulación eficiente de datos.

Los requisitos de alto nivel para IoT son los siguientes:

- Conectividad basada en la identificación: La IoT necesita que se establezca conectividad entre un objeto y IoT con arreglo al identificador del objeto. Para ello puede ser necesario además procesar de manera unificada identificadores posiblemente heterogéneos.
- Compatibilidad: Es indispensable garantizar la compatibilidad entre sistemas heterogéneos y distribuidos para el suministro y consumo de diversos tipos de información y servicios.
- Redes automáticas: Es necesario que las funciones de control de red de IoT soporte las redes automáticas (en particular técnicas y/o mecanismos de autogestión, autoconfiguración, autorestablecimiento, autooptimización y autoprotección), a fin de adaptarse a los diferentes dominios de aplicación, diferentes contextos de comunicación y números y diferentes tipos de dispositivos.
- Configuración automática de servicios: Es preciso poder configurar los servicios a partir de los datos de los objetos adquiridos, comunicados y procesados automáticamente con arreglo a las reglas configuradas por los operadores o

personalizadas por los clientes. Los servicios automáticos pueden depender de las técnicas de fusión y minería de datos automáticas.

- Capacidades basadas en la ubicación: IoT debe dar soporte a capacidades basadas en la ubicación. Las comunicaciones y servicios relacionados con objetos dependerán de la información sobre la ubicación de los objetos y/o los usuarios. Es necesario detectar y rastrear automáticamente la información sobre la ubicación. Las comunicaciones y servicios basados en la ubicación pueden estar limitados por leyes y reglamentos y deben cumplir los requisitos de seguridad.
- Seguridad: En IoT, todo 'objeto' está conectada lo que conlleva considerables amenazas de seguridad, en ámbitos tales como la confidencialidad, autenticidad e integridad de datos y servicios. Un ejemplo esencial de los requisitos de seguridad es la necesidad de integrar diferentes técnicas y políticas de seguridad para la diversidad de dispositivos y redes de usuario en IoT.
- Protección de la privacidad: Muchos objetos tienen sus propietarios y usuarios. Los datos detectados de los objetos pueden contener información privada acerca de sus propietarios o usuarios. IoT tiene que dar soporte a la protección de la privacidad durante la transmisión, combinación, almacenamiento, minería y procesamiento de datos. La protección de la privacidad no debe ser un obstáculo a la autenticación de las fuentes de datos.
- Servicios relacionados con el cuerpo humano de calidad y seguridad elevadas: IoT debe dar soporte a estos servicios. Cada país aplica leyes y reglamentos diferentes a estos servicios.
- Autoconfiguración (plug and play): IoT debe soportar la autoconfiguración que permite generar, componer o adquirir sobre la marcha configuraciones semánticas para la integración paulatina y la cooperación de los objetos interconectadas con aplicaciones, y para atender las necesidades de las aplicaciones.
- Capacidad de administración: IoT debe dar soporte a la capacidad de administración para garantizar el funcionamiento normal de la red. Las aplicaciones IoT suelen trabajar automáticamente sin la intervención humana, pero el proceso global de funcionamiento deben poderlo gestionar las partes pertinentes.

2.3.1.7.4 Modelo de referencia de IoT

La Figura 13 muestra el modelo de la arquitectura de referencia de IoT. Consta de cuatro capas y de capacidades de gestión y de seguridad relacionadas con estas cuatro capas.

Las cuatro capas son las siguientes:

- Capa de aplicación. Contiene las aplicaciones IoT.
- Capa de soporte de servicios y aplicaciones. Se compone de dos grupos de capacidades:
 - Capacidades de soporte genéricas: Son capacidades comunes que puede utilizarlas diferentes aplicaciones IoT, tales como procesamiento o almacenamiento de datos. Estas capacidades también pueden utilizarlas otras

capacidades específicas para, por ejemplo, crear otras capacidades específicas.

- Capacidades de soporte específicas: Son capacidades para atender las necesidades particulares de diversas aplicaciones. En realidad, pueden consistir en diversos grupos de capacidades precisas que ofrecen distintas funciones de apoyo a las diferentes aplicaciones IoT.

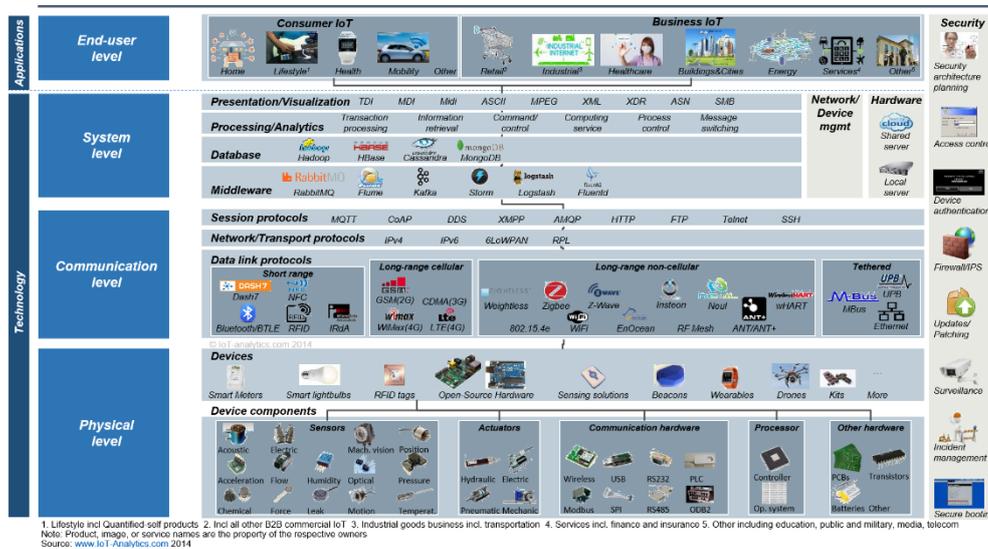


Figura 2.14 Arquitectura IoT (<http://www.IoT-Analytics.com> – 2014)

- **Capa de red.** Consiste en los dos tipos siguientes de capacidades:
 - Capacidades de red: ofrecen funciones de control de la conectividad en red, tales como funciones de control de acceso y de recursos de transporte, gestión de la movilidad y autenticación, autorización y contabilidad (AAA).
 - Capacidades de transporte: centradas en suministrar conectividad para el transporte de información y datos específicos de servicios y aplicaciones IoT, así como el transporte de información de control y gestión relacionada con IoT.
- **Capa de dispositivo.** Existen dos tipos de capacidades:
 - Capacidades de dispositivo:
 - Interacción directa con la red de comunicaciones: Los dispositivos pueden recabar y cargar información directamente (es decir, sin recurrir a capacidades de pasarela) en la red de comunicación y pueden recibir directamente información (por ejemplo, instrucciones) de la red de comunicación.
 - Interacción indirecta con la red de comunicación: Los dispositivos pueden

recabar y cargar información indirectamente en la red de comunicación, es decir, mediante capacidades de pasarela. Además, los dispositivos pueden recibir información indirectamente (por ejemplo, instrucciones) de la red de comunicación.

- Redes ad-hoc: Los dispositivos pueden construir redes de manera ad-hoc en algunas circunstancias cuando sea necesario para aumentar la capacidad evolutiva y la velocidad de despliegue.
- Modo reposo y activo: Las capacidades de dispositivo deben disponer de mecanismos para pasar a los modos "reposo" y "activo" a fin de ahorrar energía.
- Capacidades de pasarela:
 - Soporte de interfaces múltiples: En la capa de dispositivo, las capacidades de pasarela soportan dispositivos conectados mediante diferentes tipos de tecnologías alámbricas e inalámbricas, tales como el bus de red de control de zona (CAN), ZigBee, Bluetooth o Wi-Fi. En la capa de red, las capacidades de pasarela pueden comunicarse a través de diversas tecnologías, tales como la red telefónica pública conmutada (PSTN), las redes de segunda o tercera generación (2G o 3G), las redes LTE (evolución a largo plazo), Ethernet o las líneas digitales de abonado (DSL).
 - Conversión de protocolo: Hay dos tipos de situaciones en las que se necesitan capacidades de pasarela. Una es cuando las comunicaciones en la capa de dispositivo utilizan protocolos diferentes, por ejemplo, protocolos de tecnología ZigBee y Bluetooth, y la otra es cuando en la comunicación intervienen la capa de dispositivo y la de red y se utilizan protocolos diferentes en cada una, por ejemplo, el protocolo de tecnología ZigBee en la capa de dispositivo y el protocolo de tecnología 3G en la capa de red.

Adicionalmente a las capas de aplicación, soporte, red y dispositivos, existen dos planos que ofrecen capacidades de gestión y seguridad.

Capacidades de gestión. Análogamente a las redes de comunicaciones tradicionales, las capacidades de gestión IoT abarcan las clases tradicionales de fallos, configuración, contabilidad, rendimiento y seguridad (FCAPS), es decir, la gestión de fallos, de la configuración, de la contabilidad, del rendimiento y de la seguridad.

Las capacidades de gestión IoT pueden clasificarse en capacidades genéricas y específicas.

Las capacidades de gestión genéricas en IoT son esencialmente las siguientes:

- Gestión de dispositivos, como activación y desactivación de dispositivos remotos, diagnóstico, actualización del firmware y/o del software, gestión del estado de trabajo del dispositivo.
- Gestión de la topología de red local.
- Gestión del tráfico y la congestión, como la detección de las condiciones de

saturación de red y la aplicación de reserva de recursos para los flujos de datos esenciales para la vida o urgentes.

Las capacidades de gestión específicas están estrechamente relacionadas con los requisitos específicos de la aplicación, por ejemplo, requisitos de control de la línea de transmisión por la red de suministro eléctrico inteligente.

Las capacidades de seguridad permiten garantizar los objetivos de confidencialidad, autenticación e integridad. Al igual que en las capacidades de seguridad, se han definido capacidades genéricas y específicas. Las capacidades de seguridad genéricas son independientes de la aplicación y son, entre otras:

- En la capa de aplicación: autorización, autenticación, confidencialidad de datos de aplicación y protección de la integridad, protección de la privacidad, auditorías de seguridad y antivirus;
- En la capa de red: autorización, autenticación, confidencialidad de datos de señalización y de datos de uso, y protección de la integridad de señalización;
- En la capa de dispositivo: autenticación, autorización, validación de la integridad del dispositivo, control de acceso, confidencialidad de datos y protección de la integridad.

Las capacidades de seguridad específicas están estrechamente relacionadas con los requisitos específicos de la aplicación, por ejemplo, los requisitos de seguridad para el pago con el móvil.

2.3.1.7.5 IoT y 3GPP

Como se describió en el apartado 3.3.2.1.2 Redes de acceso inalámbricas, el 3GPP ha incorporado en sus recomendaciones soporte a dispositivos IoT no solo en cuanto a capacidades de transmisión, sino también en ahorro de costes, mejora de cobertura y número de dispositivos por celda

3GPP ha previsto tres líneas de evolución para dispositivos IoT, como se muestra en la Figura 2.15

- LTE-MTC
- LP-WAN
- GSM-MTC

Gran parte de los dispositivos móviles IoT actuales utilizan tecnologías de comunicaciones de segunda generación (GSM- GPRS) debido a los escasos requerimientos en general de las aplicaciones y a los menores costes que las soluciones 4G.

El coste de un módem LTE sigue siendo relativamente alto comparado con el de un módem GSM. No obstante, importantes mejoras en la gestión de la energía, en la cobertura global, eficiencia espectral, latencia, rendimiento de datos y número de dispositivos

por celda son aspectos clave para la rápida implantación de LTE-MTC y LP-WAN (NB-IoT).

La necesidad de soluciones optimizadas para el mercado del IoT (MTC) se identificó en una etapa temprana de desarrollo de LTE. Por ejemplo, en las versiones 10 y 11 se introdujeron características destinadas a proteger la red móvil frente a la sobrecarga derivada de la multitud de dispositivos del IoT (SIMTC, NIMTC), actualmente en la versión 13 > 50.000 dispositivos NB-IoT y GSM-MTC por celda).

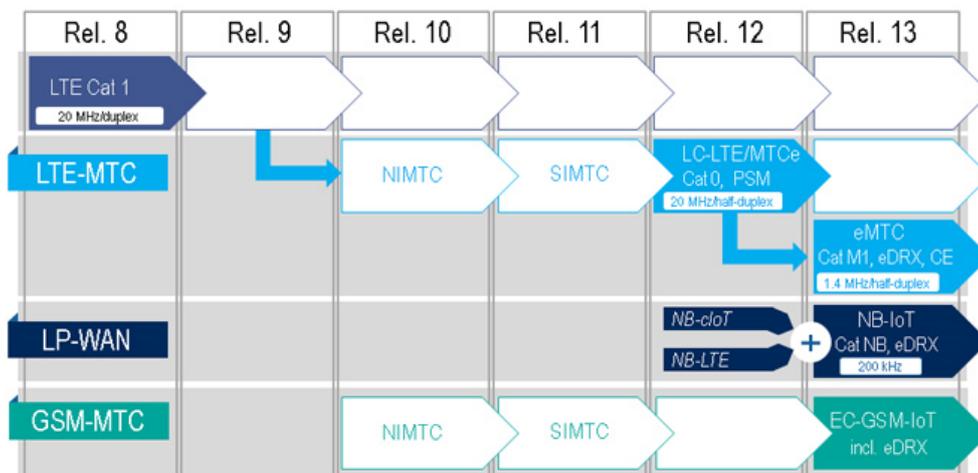


Figura 2.15 Evolución de IoT en 3GPP (<http://www.3gpp.org/specifications/releases>)

Con las mejoras de las comunicaciones de tipo máquina (MTCe) y la introducción de modo bajo coste de LTE (LC_LTE), la versión 12 introdujo soluciones optimizadas para dar respuesta a los principales requisitos de los dispositivos del IoT, entre los que se incluyen un nivel reducido de tráfico de datos, consumo de energía y costes:

- Un modo de ahorro de energía (PSM) apto para redes de sensores que solo necesitan enviar datos al dispositivo de forma ocasional.
- La nueva categoría 0 de dispositivos LTE que se supone que solo tiene un 50 % de la complejidad de un módem de categoría 1, lo que permite reducir la velocidad de transmisión compatible hasta 1 Mbps y eliminar el modo completamente dúplex y varias antenas.

La introducción de eMTC, o LTE-M en versión 13 incluye medidas adicionales de reducción de costes, mejora de cobertura y gestión de la energía para proporcionar a los dispositivos alimentados por batería una vida útil de varios años.

La versión 13 introduce una nueva tecnología denominada NB-IoT. NB-IoT ocupa menos de 200 kHz de espectro y se puede implantar dentro de banda mediante la reutilización de bloques de recursos LTE libres, o de forma independiente en el espectro entre portadoras LTE adyacentes o en el espectro GSM libre.

Entre las mejoras que aporta:

- NB-IoT es más económico que eMTC
- Mejora de cobertura de hasta 164 dB entre terminal y nodo B
- Duración de batería de hasta 10 años con una batería de 5 vatios-hora en cobertura y uso moderados.
- Soporte para un número elevado de dispositivos, hasta 50.000 -150.000 por celda.

En la Tabla 2.6 se muestran las principales categorías de LTE susceptibles de uso en IoT

Tabla 2.6 Categorías LTE para uso en MTC (IoT)

	LTE Cat. 1 (ver. 8)	LTE Cat. 0 (ver. 12)	LTE Cat. M1 (ver. 13)	NB-IoT (ver. 13)
Velocidad máxima enlace descendente	10 Mbps	1 Mbps	<1 Mbps	150 kbps
Velocidad máxima enlace ascendente	5 Mbps	1 Mbps	<1 Mbps	150 kbps
Ancho banda receptor estación móvil	20 MHz	20 MHz	1,4 MHz	200 kHz
Máx. potencia transmisión estación móvil	23 dBm	23 dBm	23 dBm o 20 dBm	23 dBm o 20 dBm
Modo dúplex	Full Dúplex	Semidúplex (opc.)	Semidúplex (opc.)	Semidúplex

En la versión 14 cabe esperar mejoras y optimizaciones adicionales, por ejemplo, la reducción de la latencia a 1 ms o la posibilidad de establecer comunicaciones entre vehículos (V2V), así como el aumento del número de dispositivos por celda.

2.3.2 Capa de Red (comunicaciones)

Este es uno de los principales elementos de la infraestructura de TIC. Esta capa permite la conectividad entre la capa de captación donde se encuentran dispositivos y subredes capilares y la capa de datos y aplicaciones.

La capa de comunicaciones se divide a su vez en subcapa de red de acceso y de agregación, que es la encargada de proveer la conectividad y el control de los usuarios, la subcapa de transporte que es la encargada de conectar las diferentes redes de acceso entre sí, y la capa de medios que es la encargada de proveer la conmutación y el interfaz con la capa de datos y aplicaciones.

La Figura 2.16 muestra un ejemplo de infraestructura en la capa de comunicaciones.

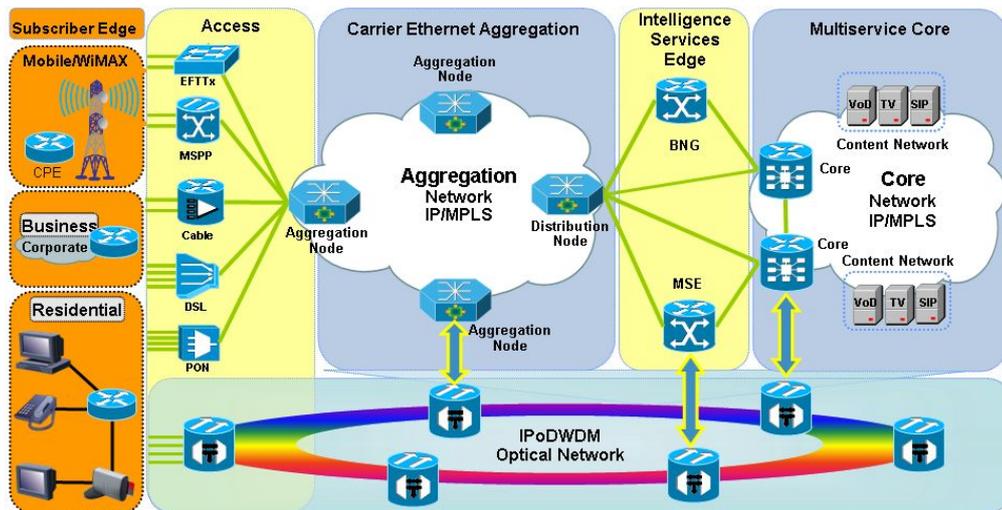


Figura 2.16 Capa de Comunicaciones (https://www.cisco.com/web/global_flagship/sp/zh/network_infrastructure/edge_network.html)

2.3.2.1 Redes de acceso

La red de acceso permite la conectividad entre dispositivos / red capilar a un nodo de la red de transporte. Es el componente más oneroso de la arquitectura dado el enorme número de usuarios y las dificultades de cobertura / costes subyacentes.

Existen múltiples tecnologías que pueden utilizarse en la red de acceso, su adopción dependerá de diferentes factores: ancho de banda, disponibilidad, costes y cobertura. En la Figura 2.17 se muestran las tecnologías más comunes, agrupadas por su naturaleza.

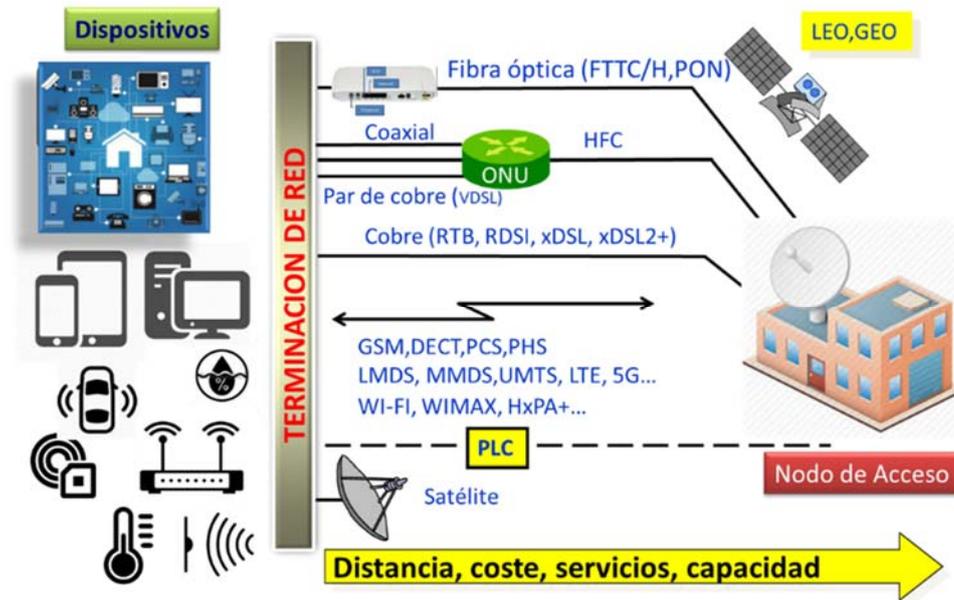


Figura 2.17 Tecnologías de Redes de Acceso

Para el acceso cableado, los portadores físicos pueden ser cobre (par trenzado o coaxial), fibra o una solución híbrida. Para el acceso inalámbrico las soluciones pueden ser satelitales, celulares o punto multipunto.

2.3.2.1.1 Redes de acceso cableadas

Las primeras soluciones para ofrecer servicios de banda ancha se basaron en la reutilización de la infraestructura telefónica y de difusión de televisión basadas en cobre. Las soluciones de acceso con fibra se reservaron fundamentalmente para el mercado de empresas.

- x digital subscriber line (xDSL)

Esta familia de tecnologías proporciona capacidad portadora de banda ancha sobre la red telefónica convencional. De esta manera, la información es transmitida en un rango de frecuencias mayor que el reservado para la voz, evitando interferencias entre ambos servicios. Esta solución requiere una nueva infraestructura en el punto de acceso (modem xDSL + filtros) y la instalación en la red de acceso de multiplexor (DSLAM) que consolide el tráfico de los usuarios.

Esta solución, ampliamente desplegada por los operadores incumbentes posee, sin embargo, importantes limitaciones derivadas de la propia naturaleza de la infraestructura de cobre: longitud del bucle e interferencias.

En la Figura 2.18 puede observarse una arquitectura típica de una red de acceso multi-servicio basada en esta tecnología.

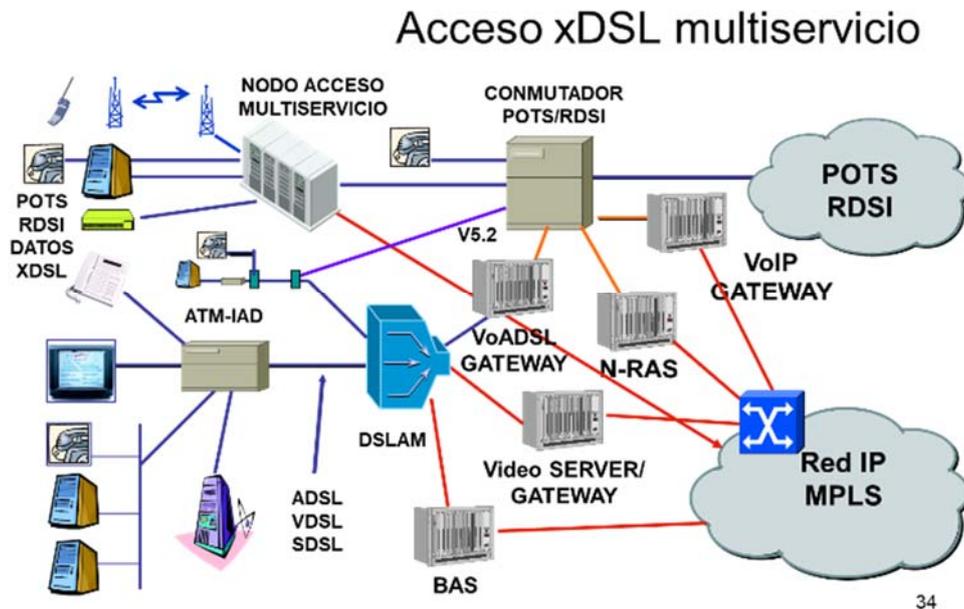


Figura 2.18 Arquitectura red de acceso xDSL

- Redes de acceso FTTx

Fibra hasta “x” es un término genérico para una arquitectura de acceso de banda ancha que utiliza fibra óptica para la provisión de parte o de la totalidad de red de acceso. Este término es la generalización de diferentes arquitecturas utilizadas en el despliegue de esta red en función de la cercanía del usuario a la misma. Comúnmente, las soluciones abarcan desde la fibra hasta el vecindario (FTTN) hasta la fibra en el puesto de trabajo (FTTD). El despliegue de una u otra solución depende de criterios técnico-económicos y de la naturaleza del servicio a proporcionar.

- Fibra hasta el nodo, vecindario o ultimo amplificador (FTTN/FTTLA): la fibra llega desde el nodo de acceso a un punto, normalmente un armario en exterior, y desde ahí la distribución al usuario final se realiza por cobre, par trenzado o coaxial que suele estar a una distancia de hasta 3 km. Esta solución es normalmente un primer paso hacia el escenario de fibra hasta el hogar.
- Fibra hasta la acera (FTTC/FTTK): Es muy similar a FTTN, pero la distancia al usuario final es mucho menor, alrededor de los 300 m. En este caso, al disminuir

la distancia de red de cobre se pueden ofrecer servicios de mayor ancho de banda basados en VDSL o disminuir el número de amplificadores en redes de cable.

- Fibra hasta el edificio (FTTB): La fibra en este caso llega al edificio y la distribución final se realiza por cobre. Esta solución representa importantes ahorros a los operadores debido a que este tramo de cobre es el más costoso de instalar y puede aprovecharse la infraestructura de cobre existente.
- Fibra hasta el hogar (FTTH): La fibra llega en este caso hasta el usuario. La terminación de red (ONT) se conecta a la red de área local o doméstica del usuario final, normalmente cable UTP Cat 5 o superior.
- Fibra hasta el puesto de trabajo (FTTD). En este caso, la red local o doméstica del usuario final también es óptica. La aparición de la fibra de plástico con menores costes de instalación y capacidad de transmisión mayor de 1Gb/s. permitirá la instalación masiva de este tipo de tecnología.

Una de las tecnologías de mayor implantación, y que permite importantes ahorros de costes son las redes ópticas pasivas (PON). Esta arquitectura punto multipunto permite, utilizando divisores ópticos pasivos, compartir secciones de una misma fibra a diferentes usuarios, típicamente de 32 a 128. Una red PON consiste en un terminador de red óptico (OLT) conectado a la red de acceso, la red óptica pasiva punto multipunto y un terminador de red óptico (ONT) en los usuarios. En la Figura 2.19 puede observarse un esquema de la arquitectura de red FTTH.

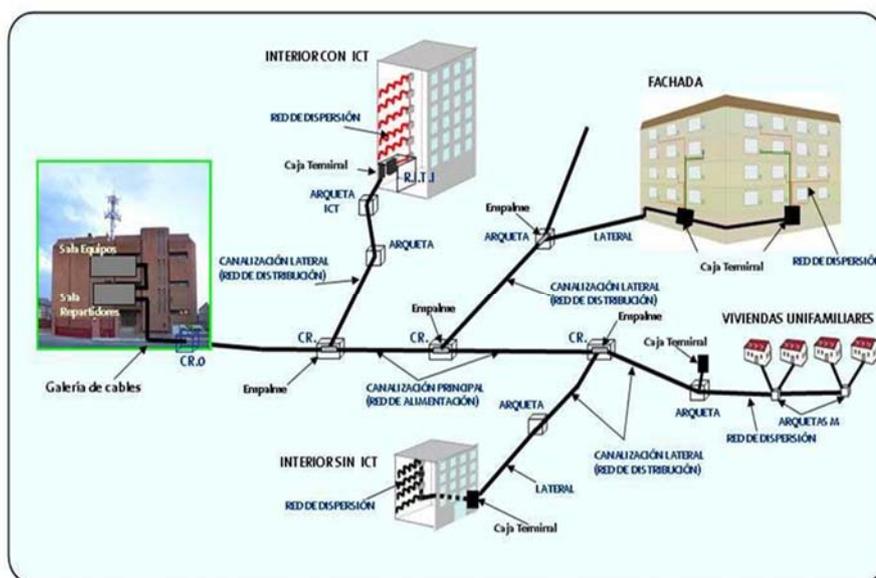


Figura 2.19 Arquitectura FTTH (www.telefonica.com)

2.3.2.1.2 *Redes de acceso inalámbricas*

Las tecnologías inalámbricas de banda ancha son la solución más adecuada de conectividad en áreas donde la disponibilidad de red fija es limitada, la dispersión de usuarios es alta, la construcción de una nueva red es onerosa o en situaciones de emergencia. Las redes de acceso inalámbricas de área amplia poseen importantes ventajas en términos de rapidez de despliegue y cobertura. También las redes de acceso inalámbricas proveen conectividad para dispositivos o usuarios en movilidad.

Estas ventajas cuantitativas y cualitativas han fomentado un importante crecimiento en la demanda usuarios, servicios y aplicaciones de características heterogéneas: ancho de banda, sensibilidad al retardo, etc. aspecto que ha impulsado el desarrollo de tecnologías de acceso radio avanzadas que optimizan el uso del espectro, y la reutilización de frecuencias.

Existen soluciones tecnológicas de nicho como LoraWan y SiGFox optimizadas para el despliegue de soluciones IoT, pero las redes celulares basadas en los estándares del Third Generation Partnership Project (3GPP) ofrecen mejoras sustanciales en flexibilidad, cobertura y evolución futura.

El 3GPP es el organismo de estandarización de los sistemas de comunicaciones móviles, especialmente en los de última generación denominados Long Term Evolution (LTE). Los estándares 3GPP se organizan en “versiones” (releases). La primera especificación de LTE se introdujo en la Versión 8 de diciembre de 2007 y desde entonces, se han ido añadiendo nuevas funcionalidades y mejoras para proporcionar nuevas clases de servicio, velocidades, movilidad, etc.

Merece la pena reseñar, pues es de interés para esta Tesis, la especificación del 3GPP TS 36.306 [53] sobre las capacidades del equipamiento de usuario (UE) en redes de acceso radio donde se definen las capacidades de transmisión de los equipos de usuario y por ende las clases de servicio y ancho de banda máximo de las fuentes de tráfico subyacentes.

2.3.2.2 *Redes de transporte*

Las redes de transporte proveen la conectividad entre diferentes redes de acceso y están caracterizadas por ofrecer elevadas capacidades de transmisión a grandes distancias. El estudio de la arquitectura de las Redes de Transporte fue abordado formalmente por la UIT-T en las recomendaciones G.803, G.805 y G.872 donde se define un modelo de “capas” con una relación cliente-proveedor entre ellas y unas funciones de conectividad (adaptación y terminación de trayectos y subredes) cuya descripción excede al alcance del presente documento.

De manera resumida, la capa de medios físicos se corresponde con la infraestructura de fibra óptica o radio subyacente. Sobre esta capa se pueden implementar y de manera no excluyente tres tecnologías dominantes: Jerarquía Digital Síncrona (SDH-G.707), Multiplexación densa por longitud de onda (DWDM-G.671) y la Red de Transporte Óptica

(OTN-G.872) que permiten no solamente multiplicar la capacidad de transporte si no también multiplexar señales de naturaleza heterogénea sobre un sistema de transporte unificado.

2.3.2.3 Capa de medios

La capa de medios (media layer) es el núcleo común de encaminamiento de información en las redes de Siguiete Generación (NGN).

Está especializada en tratar paquetes eficientemente a través de una asignación dinámica de las capacidades de la red de transporte de alta capacidad subyacente (SDH, WDM y OTN) e incluye además elementos para convertir otros formatos de comunicación (circuitos) en formato de paquete. La capa de medios utiliza una capa de control que se encarga de diferenciar tráficos de diferente naturaleza, adaptar velocidades y mantener la calidad de servicio.

MPLS es la tecnología que permite la optimización del tráfico IP mejorando la capacidad de conmutación, la gestión de diferentes clases de servicio y la creación de redes privadas virtuales (VPN).

En la capa de medios también se ubican los repositorios de información necesarios para la gestión y la supervisión de servicios. Ejemplos de esto pueden ser las bases de datos de perfiles de usuario, autenticación, autorización, ubicación de dispositivos etc.

2.3.3 Capa de Datos y Soporte

Esta capa contiene los repositorios centralizados de datos que son necesarios para el consumo de capas superiores y está compuesta fundamentalmente por los centros de datos y almacenamiento más los interfaces con las capas de aplicaciones y de red.

Esta capa de datos y soporte es un elemento crucial de la arquitectura debido a la necesidad de almacenar importantes volúmenes de datos sino también por los requerimientos de seguridad, escalabilidad y prestaciones.

Los Centros de Datos Gestionados proporcionan la infraestructura física y lógica necesaria para satisfacer estas necesidades. En la Figura 2.20 se muestra una vista conceptual de un centro de datos.

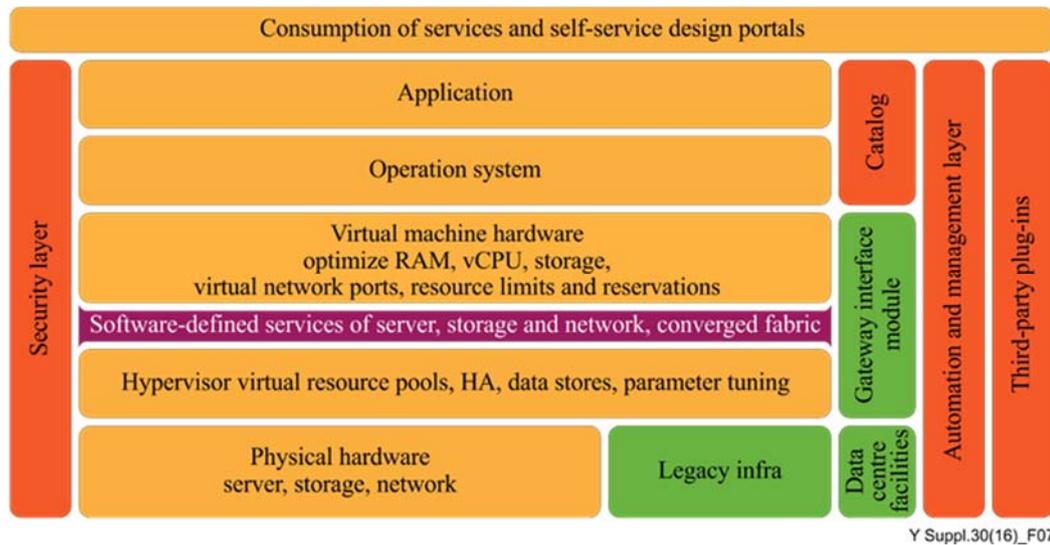


Figura 2.20 Arquitectura de un Centro de Datos Gestionado (ITU-T Y.Suppl.39(16)_F07)

2.3.4 Capa de Servicios, aplicaciones y contenidos

Esta capa incluye las aplicaciones que administran y entregan los servicios de Ciudad Inteligente.

En esta capa pueden a su vez distinguirse otras subcapas relacionadas:

- *Capa de usuarios*: organiza a los usuarios finales del servicio en grupos tanto desde el lado de la demanda como de la oferta.
- *Capa de presentación*: contiene las interfaces de usuario (web, aplicaciones, etc.) que se encuentran entre los usuarios finales y los servicios que ofrece la ciudad.
- *Capas de aplicaciones software*: Contiene todas las aplicaciones de software correspondientes que realizan los servicios de ciudad. Normalmente se definen un conjunto de “sectores verticales”, ligados a las áreas clásicas de gestión ciudadana.
 - *Energía y agua*
 - *Transporte*
 - *Salud*
 - *Seguridad y emergencias*
 - *Educación y turismo*
 - *Entorno y gestión de residuos*
 - *Edificios inteligentes*

- *Capa de negocio*: Encapsula las reglas o la forma en que los datos adquiridos desde la capa de servicio deben ser manipulados o transformados para su procesamiento por la capa de aplicación y viceversa.

2.4 Plataformas de Ciudad

En la actualidad, los municipios cuentan con diferentes aplicaciones para resolver necesidades operativas, normalmente ligadas a uno de los ejes “verticales” o áreas de actuación definidas como la movilidad, la seguridad, el alumbrado público o la gestión de residuos, y que en gran parte de los casos se corresponden con estructuras organizativas municipales. También existen sistemas externos de naturaleza heterogénea, como servicios de transporte, sistemas ligados a edificios, turismo, hoteles, redes sociales etc. gestionados por terceros.

Todos estos sistemas cuando existen son en general independientes entre ellos y muchas veces propietarios. Pueden proporcionar datos útiles acerca del estado de la ciudad y pueden ser clasificados a través de diferentes tipos:

- Plataformas IoT.
- Sistemas de Supervisión, control y adquisición de datos (SCADA).
- Plataformas No-IoT.
- Procesadores de bigdata.
- etc.

Este enfoque vertical lleva asociado una gestión de los servicios e información en silos, impidiendo en muchos casos un uso óptimo de la misma. Esta situación se ha agravado en la medida que los ciudadanos demandan cada vez servicios más complejos que en muchos casos requieren la compartición de información en tiempo real entre diferentes departamentos internos y actores externos.

El concepto de Ciudad Inteligente (SSC) persigue superar el concepto de ciudad sensorizada permitiendo la interacción entre diferentes servicios y aplicaciones tanto internas como externas, con el objeto de proporcionar servicios que mejoren la calidad de vida de los ciudadanos. La mayor dificultad para esta transformación es identificar elementos que ya existen (sensores, servicios, redes sociales etc.), definir como compartir recursos y datos, de forma que puedan interactuar para crear o mejorar los servicios de la ciudad y por último disponer de una infraestructura tecnológica unificada que permita interoperar con diferentes tipos de plataformas legacy existentes.

Por ello, desde la perspectiva de la ciudad, una plataforma de Ciudad Inteligente (P-SSC) que permita controlar múltiples aplicaciones facilitará un control más eficiente y efectivo de la infraestructura y los servicios públicos, permitirá eficiencias económicas y desempeñará un papel fundamental en el logro de objetivos ambientales y otros objetivos

sociales, tales como reducir la delincuencia y mejorar la movilidad. También puede acelerar y simplificar la implementación de otras aplicaciones.

La Plataforma SSC posee capacidades para el acceso a diferentes fuentes de información, compartir recursos, capacidad de análisis y coordinación de servicios (orquestración) basándose en análisis predictivos. El concepto de horizontalidad que provee esta plataforma, donde la información proveniente de todas las fuentes podrá ser utilizada de manera arbitraria, es clave en el concepto de Ciudad Inteligente.

Definir una arquitectura normalizada de Gestión de una Ciudad Inteligente, es decir, de una plataforma o sistema operativo de ciudad es objeto de creciente atención especialmente en aspectos como la interoperabilidad entre Plataformas SSC, entre éstas y aplicaciones legacy, sistemas de información propietarios y con dispositivos MTC, IoT.

El concepto de interoperabilidad definido en esta Norma PNE UNE 178104 “*Ciudades Inteligentes. Requisitos de interoperabilidad para una Plataforma de Ciudad Inteligente*” [54] está basado en un modelo de estructura de capas que permite la integración de diferentes sistemas, a través de un mecanismo de intercambio abierto y a ser posible estandarizado. También en la norma se presentan la definición, los requisitos y las interfaces entre las capas de la Plataforma SSC, que potencia y facilita el despliegue de servicios SCC y la reutilización de las aplicaciones existentes.

En la Norma UNE 178104 también han sido definidos los elementos básicos de cada capa, al objeto de asegurar que la Plataforma SSC proporcione los servicios esenciales requeridos por la ciudad, y al mismo tiempo permita la generación de nuevos servicios y aplicaciones que mejoren la calidad de vida de los ciudadanos. El modelo ha sido desarrollado para asegurar la máxima eficiencia, escalabilidad e integración; y también favorecer el desarrollo económico local y social a través de la competitividad e innovación.

La segunda edición de esta Norma UNE 178104 se ha elaborado teniendo en cuenta los trabajos promovidos por España en la Comisión de Estudio 20 sobre IoT y ciudades inteligentes de la Unión Internacional de Telecomunicaciones (borrador de Recomendación UIT-YSSCP).

A modo de resumen, una Plataforma de Ciudad Inteligente permite:

- El conocimiento en tiempo real del estado de la Ciudad recogiendo la información de la Ciudad, ciudadanos y empresas, cumpliendo los requisitos de privacidad que fueran pertinentes.
- La coordinación y puesta a disposición de la información disponible por parte de los gestores de los servicios de mantenimiento de la ciudad.
- La gestión dinámica de las actividades de acuerdo con datos actuales, recursos disponibles y niveles objetivos de calidad de los servicios.
- La toma de decisiones devolviendo la información refinada a los sistemas encargados de ejecutar las distintas acciones.

- El establecimiento de los canales de interacción con el gobierno de la Ciudad y con los Ciudadanos a través de subsistemas específicos que establezcan flujos bidireccionales de información.
- La oferta de datos abiertos y capacidades a desarrolladores para facilitar la creación de un ecosistema de aplicaciones sobre la plataforma, que cree un valor adicional para el ciudadano.

2.4.1 *Requisitos funcionales de una Plataforma de Ciudad Inteligente*

De manera muy resumida se detallan las funcionalidades mínimas que deberá poseer una Plataforma de Ciudad Inteligente.

2.4.1.1 *Repositorio completo y actualizado de información de la ciudad*

- Albergar un catálogo común, universal, mantenido, accesible y clasificado de datos únicos y normalizados de la ciudad incluyendo los activos de la ciudad (el mantenimiento de los mismos seguirá siendo responsabilidad de los sistemas de gestión que los generan).
- Permitir visiones analíticas transversales de la ciudad a partir de estos datos.
- Facilitar y universalizar la integración de datos y del legacy de soluciones existentes en la ciudad.
- Proporcionar interfaces abiertos para el desarrollo de aplicaciones inteligentes basados en los datos de la ciudad.
- Asegurar la seguridad y la integridad de los datos, así como la información de los usuarios.

2.4.1.2 *Gestión de las infraestructuras*

La plataforma soportará monitorización y operación centralizada, segura y multiusuario sobre los diferentes recursos, elementos o sistemas de una ciudad:

- Acceso a los datos de plataformas de sensores, bases de datos y a información de otras aplicaciones.
- Actuaciones sobre actuadores (sensores) a través de soluciones estandarizadas.
- Registro de las diferentes actividades que se desarrollan en el sistema.
- Gestión del mantenimiento de equipos e infraestructuras.
- Soporte protocolos estándar de monitorización como SNMP o JMX.
- Integración con otros sistemas y aplicaciones, como por ejemplo Sistemas para la gestión de la energía y funcionamiento de equipos (fuentes, iluminación, gestión de edificios, etc.). Control semafórico. Transporte público. Estaciones medioambientales. Gestión de agua (riego, alcantarillado). Recogida de basuras. Video vigilancia. Aparcamiento público y regulado. Puntos de recarga del vehículo eléctrico. Sistemas de información ciudadana (sistemas de quejas, notificación de incidencias en la vía pública, emergencias urbanas, redes sociales

y turismo). Redes sociales. ERP corporativo. GIS municipal. Sistemas de sensorización, etc.

2.4.1.3 *Comunicación entre sistemas*

La plataforma debe permitir la comunicación con los diferentes sistemas susceptibles de ser integrados en la ciudad.

- Proveer las interfaces necesarias para que eventos de un sistema puedan desencadenar acciones en otros sistemas.
- Usar APIS y protocolos normalizados para comunicación entre aplicaciones y otras plataformas o sistemas de gestión.
- Tener la capacidad de extenderse para soportar otros protocolos de comunicación, como comunicaciones entre sistemas que permitan la ejecución de funciones complejas, como por ejemplo la regulación semafórica para favorecer el acceso a los servicios de emergencia en caso de accidente junto con el aviso a los organismos involucrados.

2.4.1.4 *Seguridad*

Es un elemento básico del sistema ya que en la plataforma residirá información crítica para el funcionamiento de la ciudad, así como datos de ciudadanos protegidos por la ley.

2.4.1.4.1 *Aspectos generales*

La Plataforma SSC debería implementar las siguientes herramientas o sistemas que permitan:

- Realizar copias de seguridad de información crítica.
- Garantizar el no repudio en la red de seguridad.
- Soportar la anonimización de los datos.
- Garantizar la seguridad y la integridad de los datos.
- Soportar autenticación y autorización.
- Controlar el acceso a la plataforma y a todos los elementos a los que se acceda a través de esta: sensores, Scadas, centros de control, bases de datos y aplicaciones a las que se acceda a través de ella.
- Asegurar la confidencialidad en las comunicaciones con la Plataforma.
- Garantizar confidencialidad en el acceso a los datos, de modo que cada rol sólo pueda ver los datos a los que tiene acceso.
- Definir y gestionar políticas de seguridad.
- Proveer un módulo central y de fácil acceso (vía web) para la administración de los usuarios, roles y permisos.
- Integrar repositorios de usuarios ya existentes como registros usuarios de autoridades locales u otras Bases de Datos de usuarios.
- Tener capacidad de extenderse para adaptar los mecanismos de seguridad a las necesidades propias de cada ciudad.

2.4.1.4.2 Perfiles de usuario

La plataforma debe asegurar la privacidad y seguridad de los datos almacenados o gestionados por la solución, especialmente en un entorno compartido de recursos (PaaS: Plataforma como Servicio). Asimismo, se deben poder definir distintos perfiles de acceso a los diferentes tipos/grupos de datos, que eviten un uso inadecuado de los mismos.

La plataforma debe garantizar el envío y recepción segura de datos desde y hacia los dispositivos conectados a ella, así como su distribución segura a los aplicativos que los requirieran. Como mínimo debe implementar la autenticación de los elementos que originan los datos y de los aplicativos que requieran acceso a dichos datos.

La plataforma debe permitir definir diferentes roles y niveles de acceso sobre los datos, funcionalidades y servicios de la plataforma, autorizar o denegar el acceso a los distintos aplicativos y definir los privilegios requeridos para actuar sobre un determinado conjunto de datos.

Los usuarios de la Plataforma SSC pueden ser individuos o aplicaciones que consuman servicios o información. Deben considerarse diferentes tipos de acceso para los usuarios en función de diferentes roles como:

- Administrador del sistema.
- Operador.
- Directivos.
- Clientes sw de otras aplicaciones.
- etc.

La gestión de roles/permisos debería establecerse como mínimo respecto a tres niveles de seguridad:

- Acceso a los datos: limitar la información que puede visualizar cada usuario. Por ejemplo: un usuario de un determinado Servicio sólo tendrá acceso a la información correspondiente a los datos de su Servicio, datos generales como medias globales, desviaciones u otros que se obtengan del tratamiento conjunto de los datos correspondientes a todos los Servicios.
- Acceso a los elementos de la Plataforma SSC: limitar el acceso a los informes y cuadros de mando configurados en la plataforma. Por ejemplo, un usuario de un Servicio sólo podrá acceder a los informes definidos con los datos correspondientes a su ámbito.
- Funcionalidad: delimitar las acciones que puede realizar un determinado usuario en función de su perfil. Por ejemplo: un usuario de reporte podrá determinar a qué informes, a qué objetos tiene acceso un usuario consumidor-. O por ejemplo los usuarios consumidores sólo podrán realizar consultas en los informes a los que tengan permiso de acceso.

2.4.1.5 *Operación, Administración y Mantenimiento (OA&M)*

La Plataforma SSC debe incluir los indicadores necesarios para permitir su adecuado mantenimiento asegurando:

- Mantenimiento Preventivo: Almacenamiento y valoración de indicadores relevantes para la gestión del mantenimiento. Generación de planes de mantenimiento a partir de esos datos.
- Mantenimiento Correctivo (Gestión de avisos): Posibilidad de gestionar los avisos o alarmas y poder enviar mensajes, correos, SMS, llamadas en función de estos.

2.4.1.6 *Control de procesos*

La Plataforma SSC debería permitir el control de la ejecución de los siguientes procesos, ya sean internos o externos:

- Análisis de consumos, alarmas, tendencias, etc.
- Imputación de costes.
- Sostenibilidad (uso eficiente de las instalaciones, emisiones, etc.).
- Optimización de procesos y planificación.
- Control de calidad de servicios públicos por terceros.
- Sala de crisis.
- Generación de informes de explotación.

2.4.1.7 *SopORTE a la decisión*

La Plataforma SSC debería disponer de herramientas que permitan mejorar la resiliencia de la ciudad y la toma de decisiones como puedan ser:

- Simulación en base a la información actual e histórica.
- Valoración y ejecución de planes de actuación, en escenarios complejos.
- Análisis predictivo y modelado de ciudad.
- Minería de datos y el análisis estadístico.
- Integración con otros sistemas y herramientas de BI.

2.4.1.8 *Difusión de información pública en tiempo real*

La plataforma debe permitir transmitir la información de forma abierta, fidedigna y de calidad, de forma continua y sin interrupciones, en formato estándar para permitir el acceso a ella desde múltiples dispositivos. Esta información aplica a escenarios como:

- Servicios finales para al ciudadano.
- Aplicaciones de terceros (open data).
- Otros servicios públicos y administraciones.
- Rendición de cuentas (transparencia).

2.4.1.9 Disponibilidad

Este es uno de los elementos de mayor criticidad dado que la plataforma deberá garantizar la continuidad operativa de los servicios que proporciona, y que, en muchos casos, podrían requerir operación 24x7 y un nivel de disponibilidad superior al 99,9%.

Conseguir estos objetivos de disponibilidad implica la adopción de arquitecturas físicas redundantes/tolerantes a fallos tanto en computación como en almacenamiento, energía y comunicaciones.

2.4.2 Modelo de capas de la Plataforma SSC

2.4.2.1 Aspectos generales

En este apartado se describe el modelo que debe seguir la Plataforma SSC. Este modelo se ha construido como una arquitectura de múltiples capas y está diseñado para cumplir los requisitos de interoperabilidad. Para su desarrollo se han tenido en cuenta varias plataformas de ciudades inteligentes y sostenibles existentes en la actualidad.

Este modelo multicapa, que se muestra en la Figura 2.21, se describen los elementos que pueden componer la plataforma, dejando de lado los aspectos de infraestructura, de acuerdo con las siguientes capas:

- Adquisición/interconexión.
- Conocimiento.
- Interoperabilidad.
- Servicios.
- Soporte.

La infraestructura y las comunicaciones de los elementos contenidos en cada una de las capas, mostrados para información en la Figura 2.21, se han contemplado en apartados anteriores.

La comunicación entre diferentes capas y la comunicación con terceros u otras plataformas son puntos clave y se muestran como flechas moradas. En concreto, las comunicaciones clave son:

- Interfaz de Servicios. Interfaz que une las capas de servicios con y la capa de interoperabilidad: La interfaz que conecta estas dos capas debe estar abierta y debería estandarizarse. Sin embargo, siempre y cuando no exista una interfaz estandarizada, se puede utilizar opcionalmente otras interfaces. Por lo tanto, además de las API abiertas y normalizadas entre estas dos capas, podrían existir conectores específicos en la capa de servicio para proporcionar servicios específicos.
- Interfaz de Interoperabilidad. Interfaz que conecta las capas de Interoperabilidad y Conocimientos: La interfaz que conecta estas dos capas debe estar abierta y debe estandarizarse.

- *Interfaz de Recolección de Datos/Metadatos.* Interfaz que conecta las capas de Conocimiento y Adquisición: La interfaz que conecta estas dos capas debe estar abierta y debe ser estandarizada.

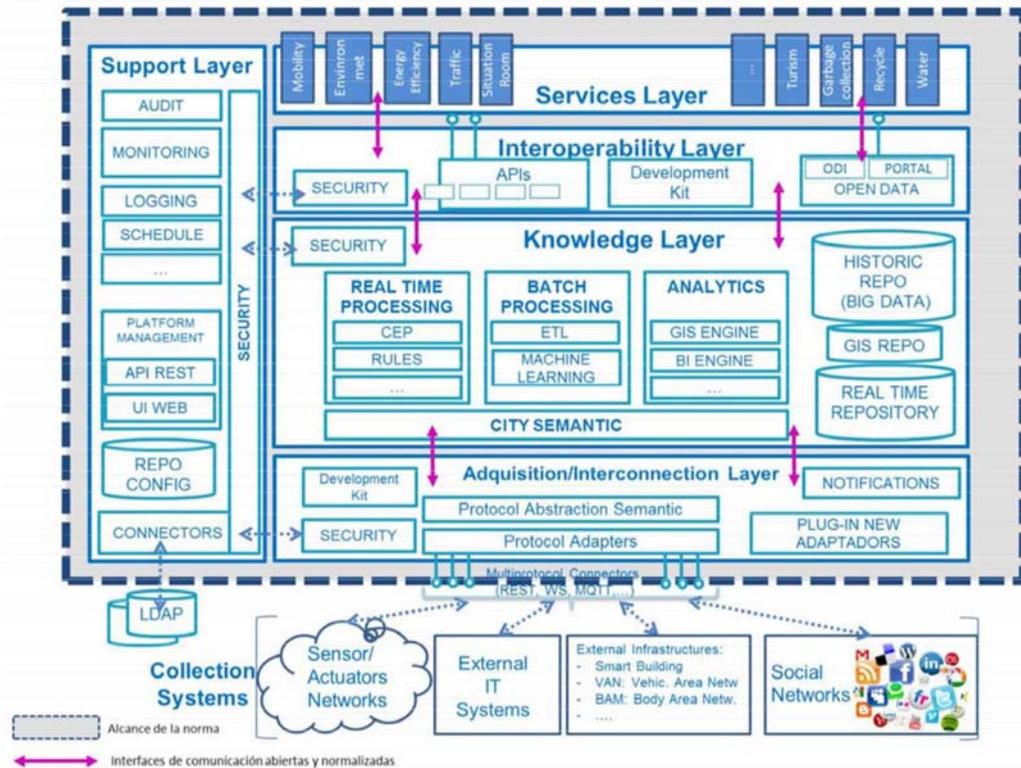


Figura 2.21 Modelo de capas de la Plataforma de Ciudad Inteligente

- La Capa de Adquisición/Interconexión: ofrece los mecanismos para la recepción de los datos desde los Sistemas de Captación, también es la encargada de permitir la interconexión con otros sistemas externos que sólo consuman datos y por último, abstrae la información de los Sistemas de Captación con un enfoque semántico.
- La Capa de Conocimiento: ofrece el soporte para el procesado de los datos, la incorporación de valor y la transformación de servicio. Recibe datos tanto de la Capa de Adquisición (sensores) como de la Capa de Interoperabilidad.
- Capa de Interoperabilidad: facilita la prestación de los servicios, ofreciendo interfaces abiertos y normalizados sobre la Capa de Conocimiento estableciendo políticas de seguridad y ofreciendo conectores para que sistemas externos puedan acceder a la Plataforma y viceversa. También permite construir servicios a

partir de los datos de la Plataforma.

- Capa de Servicios: se trata de servicios conectados a través de la capa de interoperabilidad y de los APIs provistos. Estas aplicaciones podrán estar corriendo sobre la Plataforma o ser servicios externos que publican o consumen información.
- Capa de Soporte: esta capa transversal da soporte al resto de funcionalidades ofreciendo servicios como auditoría, monitorización, seguridad
- Sistemas de recogida: Aunque no forman parte de la plataforma, están formados por todos los sensores/actuadores gestionados por una ciudad (semáforos, riego de parques, etc.), dispositivos ciudadanos (móviles, etc.), sistemas informáticos externos, redes sociales, etc.

2.4.2.2 *Capa de Adquisición/Interconexión*

La capa de Capa de Adquisición/Interconexión se encarga de:

- Integrar la información desde las fuentes de datos (Sistemas de Captación), que pueden ser:
 - Sensores, actuadores, gateways y dispositivos como semáforos, edificios, estaciones meteorológicas, etc. desde redes de Sensores gestionadas.
 - Diferentes dispositivos como móviles de las personas o aparatos del entorno del hogar, vehículos, etc. desde redes públicas.
 - Redes sociales.
 - Otros sistemas TI como SCADAS o soluciones de gestión para los dominios verticales, que pueden ser soluciones propietarias.
- Suministrar la Información a la Capa de Conocimiento con independencia de los dispositivos dando una vista semántica de los datos adquiridos, desacoplada de los protocolos de adquisición.
- Independencia del operador de red tanto de la disposición de información de red como del control de la misma.

Teniendo en cuenta, la Capa de Adquisición de una solución idónea de plataforma debería ajustarse a los principios del modelo ETSI de M2M:

- Interfaces abiertos y normalizados frente a los interfaces propietarios y cerrados, sobre los cuales será posible desarrollar aplicaciones por terceros.
- Modularidad de la solución que permita desplegarla por partes de forma sencilla, sin necesariamente, hacer uso del procesamiento avanzado y las capacidades de la capa de inteligencia. Véase la Figura 2.22.
- Capa de Adquisición única, válida para distintos servicios.
- Independencia de Tecnología de acceso y sensores. Compatibilidad de diferentes redes de acceso y protocolos M2M.
- Se requiere soporte de protocolos abiertos, traducción de protocolos u otra

solución, para independizar la plataforma de la complejidad de los dispositivos y permitir el acceso a sensores de diferentes fabricantes.

- Capacidad para añadir nuevos conectores cuando se vayan estandarizando.

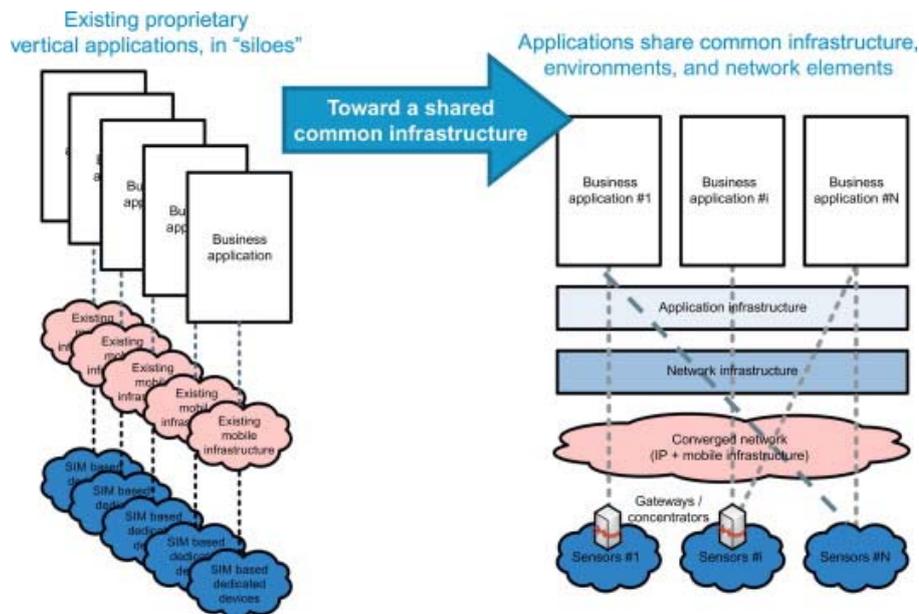


Figura 2.22 Modelo de arquitectura horizontal M2M ETSI

- Acceso a sensores de la plataforma, como un bus de eventos sobre el cual se accede la información de los sensores y dispositivos inteligentes a través de la red de transporte.

2.4.2.3 Capa de Conocimiento

La Capa de Conocimiento contiene los elementos de Tratamiento, Gestión y Explotación de la información y se encarga de:

- Acceso a toda la información tanto histórica como en tiempo real.
- Movimiento de datos recibidos desde la capa de adquisición, entre las distintas funciones de la Capa de Conocimiento para su almacenamiento, proceso y recuperación, así como hacia la Capa de Interoperabilidad. Los datos en esta capa ya están abstraídos de los dispositivos que lo generaron y deben poder tratarse siguiendo modelos estándares de datos.
- Soporte Tratamiento Tiempo Real de los datos recibidos desde la Capa de Adquisición a través de módulos como motores CEP, motores reglas...
- Soporte tratamiento Batch de los datos recibidos a través de procesos ETL,

Machine Learning.

- Soporte Tratamiento Analítico de los datos a través de procesos BI, etc.
- Soporte Tratamiento GIS de los datos recibidos permitiendo georreferenciar estos, hacer consultas geográficas, etc.
- Seguridad en el acceso a los datos, de modo que se controle que usuario/rol está accediendo a cada dato.

2.4.2.4 *Capa de Interoperabilidad*

La Capa de Interoperabilidad facilita la prestación de los servicios en el ámbito de la Ciudad Inteligente ofreciendo interfaces y funcionalidades, como son el Kit de desarrollo y el Open Data, que serán utilizadas para implementar los servicios que serán entregados a los clientes:

- Publicar APIs que pueden consumirse desde la Capa de Servicios, siendo interesante incluir el concepto de API Manager.
- Capacidad de interconexión entre aplicaciones y entre plataformas.
- Acceso desde la Plataforma a servicios externos.
- Publicar datos abiertos a través de un Portal/Repositorio Open Data.
- A través de un Kit de Desarrollo que incluye SDK y APIs permite construir Servicios dentro de la Capa de Servicios.
- Y todo ello con la seguridad integrada en el acceso a APIs, Kit de desarrollo, Open Data, etc.
- Sobre la base de un conjunto de APIs basados en estándares la capa de Interoperabilidad debe garantizar la portabilidad de aplicaciones entre ciudades y entre plataformas, de tal forma que se cree un verdadero ecosistema de aplicaciones con masa crítica y que baje la barrera de acceso a los desarrolladores de aplicaciones.
- Las APIs expuestas por la Capa de Interoperabilidad serán de fácil uso por la comunidad de desarrolladores, por lo que siguiendo la tendencia general deberá ser un API REST.
- Las APIs deben soportar distintos modos de acceso a los datos, incluyendo el modo Push (suscripción y notificación) y el Pull (petición y respuesta). También se deberán soportar consultas geo-referenciadas.
- El modelo de acceso a los datos ofrecido por el API será agnóstico respecto al modelo concreto de datos, pero para permitir la interoperabilidad debe utilizar un formato de transporte compatible con cualquier modelo existente. Se recomienda seguir el modelo OneM2M como referencia.

2.4.2.5 *Capa de Servicios*

La Capa de Servicios cubre los servicios y aplicaciones de negocio y valor añadidos. Interactúa con la plataforma a través de la Capa de Interoperabilidad.

Dentro de los servicios que pueden ser soportados por la plataforma están:

- Los Centros de Mando personalizados para diferentes ubicaciones de despliegue en función del perfil y de los permisos de los usuarios.
- Las Aplicaciones de gestión de servicios verticales como Movilidad, Eficiencia Energética, Riego Inteligente...
- Aplicaciones de gestión de contratos integrados, incluyendo SLA en base a datos reales. Los cuales incluyen funcionalidades como son:
 - Cuadro de mandos e indicadores.
 - Sistemas de predicción, simulación y planificación.
 - Sistema de tratamiento de datos.

Las aplicaciones de Ciudades Inteligentes involucran sectores como:

- Energía y sostenibilidad medioambiental.
- Gestión de edificios públicos y restantes infraestructuras urbanas.
- Transporte y movilidad.
- Comercio.
- Seguridad.
- Gobierno ciudad y relación con ciudadanos.
- Educación, cultura.
- Turismo.
- Ocio.

2.4.2.6 *Capa de Soporte*

La Capa de Soporte funciona de forma transversal dando soporte al resto de funcionalidades ofreciendo servicios como auditoría, monitorización, seguridad, etc.

- Repositorio de configuración de la Plataforma de modo que exista un lugar centralizado de almacenamiento de toda esta.
- Entorno Web de Gestión de la configuración permitiendo a través de una aplicación Web la gestión de toda esta, incluyendo además interfaces REST de gestión.
- Conectores con Repositorios de Seguridad de modo que la seguridad de la plataforma pueda delegarse al gestor de usuarios de la ciudad (LDAP).

2.4.2.7 *Métricas de la Plataforma SSC*

Las métricas de la Plataforma SSC permiten valorar cualitativamente (y no cuantitativamente) las capacidades de una Plataforma de una forma global.

- Grado de adecuación al modelo de capas y funcionalidades de la Plataforma modelo para la operación de los servicios de ciudad.
- Modularidad de la Plataforma: capacidad para instalar ciertos módulos en un momento inicial e ir ampliando componentes y extendiendo funcionalidades.

- Integración con otras Plataformas capacidad para importar o exportar datos a otras plataformas.
- Basarse en estándares abiertos: toda la plataforma debe estar sustentada en estándares abiertos, garantizando la escalabilidad y modularidad. Dentro de este criterio pueden encajar plataformas open-source y comerciales.
- Protocolos IoT soportados: debe basarse en protocolos y tecnologías estándares frente a tecnologías propietarias.
- Capacidad de extensión de la Plataforma: en este ámbito la tecnología cambia muy rápidamente por lo que la plataforma debe poder extenderse para ir añadiendo nuevas capacidades, protocolos...
- Soporte Enfoque Big Data en una Ciudad Inteligente se generan un volumen ingente de datos por lo que es imprescindible este soporte, pudiendo evaluarse e este punto aspectos como la diferenciación entre datos en tiempo real e históricos, o el soporte Hadoop.
- Soporte enfoque Open Data: La Plataforma debe incluir acceso a los datos open data según estándares abiertos. Las métricas deberán evaluar, por ejemplo, aspectos como el soporte a estándares como rdf, oData, XM, o el uso de productos como CKAN.
- La plataforma podría prestar servicio en modelo On premise y/o Cloud, en función de los requisitos del cliente. Las métricas podrán incluir la valoración del Soporte al enfoque Cloud basado en estándares.
- Inclusión capacidades GIS permitiendo georreferenciar la información y consultar y representar esta información.
- Inclusión de herramientas que permitan trabajar con la Plataforma de forma sencilla y productiva, así como configurarla.
- Niveles de disponibilidad y de nivel de servicio: RTO (Objetivo de Tiempo de Recuperación), RPO (Objetivo de Punto de Recuperación) y otros parámetros que puedan considerarse.
- Garantía, soporte y hoja de ruta de la plataforma para asegurar el uso productivo presente y futuro de la Plataforma.

2.4.3 *Plataforma de Ciudad del Ayuntamiento de València (VLCi)*

La Plataforma de Ciudad Inteligente adoptada por el Ayuntamiento de València (Plataforma VLCi) está basada en la Plataforma *THINKING CITY* desarrollada por Telefónica. La Plataforma VLCi incluye un conjunto de módulos que permiten la adquisición, el procesado, el almacenamiento y la distribución de datos, y su arquitectura se adapta al modelo genérico de capas descrito en el apartado anterior.

Esta plataforma se basa en los estándares e interfaces desarrollados en el marco del proyecto europeo *FIWARE* [54]. La documentación más detallada sobre la plataforma y

como interactuar con los distintos componentes se encuentra disponible para usuarios cualificados.

La Figura 2.23 presenta los módulos e interfaces a alto nivel de la Plataforma Thinking City donde se han definido, en línea con la arquitectura genérica, un modelo de tres capas:

- Acceso y normalización.
- Almacenamiento y Análisis.
- Visualización

2.4.3.1 *IoT AGENTS*

Este componente recopila todas las observaciones que provienen de los dispositivos y las transforma en eventos NGSi disponibles en el Context Broker. De esta forma, las aplicaciones pueden consumir información de observaciones provenientes de los dispositivos y enviar comandos (en el caso de los dispositivos que soporten esta funcionalidad).

Cada IoTAgent proporciona la capacidad de interactuar con dispositivos con un protocolo concreto. Si se desea un modelado concreto (“mapping” de atributos) es necesario proporcionar las reglas a aplicar y que pueden exigir la provisión de los dispositivos para establecer un modelo concreto asociado a un dispositivo concreto. Para más información sobre los *IoTAgents* ver [55, 56].

2.4.3.2 *Context Broker (ORION)*

El *Context Broker* es un módulo que recupera, mantiene e introduce información de contexto dentro de la plataforma *VLCi*. Es una implementación del Generic Enabler Publish/Subscribe *Context Broker* definido en *Fi-Ware*, proporcionando interfaces NGSI. Usando la interfaz NGSI los clientes pueden realizar varios tipos de acciones:

- Creación de elementos de contexto, representando entidades en el sistema, por ejemplo, un sensor determinado que mide temperatura.
- Actualizar información de contexto, por ejemplo, enviar nuevas actualizaciones de temperatura.
- Ser notificado cuando tiene lugar un cambio en la información de contextos (por ejemplo cambia el valor de la temperatura)
- Realizar consultas sobre la información de contexto. El *Context Broker* almacena información de contexto que va siendo actualizada desde las aplicaciones, de modo que las consultas se van resolviendo en base a estos datos.

Para más información sobre *Context Broker* se puede consultar la API NGSIv2 [57].

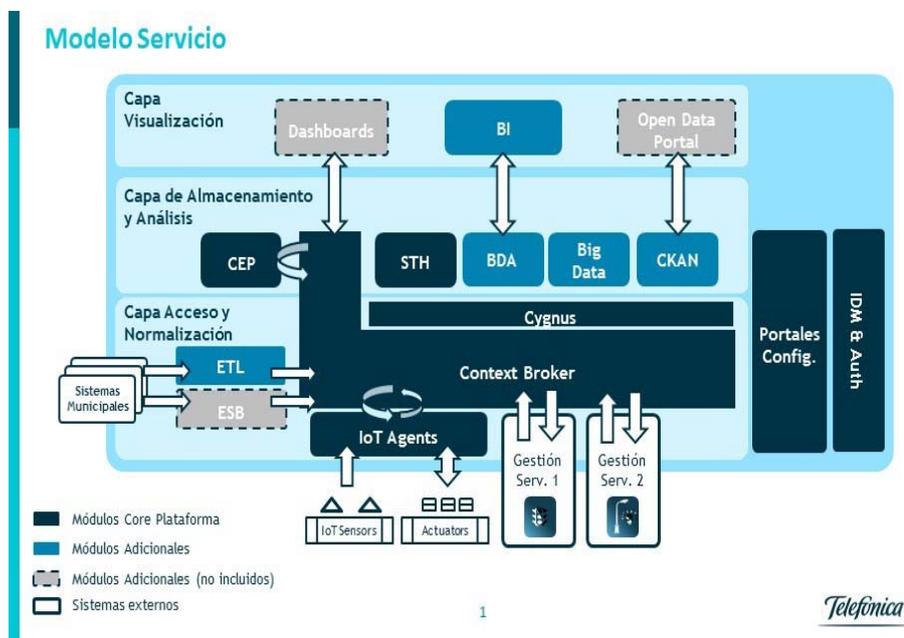


Figura 2.23 Descripción de alto nivel de la plataforma THINKING CITY de Telefónica

2.4.3.3 Adaptador de datos NGSI (CYGNUS)

Cygnus es el componente que se encarga de procesar los datos enviados por el *Context Broker* en su formato de notificación y adaptarlos a los distintos sumideros de la plataforma, tales como MySQL, CKAN, Hadoop [58] o MongoDB [59]. *Cygnus* está basado en Apache Flume [60] y proporciona una gran versatilidad y una gran capacidad de configuración para poder enviar datos a sistemas finales para su almacenamiento y explotación. En líneas generales posee dos modos de funcionamiento:

Modo row – en el cual se almacenará en el sistema final una fila por cada atributo recibido. Este modo no precisa de provisión previa.

Modo column – en el cual cada notificación se almacena en una fila y cada atributo es una columna. Este modo precisa de pre-provisión en el sistema final.

Para más información sobre el conector *Cygnus* ver [61].

2.4.3.4 Open Data Portal (CKAN)

CKAN es una herramienta para portales web de datos abiertos [62]. Facilita la gestión y publicación de colecciones de datos. Es ampliamente usado por gobiernos nacionales, regionales y locales, centros de investigación y otro tipo de organizaciones que acumulan grandes cantidades de datos.

Una vez que los datos son publicados, los usuarios pueden utilizar búsquedas específicas, tanto a nivel de portal como mediante acceso por API, para navegar y encontrar los datos que necesitan, así como previsualizarlos utilizando mapas, gráficas y tablas.

2.4.3.5 *Histórico a corto plazo (STH)*

El histórico a corto plazo es un módulo de la plataforma *VL*Ci que permite el almacenamiento de los datos procesados por el *Context Broker* y permite su posterior recuperación tanto en forma de datos crudos como de datos agregados. Proporciona una API que permite realizar distintos filtros y agregaciones de datos para facilitar la explotación de los datos almacenados en la plataforma.

Para más información sobre STH ver [63].

2.4.3.6 *Base de datos analítica (BDA)*

La base de datos analítica (ADB por sus siglas en inglés) pone a disposición de la capa de presentación los datos almacenados internamente en la plataforma.

De este modo, la base de datos analítica asegura que el tiempo de espera para el usuario que quiere visualizar el dato en la capa de presentación sea mínimo.

2.4.3.7 *Herramienta de BI*

En el caso del escenario de Valencia, la capa de presentación se basa en la herramienta de BI, MicroStrategy [64]. De este modo, la BDA almacenará cualquier dato susceptible de ser necesitado por la capa de presentación, de modo que pueda ser fácilmente presentado a los usuarios finales.

2.4.3.8 *Herramientas de carga de información e integración con sistemas externos - ETL*

Dado que hay datos almacenados en fuentes externas que han de ser cargados en la plataforma, hay dos módulos encargados de obtener información de estas fuentes, transformarla en un formato compatible con la plataforma y almacenarla en los diferentes sumideros de datos de la plataforma. Dentro de la plataforma se proporcionan los mecanismos para construir los trabajos ETL y un ESB. La combinación de estas dos herramientas permite la carga de datos en tiempo real (ESB) o batch (ETL), siempre que haya conectividad con las fuentes.

La herramienta utilizada para construir los trabajos ETL es Talend [65,66] y para el ESB JBoss [67].

2.4.3.9 *Gestor de Eventos (CEP)*

El módulo CEP (Complex Event Processing,) analiza datos provenientes de eventos en tiempo real, infiere insights de forma inmediata y hace posible una respuesta instantánea a las condiciones cambiantes. Mientras que las aplicaciones reactivas estándares están basadas en reacciones a eventos aislados, el CEP reacciona a situaciones y no a

eventos singulares. Una situación es una condición que se basa en una serie de eventos que ocurren dentro de una ventana de tiempo dinámica a la que llamamos contexto de procesado. Las situaciones incluyen eventos compuestos (por ejemplo, agregaciones), así como operadores de ausencia. El módulo CEP se basa en Esper.

Para más información sobre CEP ver [68].

2.4.3.10 *Big Data (HADOOP)*

La solución de Big Data para IoT se basa en Apache Hadoop, que es un framework de código libre para el almacenamiento y el procesado a gran escala de conjuntos de datos en clústeres de hardware de consumo.

El Framework de Apache Hadoop se compone de los siguientes módulos:

- *Hadoop Common* – contiene librerías y utilidades necesarias para otros módulos de Hadoop.
- *Hadoop Distributed File System (HDFS)* – un sistema de ficheros distribuido que almacena datos en equipos de consumo, proporcionando un ancho de banda agregado muy alto a nivel de clúster.
- *Hadoop YARN* – una plataforma de gestión de recursos responsable de la gestión de los recursos de computación disponibles en los clústeres y de su utilización para el scheduling de las aplicaciones de los usuarios.
- *Hadoop MapReduce* – un modelo de programación para el procesamiento de datos a gran escala.

Para más información sobre la herramienta de Big Data ver [58].

2.4.3.11 *IDM&AUTH*

La plataforma soporta un esquema de gestión de identidad, autenticación y autorización basada en tres elementos principales: IDM (Identity Manager), PEP (Policy Enforcement Point), PAP/PDP (Policy Administration Point/Policy Decision Point). Estos elementos se encuentran encarnados en los siguientes elementos de la arquitectura de plataforma:

- *PEP Proxy*: Artefacto especializado por cada GE/Componente encargado de interceptar las peticiones al componente en cuestión y de forzar al cumplimiento de los requisitos de Identificación y Autorización como paso previo a la utilización del componente. El PEP proxy es el encargado de orquestar toda la comunicación entre el IDM y PDP.
- *IDM (Keystone)*: Artefacto que contiene la información relativa a Usuarios, Roles y Perfiles. También se hace cargo de la emisión y validación de tokens, así como de los mecanismos de Autenticación.
- *KeyPass*: Motor de gestión y evaluación de políticas XACML, es el responsable de autorizar las peticiones, decidiendo que acciones/peticiones están

permitidas, y cuáles no, para cada ROL-Componente.

Para más información ver [69, 70].

2.4.3.12 *Portal de administración/configuración*

El portal de administración/configuración proporciona un interfaz de usuario para facilitar determinadas operaciones de la plataforma tales como creación de servicios, subservicios y usuarios, así como la administración de entidades, dispositivos, suscripciones y reglas de cada uno de los servicios dados de alta. Se permite el acceso tanto a roles capaces de crear servicios como a administradores y usuarios de los servicios creados.

2.4.3.13 *Inserción/extracción de datos en la Plataforma*

Como se apuntó en apartados anteriores, la Plataforma VLCi dispone de un conjunto diferenciado de interfaces lógicas que permite una gestión más eficiente de la comunicación bidireccional de la Plataforma con dispositivos y otros sistemas.

La naturaleza de los datos a insertar por cada una de las interfaces no es una cuestión relevante desde el punto de vista de la plataforma, pudiendo integrarse por cada uno de los canales habilitados datos generados o procedentes de fuentes externas a la plataforma o de los propios servicios.

Se han identificado 4 grandes grupos de interfaz para la inserción de datos:

- a) *Datos procedentes de sensores*: datos procedentes de sensores o dispositivos que acceden al sistema a través de los IoTAgents. Los protocolos de comunicación con los dispositivos que están actualmente soportados por la plataforma para la recepción de medidas por parte de IoTAgents son UL2.0 y JSON, permitiendo como protocolos de nivel superior HTTP y MQTT. Si se necesita proporcionar soporte para otros protocolos, puede conseguirse desarrollando el correspondiente IoTAgent. El API para la provisión y configuración de los dispositivos en IoTAgents puede encontrarse en [xx].
- b) *Datos procedentes de procesos ETL*: datos procedentes de procesos de extracción, transformación y carga de/hacia otros sistemas con información que no requiere tiempo real (no-RT). Los datos de servicio pueden provenir de fuentes muy diferentes (datos internos a la ciudad, datos recopilados desde las compañías responsables de la gestión de aquellos servicios que hayan sido externalizados, etc.). La plataforma de VLCi resuelve el problema de la heterogeneidad de fuentes proporcionando los mecanismos para poder construir procesos ETL (Extracción, Transformación y Carga (Load)). De modo que, mientras haya conectividad a las fuentes de datos, estos datos pueden ser transformados y cargados en la plataforma VLCi.
- c) *Datos procedentes de aplicaciones ESB*: datos de aplicaciones conectadas a través de un bus de servicios (ESB). La funcionalidad es similar a la proporcionada

por los procesos ETL, adquisición de información proporcionada por servicios, contratas y agentes externos, pero en este caso, para procesos en tiempo real (RT).

- d) *Datos procedentes de aplicaciones NGSI*: datos de aplicaciones desarrolladas para interactuar directamente con el interfaz NGSI que expone el *Context Broker*.

Desde el punto de vista de gestión de los datos, la plataforma dispone de 4 puntos de almacenamiento diferentes, siendo los flujos de información también diferentes para ellos, aunque con puntos en común. Es necesario tener en cuenta que habrá una instancia de Cygnus para cada uno de estos almacenamientos, que son:

- Histórico a Corto Plazo (STH)
- Base de Datos Analítica (BDA)
- Portal de Datos Abiertos(CKAN)
- Big Data (HADOOP)

Capítulo 3

Fuentes de tráfico en una Smart City

3.1 Introducción

El tráfico de voz basado en conmutación de circuitos ha sido durante un largo período de tiempo el tráfico mayoritario en la red y sus características están perfectamente definidas bien analíticamente o con datos obtenidos de redes reales a lo largo de los años [70, 71]. Además, la propia naturaleza de la conmutación de circuitos, basada en una asignación de recursos dedicados, simplifica la operación, administración y diseño de la red.

Sin embargo, las redes convergentes basadas en la compartición de recursos y protocolo IP, se enfrentan a un número prácticamente ilimitado de nuevos servicios con patrones de tráfico y requerimientos de calidad de servicio diferentes (retardo, Jitter y pérdida de paquetes) que es necesario garantizar. Servicios de voz, vídeo y datos en NGN [72] tienen características muy diferentes al tráfico de voz cursado en redes de conmutación de circuitos.

En el caso concreto de una Smart City, la red se enfrenta a variaciones de tráfico muy elevadas de características desconocidas que pueden degradar la calidad de servicio del sistema y generar congestión. Las redes con conexiones oportunísticas [73, 18] como las redes móviles o de sensores son especialmente vulnerables a estos cambios por lo que se hace necesaria la caracterización de las diferentes fuentes de tráfico que intervienen en una conexión y su comportamiento agregado para garantizar la calidad de servicio recogida en las recomendaciones internacionales.

El reto de modelar estadísticamente fuentes de tráfico en una red convergente es una tarea de gran complejidad y conlleva un coste elevado. Existen básicamente tres enfoques diferentes de afrontar esta tarea:

- Modelos matemáticos que describen el comportamiento del tráfico de manera analítica. En general suelen ser preferidos por su mayor flexibilidad, pero son modelos aproximados especialmente en situaciones de número reducido de fuentes o fuentes heterogéneas.
- Modelos de simulación. En este caso existe una gran flexibilidad dado que las fuentes pueden ser desde distribuciones estadísticas hasta datos reales. En contra, suelen ser específicos y requieren en general gran cantidad de proceso.
- Modelos físicos (prototipos). Su campo de aplicación son pruebas de sistemas reales.

La metodología más utilizada, y en general la que resulta más efectiva para tráfico homogéneo, consiste en la monitorización pasiva de las fuentes de tráfico y la definición de un modelo estadístico adecuado para describir su comportamiento [74]. Esta solución permite adquirir un conocimiento bastante cercano de la fuente, pero solo es válida para un tipo particular.

Cuando el número de posibles tipos de fuentes crece, modelar analíticamente su comportamiento (distribución del tamaño de los paquetes y el tiempo entre llegadas) incluso de servicios homogéneos como pueden ser la voz o el vídeo, se convierte en una tarea compleja. Por ejemplo, una señal de vídeo codificada en H.265 necesitará por término medio un 40% menos de ancho de banda que si hubiese sido codificada en H.264 [89, 90]. En este sentido, en un entorno de tráfico heterogéneo, como son las redes convergentes, los modelos de simulación aportan mayor flexibilidad [75, 76].

Un reto aún mayor aparece cuando fuentes heterogéneas comparten el ancho de banda disponible. En este caso la complejidad de evaluar aspectos relacionados con garantizar la calidad de los propios servicios y la coexistencia de diferentes tipos de clases de servicio en la red llevó a la ITU-T a definir arquitecturas tipo para la realización de pruebas y evaluación de prestaciones [73]. Para ello se caracterizaron analíticamente diferentes fuentes de tráfico de uso común [77] y en los parámetros de calidad y disponibilidad para garantizar la calidad de servicios y prestaciones de la red [78, 79].

Este capítulo tiene como objetivo definir un modelo de red simplificado que emule el comportamiento de una red de telecomunicaciones municipal. Caracterizar la demanda de tráfico mediante el modelado de los diferentes tipos de fuentes que pueden existir en una Smart City e identificar los objetivos de calidad de servicio extremo a extremo.

3.2 Modelo de red simplificado

Para la definición de una red municipal arbitraria se ha escogido la versión más simplificada del modelo de red NGN [73], es decir el modelo de red dedicado según se refleja en el Apartado 9.2.1 (Dedicated model network) de la norma y cuya arquitectura básica se dibuja en la Figura 3.1

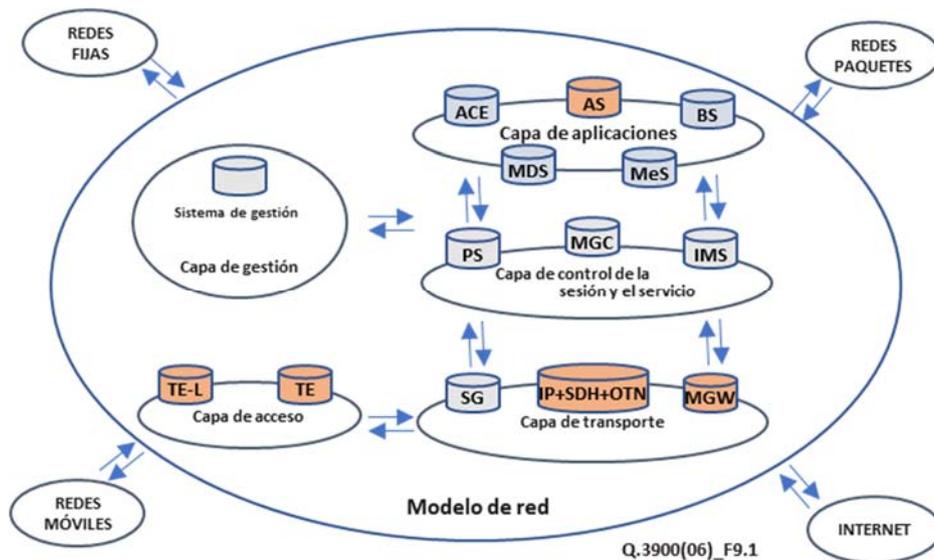


Figura 3.1 Arquitectura básica de un modelo de red dedicado

En esta arquitectura, y para los objetivos de esta tesis, solamente se considerarán los elementos funcionales relacionados con el Nivel 2 (Pruebas de redes), concretamente los Niveles 2.3, 2.4 y 2.5 relacionados con los servicios, el transporte extremo a extremo y la calidad del servicio. Por tanto, no se han considerado las funciones de establecimiento de sesión en las capas de control y gestión, asumiendo que han asignado la clase de servicio correspondiente a las fuentes durante el inicio de la sesión, es decir están ya configuradas. Por tanto, los elementos que se consideran representan un subconjunto de las funciones del plano de transporte: terminales (TE legacy y TE NGN), la red de transporte (IP+SDH+OTN), el Media Gateway (MGW) y la Plataforma de Ciudad que actúa como Servidor de Aplicaciones.

En la Figura 3.2 se muestra un hipotético entorno de comunicaciones municipal donde las diferentes fuentes de tráfico generan flujos de datos hacia la Plataforma de Ciudad, hacia otras redes (como la RDSI para el tráfico de telefonía a través de los Media Gateway) o a los servidores de aplicaciones del operador (AS NGN). La asignación de la Clase de Servicio a cada fuente de datos se realizará de manera prefijada en base a los requerimientos de calidad necesarios para garantizar la calidad requerida, y de acuerdo con las recomendaciones internacionales [78, 79].

En la figura se han reflejado diferentes bloques funcionales para representar las posibles fuentes de tráfico, su conectividad a la Red Corporativa Municipal y de ésta con el exterior. A continuación se describirán cada uno de ellos.

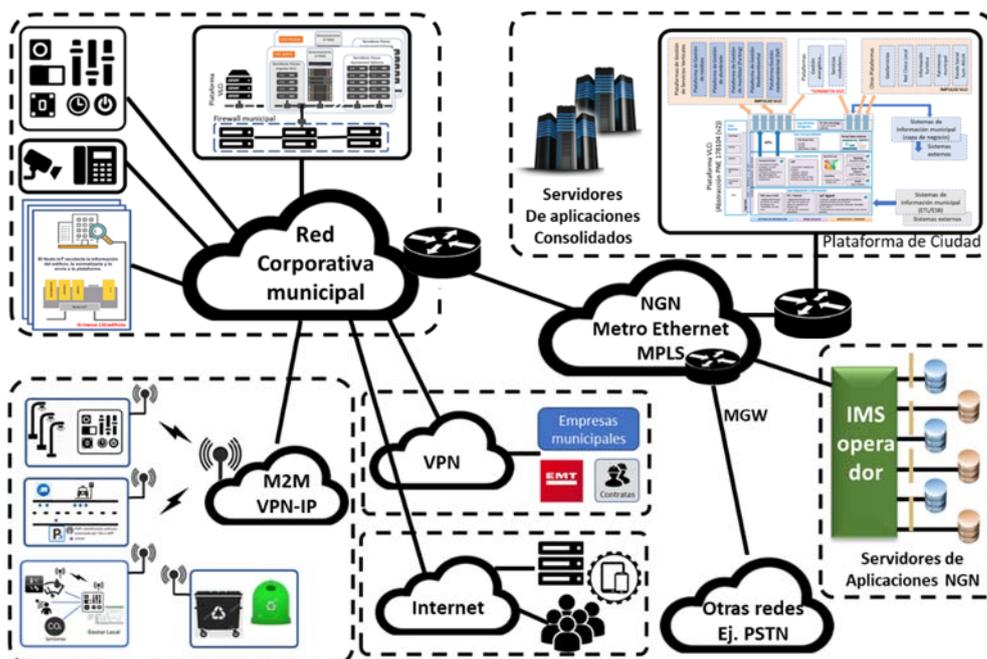


Figura 3.2 Arquitectura de comunicaciones típica de una Smart City

3.2.1 Usuarios y Dispositivos conectados a la Red Corporativa Municipal

La Red Corporativa Municipal (RCM) provee la conectividad entre los diferentes edificios y dispositivos conectados a una infraestructura de comunicaciones común, en muchos casos propietaria, en cesión de uso por parte de una operadora o mediante servicios de red privada virtual.

A esta red se conectan los diferentes servicios de conectividad y de control que utiliza el ayuntamiento en la ejecución de sus tareas diarias. Ejemplos de ello pueden ser los servicios de telefonía, el acceso a los servidores de aplicaciones en los Centros de Proceso de Datos municipales, las cámaras de video vigilancia, los sistemas de control semafórico, los sensores de tránsito de vehículos en las calles, etc. En general, la conectividad de dispositivos y servicios se realiza de manera cableada con fibra óptica, pares de cobre, o inalámbrica por enlaces radio (Wimax, radio fija o Wi-Fi) en función de los requerimientos de conectividad y los costes.

3.2.2 Acceso a dispositivos IoT a través de servicios móviles M2M VPN-IP

Este caso aplica a dispositivos IoT o sistemas complejos que se conectan a la infraestructura de comunicaciones municipal utilizando un servicio de red M2M VPN del operador.

Los dispositivos IoT que no pueden conectarse directamente a la RCM, por razones de movilidad, seguridad o cobertura, utilizan en general un servicio de red privada virtual M2M contratado al operador. Este servicio M2M emplea redes móviles 2/3/4G para conectarse con el dispositivo y encamina el tráfico a la RCM a través de un enlace de red fija basado en estándares Metro Ethernet o GPON.

Estos dispositivos pueden ser elementos simples, como los sensores de llenado en contenedores, o puertas de enlace de una capa de captación en la arquitectura IoT. Un ejemplo de este último escenario puede ser la gestión de alumbrado donde diferentes controladores de luminarias se conectan entre sí por tecnologías tipo ZigBee o PLC y con un controlador de segmento que a su vez establece conexión con la red corporativa municipal a través del servicio M2M VPN.

3.2.3 Acceso a servicios y dispositivos IoT a través de servicios VPN-IP

Este caso aplica a dispositivos IoT, sistemas complejos o usuarios (edificios) que se conectan a la infraestructura de comunicaciones municipal utilizando la red VPN de un operador.

Los dispositivos IoT que no pueden conectarse directamente a la RCM, por razones de gestión municipal, eficiencia o costes, utilizan en general un servicio de red privada virtual contratado al operador. Este servicio conecta de manera segura al proveedor municipal externo con la RCM encaminando el tráfico a la RCM a través de un enlace de red fija basado en estándares Metro Ethernet, GPON, xDSL, etc.

Esta conectividad se define sobre grupos de dispositivos IoT que son gestionados por terceros de manera individual y ofrecen un interfaz agregado. Ejemplos de este escenario pueden ser la gestión del consumo de agua a través de contadores inteligentes, la ubicación exacta de cada autobús en la ciudad o la gestión del servicio municipal de alquiler de bicicletas. En ambos casos, la información de los contadores, GPS's y Bus-CAN de los autobuses y estaciones de alquiler de bicicletas se recibe a través de este acceso, quedando oculta la arquitectura de comunicaciones utilizada para la captación de consumo de los contadores o la red para recibir en tiempo real la ubicación exacta de cada vehículo municipal o el estado de cada punto de anclaje de la red de estaciones de alquiler de bicicletas.

3.2.4 Acceso a servicios y dispositivos IoT a través de internet

Este caso aplica al resto de dispositivos IoT para sistemas sin especiales requerimientos de calidad, seguridad y disponibilidad, o conexiones esporádicas a la infraestructura de comunicaciones municipal.

La corporación municipal dispone de accesos internet para la relación con los ciudadanos, acceso remoto de miembros de la corporación y otros dispositivos o sistemas externos.

Los dispositivos IoT que no se conectan directamente a la RCM y que poseen escasos requerimientos de calidad, seguridad y disponibilidad, encaminan su tráfico a través del acceso a internet corporativo existente. Ejemplos de este escenario pueden ser sensores medioambientales embarcados en autobuses de la flota municipal, medidores de ruido urbano o dispositivos de los ciudadanos (smartphones). En estos casos, la escasa variabilidad de la información entre muestras aconseja un acceso de bajo coste.

3.2.5 Acceso a la Plataforma de Ciudad

Por último, la infraestructura tecnológica de la Plataforma de Ciudad reside en servidores externos a la Corporación Municipal conectados por un acceso Metro-Ethernet.

3.3 Tipos de tráfico en las redes de telecomunicaciones

La naturaleza y el volumen de tráfico cursado por la red incide directamente sobre su rendimiento, y por lo tanto en la calidad de servicio que ésta ofrece. La demanda de los usuarios (personas/sistemas) se puede modelar por las propiedades estadísticas del tráfico mediante modelos matemáticos que posteriormente han de ser comparados con valores reales.

Las propiedades del tráfico de la red se describen generalmente en términos de variables aleatorias para modelar la llegada de paquetes o llamadas y el tiempo de servicio en el sistema. En la Figura 3.3 se muestra de manera esquemática la terminología asociada a los procesos de tráfico.

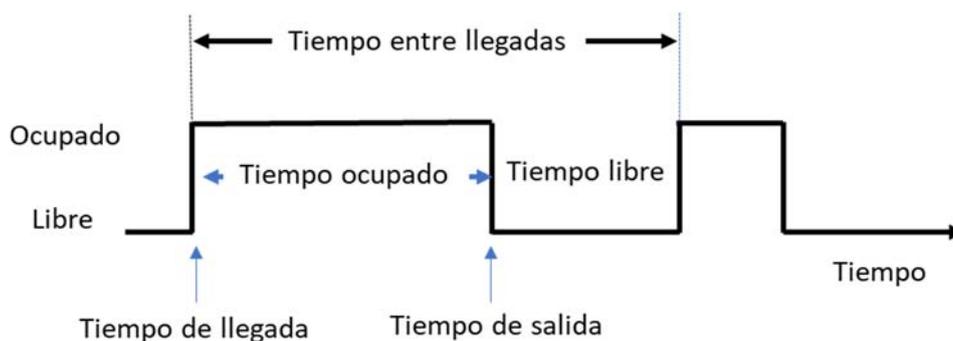


Figura 3.3 Terminología asociada a los procesos de tráfico

Podemos definir la intensidad de tráfico en un grupo de recursos como el número de recursos ocupados en un instante dado de tiempo. Los valores de valor medio y varianza

de la intensidad de tráfico pueden calcularse sobre un período de tiempo T. Por ejemplo, para la intensidad media de tráfico tendremos:

$$Y(T) = \frac{1}{T} \int_0^T n(t)dt \quad \text{Ecuación 3. 1}$$

Donde n(t) refleja el número de recursos ocupados en el instante t.

Definimos la tasa media de llegadas (λ) como el número medio de paquetes recibidos dentro de un intervalo de tiempo predefinido y el tiempo de servicio (μ) como el tiempo medio en el que el dispositivo o servidor están ocupados. El tiempo entre llegadas es por tanto ($1/\lambda$).

$$\lambda = \frac{\text{llamadas}}{\text{segundo}} \left(\frac{1}{s} \right) \quad \text{Ecuación 3. 2}$$

$$\mu = \frac{1}{d} (s^{-1}) \quad \text{Ecuación 3. 3}$$

donde d es la duración media o tiempo ocupado (en segundos).

El tráfico (A) se puede calcular como:

$$A = \lambda d = \frac{\lambda}{\mu} \quad \text{Ecuación 3. 4}$$

Si (A) es una variable aleatoria que modela el tiempo entre llegadas tendremos:

$$A(T) = P(A \leq T) = \int_0^T a(t)dt \quad \text{Ecuación 3. 5}$$

El tiempo medio entre llegadas se puede calcular como:

$$\frac{1}{\lambda} = \int_0^{\infty} ta(t)dt \quad \text{Ecuación 3. 6}$$

El tiempo entre llegadas se puede modelar como una distribución exponencial de parámetro λ

$$A(t) = 1 - e^{-\lambda t} \text{ y } a(t) = \lambda e^{-\lambda t} \quad \text{Ecuación 3. 7}$$

El valor esperado y la varianza se calculan como:

$$E(A) = \frac{1}{\lambda} \text{ y } varA = E(A^2) - E(A)^2 = \frac{1}{\lambda^2} \quad \text{Ecuación 3. 8}$$

Existe abundante bibliografía sobre ingeniería de tráfico y planificación de redes [80, 81, 82] para tratar de abordar el análisis estadístico de las fuentes. En muchos casos existe una correlación entre el comportamiento del tiempo entre llegadas de paquetes (que se puede describir como una función de distribución) y el tiempo de servicio [14]. Esta información podría utilizarse para tratar de predecir el comportamiento del sistema.

De manera muy resumida, podemos dividir el tráfico en dos grandes grupos, tráfico de Poisson y tráfico autosimilar.

3.3.1 Tráfico de Poisson

El proceso de llegada del tráfico de voz para las redes de telecomunicaciones con conmutación de circuitos, basado en la observación a largo plazo, es un Proceso de Poisson.

Las propiedades fundamentales de un proceso de Poisson son la estacionalidad, la independencia entre instantes de tiempo y la regularidad [80].

- El número de eventos dentro de un intervalo de tiempo de longitud fija es también un proceso de Poisson distribuido.
- Los intervalos de tiempo entre eventos consecutivos se distribuyen exponencialmente.

El número de llegadas (N) que ocurren en un tiempo T tiene una distribución de Poisson λT .

$$P(N) = \frac{(\lambda T)^N e^{-\lambda T}}{N!}$$

Ecuación 3. 9

El proceso de Poisson también es adecuado para la presentación de otros procesos de llegada, por ejemplo, consultas a internet.

3.3.2 Tráfico autosimilar

Los modelos de tráfico basados en procesos de Poisson dejan de ser válidos para explicar muchas de las características observadas en redes que transportan tráfico de imágenes, video e internet [84].

- No se cumple que los tiempos entre eventos sean independientes entre sí.
- Tampoco se cumple que la duración de un evento sea independiente de otros.
- El efecto de correlación entre estas variables puede persistir durante mucho tiempo.

Este fenómeno se puede representar adecuadamente mediante modelos de tráfico fractal o autosimilar. El proceso de tráfico de llegada se considera autosimilar si la función de distribución para el original y los procesos agregados son iguales. El proceso agregado de llegada se crea como la media del proceso original en bloques de tamaño “m” (el bloque es parte del proceso original). No siempre es fácil predecir la auto similitud del tráfico, especialmente teniendo en cuenta las diferentes escalas de tiempo en las que se puede evaluar. Muy a menudo, el patrón de tráfico es similar en diferentes escalas de tiempo

La característica más importante del proceso de llegada autosimilar es el parámetro Hurst [81]. El conocimiento del valor del parámetro Hurst brinda la posibilidad de generar el flujo autosimilar con características especificadas. Un pequeño valor del parámetro Hurst corresponde al conocido proceso de Poisson donde la varianza del proceso es igual al valor medio. Por el contrario, valores altos del parámetro Hurst indican una gran dependencia entre los paquetes. El valor medio y la varianza de algunas distribuciones

como Pareto, Log-Normal, Weibull son relativamente altos y por lo tanto de interés para ser utilizadas en el modelado de este tipo de tráfico.

3.3.2.1 Coeficiente de Hurst

El coeficiente o exponente de Hurst es una medida de independencia de las series de tiempo que fue estudiada inicialmente por el científico británico Harold Edwin Hurst (1880-1978), como elemento para distinguir series fractales. Hurst descubrió que muchos fenómenos naturales exhiben un comportamiento que puede ser caracterizado por un proceso aleatorio sesgado, en el cual existe “memoria de largo plazo” entre las observaciones, es decir, que los eventos de un periodo influyen en todos los siguientes.

Posteriormente, Benoit B. Mandelbrot generalizó su trabajo y lo llamó análisis de rango reescalado (R/S), definido como un método estadístico utilizado para evaluar la ocurrencia de eventos poco comunes, dando origen a una herramienta ideal para procesos físicos, financieros, y demás, por lo que puede ser usado en cualquier serie de tiempo.

El coeficiente de Hurst, (H) podría calcularse de la siguiente manera:

$$D(X^{(m)}) = m^{2(H-1)}D(X) \quad \text{Ecuación 3. 10}$$

$D(X^{(m)})$ es la varianza del flujo agregado y $D(X)$ es la varianza del flujo original.

La siguiente ecuación muestra la relación entre varianza (D) y el coeficiente de Hurst (H)

$$\ln\left\{\frac{D(X^{(m)})}{D(x)}\right\} = (2H - 2)\ln(m) \quad \text{Ecuación 3. 11}$$

$$\ln\left\{\frac{D(X^{(m)})}{D(x)}\right\} = f(\ln(m)) \quad \text{Ecuación 3. 12}$$

3.4 Clasificación de las Fuentes de tráfico

Existe un cierto consenso en la literatura sobre la clasificación de las fuentes de tráfico según el tipo de servicio al que están asociadas y sus requerimientos, es decir las características de transferencia de información y las características de calidad.

Sin embargo, para el objeto de esta tesis se realizará una clasificación atendiendo a la naturaleza de la comunicación, humano-humano o máquina-máquina.

Las fuentes HTC o comunicaciones humano-humano están asociadas a los servicios de comunicaciones de voz como la telefonía, y multimedia como los servicios de difusión, videovigilancia, videoconferencia, etc.

Las fuentes MTC o M2M están asociadas a comunicaciones entre sistemas y dispositivos o sistemas entre sí como pueden ser sistemas de captación, sensores, actuadores, acceso a bases de datos, transferencia de ficheros, etc.

Atendiendo a los requerimientos de los flujos de información podemos establecer otros tipos de clasificación:

- Real Time. Se basa en la adquisición y transmisión en tiempo real de información, imprescindible en aplicaciones y/o situaciones críticas. Debe garantizarse un retardo mínimo y un ancho de banda suficiente. Las aplicaciones de tiempo real en redes NGN/IP normalmente utilizan la capa de transporte UDP.
- Store and Forward y Best Effort. Agrupa a aquellas aplicaciones que no son de tiempo real, no poseen requerimientos de retardo o de ancho de banda, pero sí son susceptibles a pérdidas. Las aplicaciones no tiempo real en redes NGN/IP normalmente utilizan la capa de transporte TCP.
- Tasa de bit constante o variable, garantizada o no.
- Unidireccional, bidireccional, punto a punto, punto a multipunto.
- Etc.

Por último, atendiendo a los criterios de calidad de la información.

- Máximo retardo. El tiempo máximo que la información tarda en atravesar la red entre dos puntos de acceso a la misma. Este valor máximo dependerá de la naturaleza del servicio subyacente.
- Jitter o variación del retardo. Se refiere a la variación del retardo entre dos puntos de acceso a la red. Esta información afecta especialmente a determinados servicios y su máximo valor dependerá, como en el párrafo anterior, de la naturaleza del servicio subyacente.
- Tasa de error. Se refiere a la relación entre el número de bits recibidos erróneos y el total de los bites transferidos. Su máximo valor dependerá de la naturaleza del servicio subyacente.
- Tasa binaria. Se refiere a la cantidad de bits transmitidos por unidad de tiempo.

3.5 Human Type Communication (HTC)

Dentro de las comunicaciones HTC destaca por su importancia el tráfico de voz que en redes convergentes posee características particulares.

Los elementos clave que van a determinar el impacto del retardo o pérdidas, están relacionadas con el tipo de códec utilizado para codificar la señal y también con la implementación del sistema final. Existe un compromiso entre ancho de banda y calidad percibida según los estándares de calidad basados en “Mean Opinion Score”, (MOS).

Generalmente una comunicación HTC de voz mayoritariamente es una conversación telefónica donde el emisor envía ráfagas de voz (talk spurts) seguido de silencios intercalados entre palabras y frases o escucha. Estas características, como la supresión de silencios, pueden aprovecharse para mejorar la eficiencia disminuyendo el ancho de banda utilizado.

3.5.1 Características de las Fuentes de voz en redes TDM

El tráfico de voz en redes TDM está perfectamente caracterizado en las recomendaciones internacionales [72]. La intensidad del flujo de tráfico en Erlangs y los intentos de llamadas de hora ocupada (BHCA) se muestran en la Tabla 3.1 para carga normal de nivel A. Para el caso de condiciones de sobrecarga, se asume que el tráfico se aumente un 25% en Erlangs y un 35% en BCHA (apartado 2.1.2.2 de UIT-T Q.543).

Tabla 3.1 Características del tráfico de voz

Clases de flujo de tráfico	Tipo de distribución	Intensidad de tráfico (Erlang)	Número medio de BHCA
1	Poisson	0.03	1.2
2	Poisson	0.06	2.4
3	Poisson	0.10	4
4	Poisson	0.17	6.8

Fuente: ITU-T Q.543

3.5.2 Características de las Fuentes de voz en redes convergentes

Las características de flujo de tráfico de voz en NGN están determinadas por la transferencia de tráfico de voz a través de protocolos IP (VoIP). Las características de tráfico de voz de la Tabla 3.1 en cuanto al comportamiento de los usuarios se han demostrado muy estables y podrían utilizarse igualmente para la intensidad de tráfico promedio de VoIP, así como el número promedio de intentos de llamadas en hora cargada (BHCA).

La observación estadística de los flujos de tráfico de VoIP en las redes reales ha demostrado que el flujo de tráfico de VoIP puede describirse mediante el proceso de Poisson por lo que el modelo de flujo de tráfico de Poisson puede utilizarse para el tráfico de VoIP [84].

El tráfico VoIP se transporta generalmente como un flujo codificado de datos usando RTP (Real-time Protocol) sobre UDP. Los factores clave que van a determinar el impacto que van a tener los factores de calidad que afectan al nivel del servicio, como retardo o pérdidas, están relacionados con el tipo de códec utilizado y también en la implementación del sistema final.

Los códecs más utilizados están basados en los estándares de la ITU G.71x y G.72x. El códec más sencillo, G.711 es un codificador de onda que usa una modulación por pulsos (PCM), donde la señal analógica se muestrea a intervalos regulares. Estas muestras se cuantifican para producir la señal digital. Códecs más avanzados como G.729 utilizan codificaciones diferenciales adaptativas (ADPCM) que consiguen la misma calidad que G.711 pero con menos ancho de banda. Otros códecs como G.729 y G.723.1 utilizan técnicas más complicadas como ACELP (Algebraic Code Excited Linear Prediction) que consiguen disminuir la tasa binaria final. En la Tabla 3.2 se listan las características de los códecs más utilizados.

Tabla 3.2 Tipos de Codecs más comunes para VoIP

ITU-T Codec	Codec type	Maximum codec delay (ms)	Bitrate (bps)	Packetization Interval (ms)	pps	Payload size (bytes)	IP pkt size (bytes)	IP bps
G.711	PCM	0.375	64 000	10	100	80	120	96 000
G.711	PCM	0.375	64 000	20	50	160	200	80 000
G.711	PCM	0.375	64 000	30	33.33	240	280	74 659
G.723.1	ACELP	97.5	5 300	30	33.33	20	60	15 998
G.723.1	ACELP	97.5	5 300	15	16.67	40	80	10 669
G.726.16	ADPCM	0.375	16 000	10	100	20	60	48 000
G.726.16	ADPCM	0.375	16 000	20	50	40	80	32 000
G.726.16	ADPCM	0.375	16 000	30	33.33	60	100	26 664
G.726.24	ADPCM	0.375	24 000	10	100	30	70	56 000
G.726.24	ADPCM	0.375	24 000	10	50	60	100	40 000
G.726.24	ADPCM	0.375	24 000	10	33.33	90	130	34 663
G.726.32	ADPCM	0.375	32 000	10	100	40	80	64 000
G.726.32	ADPCM	0.375	32 000	20	50	80	120	48 000
G.726.32	ADPCM	0.375	32 000	30	33.33	120	160	42 662
G.726.40	ADPCM	0.375	40 000	10	100	50	90	72 000
G.726.40	ADPCM	0.375	40 000	20	50	100	140	56 000
G.726.40	ADPCM	0.375	40 000	30	33.33	150	190	50 662
G.728	LD-CELP	1.875	16 000	10	100	20	60	48 000
G.728	LD-CELP	1.875	16 000	20	50	40	80	32 000
G.728	LD-CELP	1.875	16 000	30	33.33	60	100	26 664
G.729A	CS-ACELP	35	8 000	10	100	10	50	40 000
G.729A	CS-ACELP	35	8 000	20	50	20	60	24 000
G.729A	CS-ACELP	35	8 000	30	33.33	30	70	18 665

Los códecs mencionados anteriormente generan diferentes patrones de tráfico a velocidades que oscilan entre 5 y 64 kbps dependiendo del tipo utilizado. Además, la detección de la actividad de la voz también influye en el flujo de paquetes. Por lo tanto, las fuentes de voz muestran diversas características de tráfico y por lo tanto requisitos diferentes de calidad de servicio.

Para caracterizar este conjunto de fuentes la metodología más utilizada, como se apuntó en la introducción, consiste en la monitorización pasiva del tráfico y la generación de un modelo estadístico adecuado para describir su comportamiento.

El tráfico de voz a nivel de fuente en una conversación basada en VoIP se caracteriza por la concatenación de un período de actividad vocal o conversación (ON) seguido de

un período de silencio o pausa (OFF). Durante el período ON, la fuente envía paquetes a intervalos regulares de longitud T (tiempo de paquetización). La duración de cada estado vendrá determinada por una variable aleatoria y una distribución asociada. [73, 84].

Este comportamiento ha sido caracterizado para diferentes tipos de idiomas y demuestra, que la duración del ciclo de conversación y la duración de la pausa presentan una distribución exponencial de parámetros α y β [19, 85,103]. Una fuente de tráfico VoIP podría describirse como un proceso de Markov ON-OFF con tres parámetros:

- La duración media del período ON: $T_{ON} = 1/\alpha$. Donde α es el parámetro de la función exponencial del período ON.
- La duración media del período OFF: $T_{OFF} = 1/\beta$. Donde β es el parámetro de la función exponencial del período OFF.
- La tasa de transmisión de paquetes durante el período ON: $\lambda = 1/T$

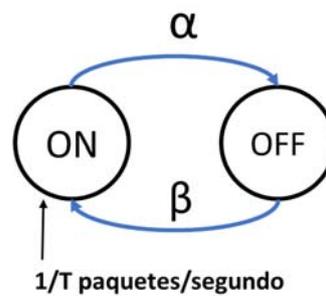


Figura 3.4 Modelo ON-OFF VoIP

Puede calcularse la tasa media de la fuente $\bar{\lambda}$ en paquetes por segundo para todos los estados:

$$\bar{\lambda} = \frac{T_{ON}}{T(T_{ON}+T_{OFF})} = \frac{\beta}{T(\alpha+\beta)} \quad \text{Ecuación 3. 13}$$

Un análisis sobre la distribución estadística de los tiempos ON-OFF para fuentes de VoIP se muestra en [18] donde se ha analizado el tráfico de voz en una red 3G con diferentes tipos de códecs. Para ello se han recogido los tiempos de llegada de paquetes de una conversación telefónica y después de un filtrado para eliminar los valores atípicos (cabeceras de nivel de paquete y de transporte), se ha tratado de encontrar una distribución estadística cuyo comportamiento encaje aproximadamente con las muestras recibidas.

Utilizando un códec GSM 06.10 y una llamada de duración 1 minuto, los principales parámetros estadísticos se muestran en la Tabla 3.3 y la densidad de probabilidad de los tiempos entre llegadas se muestra en la Tabla 3.4.

Tabla 3.3 Parámetros estadísticos en una red 3G para un códec GSM

VoIP, GSM	Media (ms)	Varianza (ms)
Sin filtro	12.6	15675
Después de eliminar valores atípicos	10.16	55.26

Tabla 3.4 Distribución de los tiempos entre llegadas de paquetes para un códec GSM

VoIP, GSM	Función densidad de probabilidad de tiempo entre llegadas
Sin filtro	Distribución Exponencial $\lambda = 0.8$
Después de eliminar valores atípicos	Distribución Gamma, shape = 0.55, escala = 18.17

La Figura 3.5 muestra la función de distribución de probabilidad de los tiempos de llegada entre paquetes. Se puede observar que la mayor parte de los paquetes se generan, como es natural, cuando la voz está activa. Los tiempos entre la llegada de los paquetes se muestran en el eje x. El tamaño medio del paquete es 90 bytes y la tasa media de transmisión es 55 kb/s.

En cambio, los resultados para un códec G.711 ley A con la misma duración de llamada serán diferentes debido a que el tamaño medio de los paquetes es de 240 bytes (más de 2.5 veces el tamaño para GSM) y la tasa binaria es de 75 kbps. con un intervalo de paquetización de 30 ms. En la Tabla 3.5 se muestran los principales parámetros estadísticos y en la Tabla 3.6 se muestra la densidad de probabilidad de los tiempos entre llegadas.

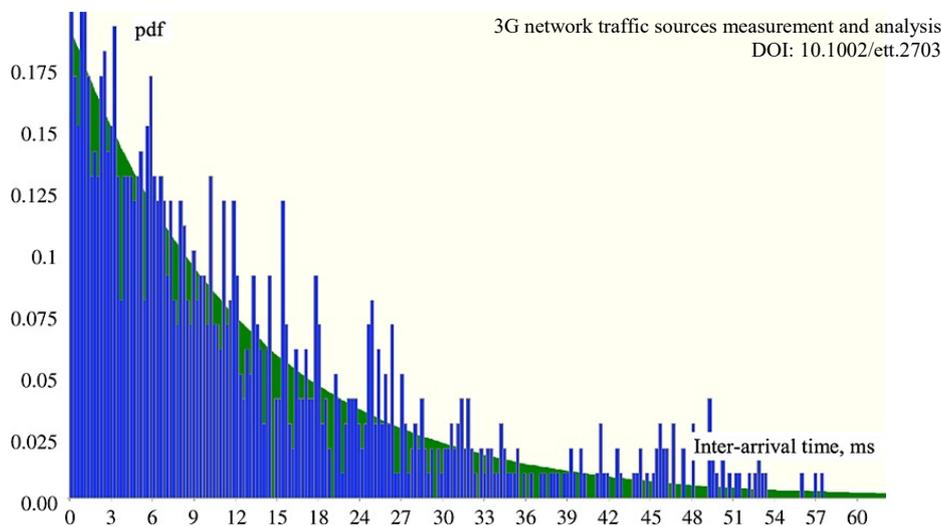


Figura 3.5 Densidad de probabilidad medida y evaluada de los tiempos de llamada de paquetes para una llamada de VoIP (GSM codec) sin filtrado, y una distribución exponencial.

Tabla 3.5 Parámetros estadísticos en una red 3G para un códec G.711 ley A

VoIP, G.711 ley A	Media (ms)	Varianza (ms)
Sin filtro	22.7	235125
Después de eliminar valores atípicos	10.23	52.482

Tabla 3.6 Distribución de tiempos entre llegadas de paquetes en un códec G.711 ley A

VoIP, G.711 ley A	Función densidad de probabilidad de tiempo entre llegadas
Sin filtro	Distribución Gamma $\kappa = 0.2292$, $\theta = 99.11$
Después de eliminar valores atípicos	Distribution Gamma location = 0, shape = 0.354, scale = 28.35

La Figura 3.6 muestra la función de distribución de probabilidad de los tiempos de llegada entre paquetes sin filtrado. Se puede observar que la práctica totalidad de los paquetes están concentrados en pequeños intervalos cuando la voz está activa. Los tiempos entre la llegada de los paquetes se muestran en el eje x y siguen una distribución Gamma.

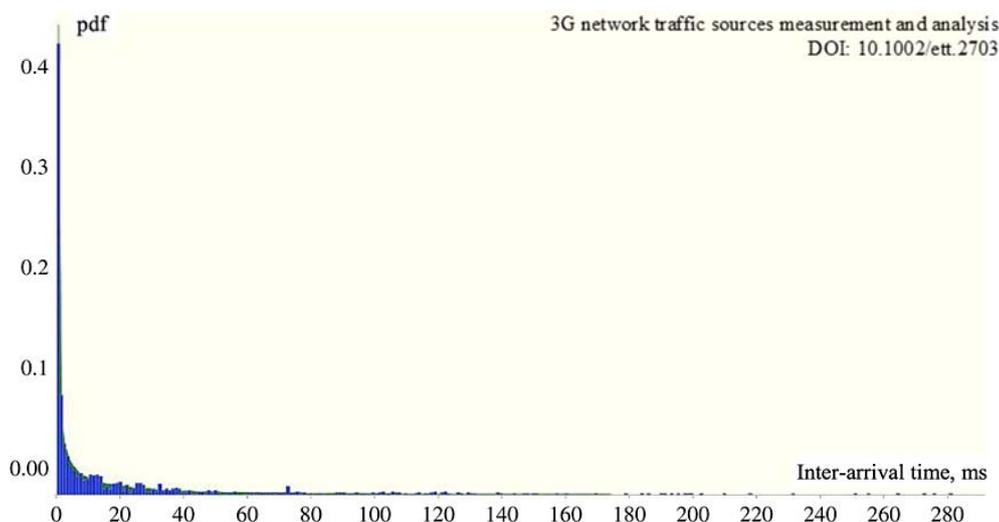


Figura 3.6 Densidad de probabilidad medida y evaluada de los tiempos de llamada de paquetes para una llamada de VoIP (códec G.711) sin filtrado, y una distribución Gamma.

Por último, para el caso de un códec G.729 con la misma duración de llamada el tamaño medio de los paquetes es de 60-80 bytes y la tasa binaria es de 48kb/s. Los principales parámetros estadísticos se muestran en la Tabla 3.7 y la densidad de probabilidad de los tiempos entre llegadas se muestra en la Tabla 3.8.

Tabla 3.7 Parámetros estadísticos en una red 3G para un códec G.729

VoIP, G.729	Media (ms)	Varianza (ms)
Sin filtro	12.2	14417
Después de eliminar valores atípicos	9.714	41.85

Tabla 3.8 Distribución de los tiempos entre llegadas de paquetes para un códec G.729

VoIP, G.729	Función densidad de probabilidad de tiempo entre llegadas
Sin filtro	Log-normal distribution, location = -2.63, $\mu = 10.02$, $\sigma = 8.61$
Después de eliminar valores atípicos	Distribution Gamma location = 0, shape = 1.06, scale = 8.98

La Figura 3.7 muestra la función de distribución de probabilidad de los tiempos de llegada entre paquetes sin filtrado. Se puede observar que la mayor parte de los paquetes se generan, como es natural, cuando la voz está activa y el perfil de las muestras se ajusta a una distribución log-normal.

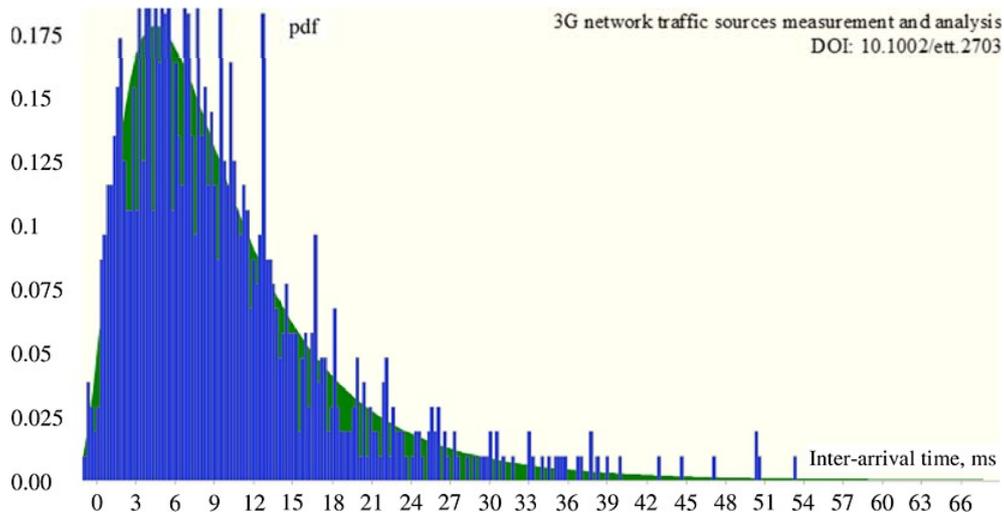


Figura 3.7 Densidad de probabilidad medida y evaluada de los tiempos de llamada de paquetes para una llamada de VoIP (G.729 codec) sin filtrado, y una distribución log normal.

3.5.3 Caracterización propuesta para una fuente de voz

La caracterización de una fuente de voz se diseñará como un flujo de datos periódico según se refleja en la Figura 3.8 donde la variable aleatoria T_{activo} representa el tiempo de conversación y T_{inactivo} representa el tiempo de línea libre entre dos llamadas consecutivas.

Durante el tiempo de conversación, la variable aleatoria T_{on} representa el tiempo de actividad vocal (talk spurts) y T_{off} representa el tiempo de silencio. La tasa binaria y tamaño del paquete complementan la definición de la fuente.

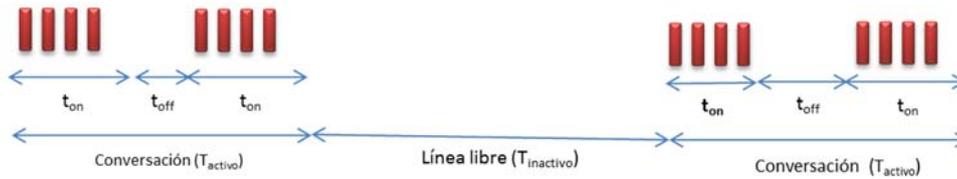


Figura 3.8 Modelado de una fuente de voz

En la definición del códec se definen las variables aleatorias T_{on} , T_{off} , la tasa binaria y tamaño del paquete. Por lo tanto, será únicamente necesario seleccionar el tipo de códec donde se generarán los paquetes de datos según la función de distribución correspondiente para los instantes T_{ON} y T_{OFF} .

El perfil de la comunicación se modelará mediante funciones de distribución de variable aleatoria tanto para la duración como para el tiempo de inactividad. En la Tabla 3.10 se resumen los parámetros referenciados.

Tabla 3.9 Parámetros para la caracterización de una fuente de voz

Parámetro	Tipo	Descripción
PacketSize	Cód- dec	Indica el tamaño del paquete de voz (B)
SendInterval		Indica el tiempo entre el envío de un paquete de voz y el siguiente (ms)
Duration	Perfil	Indica la duración de la llamada => Tactivo (minutos)
TimeInactive		Indica el tiempo durante el cual la fuente no genera tráfico => Tinactivo (minutos)

En la Figura 3.9 se muestra un ejemplo para una fuente de VoIP utilizando la herramienta SimulCity que se describirá en el siguiente apartado con los siguientes parámetros:

- Codec = G711 (PacketSize = 172B, SendInterval = 0.02 seg.).
- Duration (minutos) = distribución exponencial, media = 120 min = 60 max = 180.
- TimeInactive (minutos) = distribución exponencial, media = 60 min = 0 max = 120.

Puede observarse que la duración de la llamada ha sido de 110 segundos y también la distribución de los tiempos T_{ON} y T_{OFF}

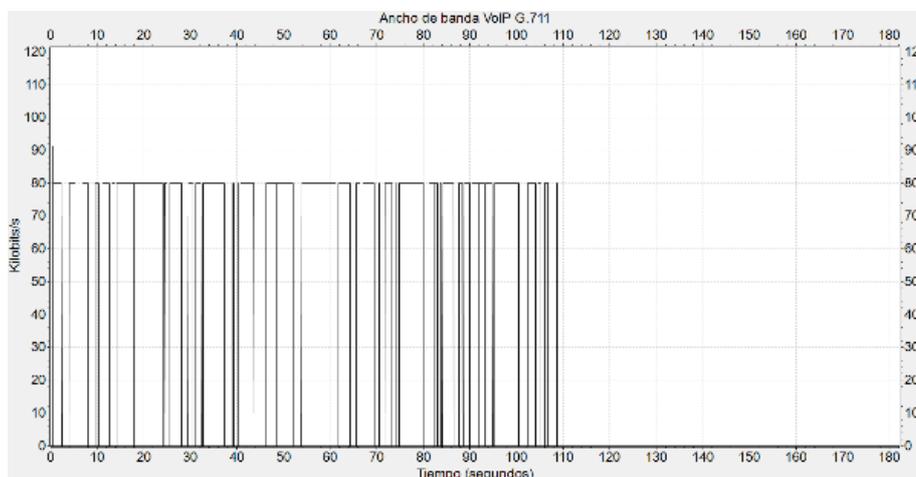


Figura 3.9 Comportamiento de una fuente de voz durante 3 minutos con un códec G.711

Este comportamiento ha sido caracterizado en [85, 19, 86] y se puede considerar para la generación de tráfico VoIP que la duración del ciclo de conversación y la duración de la pausa presentan la distribución exponencial. En la conversación, la duración media de la pausa compartida es del 61.47% y la duración media del intercambio de silencio mutuo es del 22.48%.

3.5.4 Características de las fuentes de vídeo en redes NGN

Las características de las fuentes de vídeo en las redes NGN dependen fundamentalmente del tipo de códec utilizado. El vídeo sobre IP se transporta como un flujo de datos mediante RTP sobre UDP.

La observación estadística del flujo de tráfico IPTV ha demostrado que éste tiene un comportamiento autosimilar tanto para el tráfico *multicast* como el *unicast* [8]. Los valores sugeridos del parámetro de Hurst se muestran en la Tabla 3.10

Tabla 3.10 Características del flujo de tráfico IPTV.

Tipos de tráfico	Tipos de flujos	Valores de Hurst
Multicast	Auto-similar	0.55 – 0.6
Unicast	Auto-similar	0.75 – 0.8

UIT-T Q.3925

Existen múltiples estándares de codificación de vídeo, siendo los más relevantes aquellos basados en MPEG (Motion Picture Expert Group). Los códecs MPEG generan tráfico de tasa de bits variable (VBR) siendo los más extendidos MPEG -2 (H.261) [87], MPEG-4 AVC (H.264) [88, 89] y MPEG-H (H.265) [92, 93] siendo este último el que genera

mayores eficiencias de codificación, pero su presencia es aún escasa. El códec H.264 es actualmente el más extendido en el mercado.

A grandes rasgos, un codificador H.264 convierte y comprime una señal de vídeo en una serie de cuadros, (I, P y B) (ver Figura 3.10).

- Cuadro I. Contiene un cuadro completo de vídeo, sin ninguna referencia a otros cuadros.
- Cuadro P. Usa información de cuadros anteriores (un cuadro I u otro cuadro P), por lo que ocupa menos que un cuadro I.
- Cuadro B. Usa información tanto de cuadros anteriores como de cuadros posteriores, por lo que permite una mayor compresión.

Los cuadros se agrupan en estructuras conocidas como GOP (Group of Pictures). Hay muchas estructuras de GOP posibles, y la disposición de las tramas I, P y B en el GOP está determinada por el formato de la señal de vídeo original (que determina la ratio de compresión). Cada GOP tiene un cuadro I, y típicamente de 2 a 14 cuadros P y de 2 a 10 cuadros B.

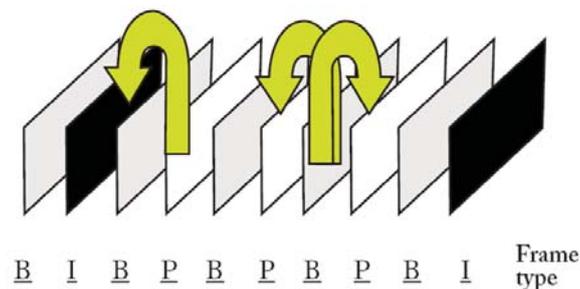


Figura 3.10 Estructura de un GOP, con cuadros I, B y P

En este caso, a diferencia de los códecs de voz, una fuente de vídeo es un generador continuo con tasa de bit variable. Esta tasa resultante del codificador es un proceso aleatorio que depende *de* la actividad de la escena. Krunz and Makowski [92] introdujeron el modelo $M/G/\infty$ para describir las correlaciones en el tráfico de vídeo del MPEG. Utilizando el modelo $M/G/\infty$ se pueden producir secuencias del MPEG a nivel de GOP, pero en este modelo es necesario conocer el tipo de correlaciones y la distribución del tamaño de GOP o cuadros.

El modelo $M/G/\infty$ representa el proceso de ocupación de la cola, donde el proceso de llegadas es exponencial; la disciplina de servicio es general (definida por su media y varianza) y el número de servidores es infinito. El proceso de ocupación X_n representa el número de clientes en el comienzo del intervalo temporal $[nT, (n+1)T]$, donde T es el intervalo de tiempo que representa el tiempo de generación del GOP en el modelo de

video. El proceso X_n resultante es estacionario y correlado. La correlación puede ser de corto o largo rango (SRD o LRD).

Con esta aproximación, el tamaño de las secuencias generadas por X_n no corresponde a los tamaños reales del GOP. La secuencia X_n generada sigue una distribución de Poisson de parámetro $\lambda E(\sigma)$, los dos parámetros λ y σ corresponden a M y G respectivamente.

Diferentes estudios estadísticos muestran que la función de autocorrelación (ACF) en el tráfico de vídeo se puede aproximar por una de las funciones descritas en la Tabla 3.11 donde k representa el paso de correlación y b debe ser mayor que cero:

Tabla 3.11 Funciones de autocorrelación para el tráfico de vídeo

Tipo	ACF
LRD	$\rho(K) = (k + 1)^{-b}$
SRD	$\rho(K) = e^{-b \sqrt{k}}$
SRD	$\rho(K) = e^{-bk}$

El modelo $M/G/\infty$ asociado con una secuencia de vídeo se caracteriza por los parámetros siguientes:

- Escala de tiempo o ranura de tiempo: a nivel de cuadro o GOP.
- Distribución del tamaño del GOP o Cuadro.
- Tipo de correlación: SRD o LRD.
- Distribución del tamaño de Paquete.

La Tabla 3.12 muestra los parámetros del tráfico de vídeo para tres películas codificadas con MPEG1, MPEG2 y MPEG4, la duración media del vídeo es 30 min y el tamaño de los paquetes utilizados son constantes de 1000 bytes. El patrón de tráfico contiene sólo un período ON.

Tabla 3.12 Parámetros del tráfico de vídeo

CÓDEC	Ranura de tiempo	ACF	Cuadro o GOP distribución por tamaño (Byte)
MPEG 1 (VBR)	GOP 12/25s	$\rho(K) = e^{-0.35 \sqrt{k}}$	LogNormal Media=141350 Var=2.1544e+09
MPEG 2 (VBR)	Cuadro 1/25s	$\rho(K) = e^{-0.055 \sqrt{k}}$	LogNormal Media=21015 Var=1.1443e+08
MPEG 4 (VBR)	GOP 12/25s	$\rho(K) = e^{-0.154 \sqrt{k}}$	Gamma+LogNormal Media=9265 Var=2.9235e+07

Sin embargo, este modelo no considera determinados aspectos como la calidad del vídeo (resolución), la tasa de refresco de la señal (cuadros) y la “cantidad de movimiento” de la imagen. La definición de estos parámetros puede permitir mayor flexibilidad en el diseño de la fuente de vídeo.

La resolución del vídeo determina el número de píxeles (unidad de medida de menor tamaño que forma parte de una imagen digital) en un cuadro, el cual es un múltiplo del ancho por el alto del cuadro. Las resoluciones que podrían utilizarse son: SD, HQ, HD, HD+ y 4K.

El número de cuadros por segundo (fps) representa cuántas imágenes fijas se reproducen en el vídeo por segundo. En cinematografía se utilizan 24 fps, el sistema Pal 25 fps y el NTSC 30fps.

Sería posible modelar una fuente H.264 a partir de la resolución, la cantidad de cuadros por segundo, y un factor que representa la cantidad de movimiento en la imagen, siendo 1 movimiento bajo (por ejemplo, una persona hablando sin moverse mientras la cámara y el fondo están quietos), 2 para movimiento medio (algún grado de movimiento, pero predecible y ordenado), y 4 para un movimiento alto. Este algoritmo denominado “The Kush Gauge” [93], permite obtener de manera muy sencilla el ancho de banda medio, máximo y mínimo de una fuente de vídeo H.264 en función de estos factores.

$$data\ rate\ \left(\frac{b}{s}\right) \approx 0.07 (res.x)(res.y) (frames.sec)(motion) \quad \text{Ecuación 3. 14}$$

$$minimum\ dr\ \left(\frac{b}{s}\right) \approx 0.0525 (res.x)(res.y) (frames.sec)(motion) \quad \text{Ecuación 3. 15}$$

$$maximum\ dr\ \left(\frac{b}{s}\right) \approx 0.0875 (res.x)(res.y) (frames.sec)(motion) \quad \text{Ecuación 3. 16}$$

En la Figura 3.11 se muestran las tasas binarias que se han calculado para cinco tipos de fuente de vídeo para diferentes resoluciones y grados de movimiento. Para cada resolución y movimiento se muestran los valores mínimos, medios y máximos generados por el códec H.264 VBR.

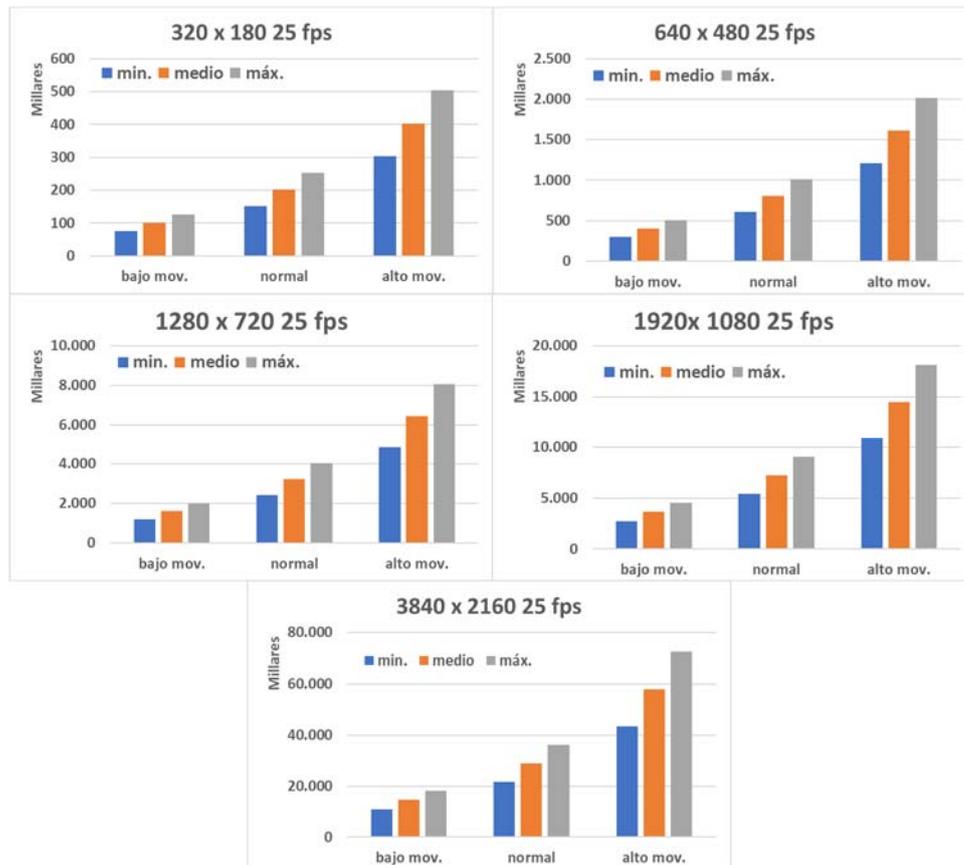


Figura 3.11 Tasas binarias para diferentes tipos de fuentes de vídeo.

3.5.5 Caracterización propuesta para una fuente de vídeo

La caracterización de una fuente de vídeo se diseñará como un flujo de datos periódico según se refleja en la Figura 3.12 donde la variable aleatoria T_{activo} representa el tiempo de actividad y T_{inactivo} representa el tiempo inactivo de la fuente.

Durante el tiempo de vídeo activo, el códec de vídeo generará una tasa de paquetes que seguirá una variable aleatoria acotada entre los valores definidos “mínimo y máximo” calculados en función del número de tramas por segundo y la calidad de la fuente.

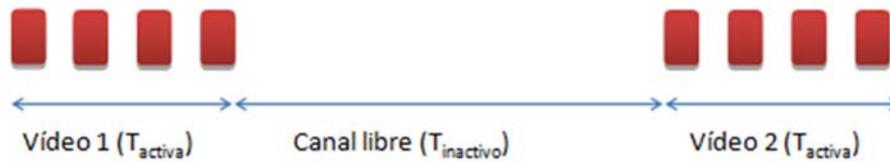


Figura 3.12 Modelado de una fuente de video H.264

En la Tabla 3.13 se muestra la tasa binaria del códec para diferentes resoluciones con un perfil de “poco movimiento” y 25 fps.

Tabla 3.13 Tasa binaria H.264 -bajo movimiento - 25 fps

Calidad	Resolución	Tasa binaria del códec
SD	180p (320x180)	75~150 kbps
HQ	360p (640x360)	300~600 kbps
HD	720p (1280x720)	1.200~2.500 kbps
HD+	1080p (1920x1080)	2.750~5.500 kbps
4K	4K (3840x2160)	10.886~18.144 kbps

El perfil de la comunicación se modelará mediante funciones de distribución de variable aleatoria tanto para la duración como para el tiempo de inactividad. En la Tabla 3.14 se describen los parámetros utilizados en la definición de la fuente.

Tabla 3.14 Parámetros para la configuración de una fuente de video

Parámetros	Tipo	Descripción
ResX	Resolución	Ancho del cuadro (en píxeles)
ResY		Alto del cuadro (en píxeles)
Duration	Perfil	Tiempo que dura la fuente de video - T_{activo} (minutos)
TimeInactive		Tiempo en el que la fuente no envía tráfico - $T_{inactivo}$ (minutos)
Movement		Factor que indica la cantidad de movimiento del video (de 1 a 4)
fps		Cantidad de cuadros por segundo

En la Figura 3.13 se muestra los resultados de simulación de una fuente de video H.264 de un hipotético servicio de Telepresencia (videoconferencia de alta definición y máxima calidad) utilizando la herramienta SimulCity que se describirá en el siguiente capítulo con los parámetros:

Parámetro	Valor
Codec	H.264
ResX	1920
Res Y	1080
Duration	distribución exponencial, media =5 min=2 max=10
TimeInactive	distribución exponencial, media=60 min=0 max=120
Movement	4
Fps	25
Tiempo simulación	300 (segundos)

La simulación proporciona un valor de pico de 16,872 Mb/s, una tasa media de 12,65 Mb/s y un valor mínimo de 8,441 Mb/s, valores que se sitúan aproximadamente entre un 15 y un 8% por debajo de los valores calculados por el algoritmo.

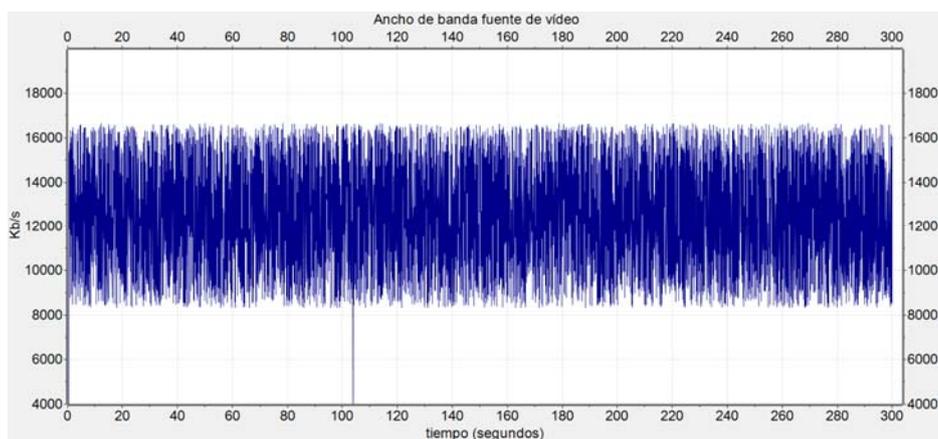


Figura 3.13 Fuente de vídeo Telepresencia durante 3 minutos con un códec H.264 HD+

3.6 Machine Type Communication (MTC)

A diferencia de las comunicaciones HTC, este tipo de comunicaciones también llamadas M2M (Machine to Machine) abarca la comunicación de datos entre entidades que no precisan necesariamente de la intervención humana [10].

El tráfico HTC presenta patrones muy diferentes al tráfico de voz y está experimentando un importantísimo crecimiento. Se estima que más de 50.000 millones de dispositivos estarán conectados en 2020 [94] por lo que probablemente se convierta en el tráfico dominante en la red.

En este apartado también se incluyen el tráfico derivado de la interconexión entre sistemas de información, el acceso a bases de datos, transferencia de ficheros, etc.

El soporte de gran cantidad de dispositivos con requerimientos específicos en redes que inicialmente no se diseñaron para ello, especialmente celulares, representa un importante reto en múltiples aspectos de la propia arquitectura de la red, como la señalización, la naturaleza del tráfico, las clases de servicio o el número de dispositivos por celda.

Como características generales del tráfico HTC podemos destacar [95]:

- Gran número de dispositivos.
- Patrones de tráfico con ciclo de trabajo (duty cycle) bajo, es decir un período largo entre dos transmisiones.
- La cantidad de paquetes transmitidos por dispositivo es “baja” y su tamaño es “pequeño”.
- El patrón de tráfico de los dispositivos muestra pequeñas variaciones.
- El tráfico generado por los dispositivos suele ser asimétrico con el enlace de subida superior al de bajada.
- Existen diferentes clases de servicio (best effort y tiempo real).
- Un dispositivo puede contener tráfico de múltiples fuentes (paquetes agregados).
- El tráfico de los dispositivos puede estar coordinado por requerimientos de la aplicación.

Las diferencias entre el tráfico HTC y el MTC es que el tráfico HTC exhibe un comportamiento heterogéneo mientras que el tráfico MTC es en general homogéneo (miles de dispositivos muestrean el entorno a una tasa determinada más o menos uniforme y generan un volumen de datos más o menos constante, o sistemas de almacenamiento en red actúan de manera similar). Además, las fuentes HTC son generalmente independientes (descoordinadas), mientras que en MTC las fuentes pueden coordinarse, es decir, muchos dispositivos pueden responder de manera simultánea ante determinados eventos [96].

Los primeros estudios sobre los servicios MTC ya indicaron que los modelos de tráfico diseñados para HTC no son suficientemente válidos para un entorno MTC y se hace necesario definir nuevos modelos de tráfico y escenarios de aplicaciones para su validación en las redes existentes [97, 98].

3.6.1 Aplicaciones y requisitos de servicio de redes MTC

Una primera aproximación a la definición de modelos de tráfico y escenarios MTC fue propuesta por el 3GPP [99], donde se identificaron diferentes escenarios de aplicación y requerimientos específicos de servicio.

3GPP ha clasificado 7 posibles escenarios de aplicación, cada uno de ellos con características específicas:

- **Seguridad:** Sistemas de vigilancia, Respaldo a líneas terrestres, Control de acceso físico, Seguridad coche/conductor, etc.

- Monitorización y Seguimiento: Gestión de flotas, Gestión de órdenes, monitorización de activos, Navegación, Información del tráfico, Peajes, Optimización del tráfico, etc.
- Pagos: Puntos de Venta (PoS), Máquinas expendedoras y Maquinas de juegos.
- Salud: Monitorización de constantes vitales, Apoyo a ancianos y discapacitados, Telemedicina, Diagnostico remoto, etc.
- Control y Mantenimiento Remoto: Sensores, Iluminación, Bombas, Válvulas, Control de ascensores, Control de máquinas expendedoras, Diagnostico de vehículos, etc.
- Medición: Energía, Gas, Agua, Calefacción, Control de redes eléctricas y Medición industrial, etc.
- Dispositivos de Consumo: Cámara digital, smartphone, Libro electrónico (eBook), etc.

Junto a estas áreas de aplicación 3GPP ha definido un conjunto de posibles “características” de una comunicación MTC en función de aspectos tales como movilidad, criticidad de la información, seguridad, costes, etc. En la última revisión de la especificación técnica (V14.0.1 de agosto de 2017) [95] se han reducido a 8 estos requerimientos de los 14 inicialmente definidos.

Estos tipos o comportamientos de dispositivos MTC que se describen a continuación persiguen conseguir una mayor eficiencia y reducción de costes.

- Movilidad reducida: aplica a dispositivos MTC que no se mueven, se mueven muy poco, o lo hacen en cierta localidad.
- Tiempo real (controlado): útil para aplicaciones que permitan enviar/recibir información durante períodos de tiempo definidos. La red no consume recursos de señalización innecesaria fuera de estos intervalos.
- Tiempo no real: El operador puede rechazar la solicitud de comunicación de un dispositivo en base a diferentes criterios y permitir su conexión en momentos de bajo tráfico.
- Baja transmisión de datos: Dispositivos para enviar o recibir pequeñas cantidades de datos utilizando la mínima señalización posible.
- Móvil Terminado Infrecuente (MT): Dispositivos MTC que principalmente utilizan acceso móvil y donde el operador puede reducir la frecuencia de la gestión de movilidad.
- Monitorización MTC: supervisar el comportamiento de un dispositivo MTC de acuerdo a sus características, como la pérdida de conectividad o fallos de comunicación.
- Conexión segura: asegurar la conexión entre dispositivos MTC y servidores MTC.
- Características MTC basadas en grupo: gestión optimizada de todos los dispositivos MTC que pertenecen a un mismo grupo MTC (políticas de control, direccionamiento IP y otras).

3.6.2 Modelos de tráfico MTC.

La primera cuestión que surge a la hora de diseñar modelos de tráfico MTC es verificar la viabilidad práctica de un modelo que incorpore miles de fuentes de entrada en una red multiservicio HTC+MTC [29]. La primera aproximación consistiría en tratar las fuentes HTC de manera individual como ocurre en los servicios de voz y vídeo, es decir, un “modelado por fuentes de tráfico”. Otra aproximación podría considerar la información agregada de todos los dispositivos MTC como un único flujo como ocurre en el modelado de enlaces troncales de alta velocidad con tráfico multiservicio, es decir, un “modelado por tráfico agregado”.

Los modelos de tráfico agregado han sido utilizados para modelar el tráfico MTC con la suposición de que los dispositivos son sencillos y muy numerosos. Este comportamiento puede ser descrito por un proceso de Poisson en el que la tasa de llegada (λ) puede cambiar con el tiempo $\lambda(t)$ debido a la posible sincronización del tráfico generado por los dispositivos [97]. Sin embargo, esta aproximación es más cuestionable cuando los dispositivos individuales son más complejos o menos numerosos y, por lo tanto, el modelo de tráfico agregado empieza a no ser válido [98]. Sin embargo, el modelado por fuentes de tráfico, pese a ser más realista, lleva asociado un elevado grado de complejidad [99].

3GPP en TR 43.868 [100] ha definido tres modelos de tráfico de fuentes agregadas, que incluyen el acceso a la red de manera descoordinada o sincronizada. En la Tabla 3.15 y en Tabla 3.16 se describen sus principales características.

Tabla 3.15 Modelos de tráfico del 3GPP TR 43.868

Modelo de tráfico	Descripción
T1	Dispositivos MTC que acceden a la red de una manera descoordinada / no sincronizada
T2	Dispositivos MTC que acceden a la red de forma coordinada / sincronizada con una cierta distribución
T3	Dispositivos heredados que acceden a la red de forma descoordinada / no sincronizada

Tabla 3.16 Configuración de los modelos de tráfico TR 43.868

Escenario	T1	T2	T3
Número de dispositivos	$\lambda /$ (Intervalo de medida) (Nota 1)	X	$\lambda /$ (intervalo de medida) (Nota 1)
Proceso de llegada	Poisson Intensidad de llegada: λ [llegadas/segundo]	Distribución de eventos determinística de tiempo limitado.	Poisson Intensidad de llegada: λ [llegadas/segundo] Caso 1: $\lambda = 5$ para el tráfico CS Caso 2: $\lambda = 5$ para el tráfico CS y $\lambda = 15$ para el tráfico PS
Intervalo de medida	5 segundos 15 minutos 1 hora 1 día	ver Nota 2	N/A
Tamaño del bloque	10 bytes 200 bytes 1.000 bytes	10 bytes 200 bytes 1.000 bytes	N/A
NOTA 1: Esta hipótesis es aproximadamente cierta, siempre y cuando la duración de la sesión de datos sea más corta que el intervalo de medida. NOTA 2: Con este modelo de tráfico el intervalo de medida no se define, ya que el número de dispositivos es fijo y el acceso debe ser terminado por todos los dispositivos antes de que el acceso siguiente pueda ocurrir.			

El escenario T1 puede considerarse bastante realista, ya que, para una gran cantidad de usuarios, el proceso global de llegada se puede modelar como un proceso de llegada de Poisson, independientemente del proceso de llegada individual.

El escenario T2 modela el comportamiento cuando gran cantidad de dispositivos generan información al mismo tiempo. En este escenario se supone que los dispositivos MTC están sincronizados dentro de un intervalo de T segundos.

El escenario T3 modela el comportamiento de los dispositivos heredados que utilizan conmutación de circuitos (CS) y conmutación de paquetes (PS) donde los procesos de llegada general (separados para CS y PS) pueden modelarse como un proceso de llegada de Poisson, ya que se supone que los dispositivos se inician independientemente el uno del otro.

El objetivo general del escenario T3 es poder ser utilizarlo junto con el escenario T1 o T2, respectivamente, para evaluar el impacto del tráfico MTC en el tráfico de voz y vídeo.

De manera complementaria, y para evaluar las prestaciones de una red LTE bajo diferentes condiciones de carga, se definieron en TS-37.868 dos modelos de tráfico también de tipo agregado [101] según se describen en la Tabla 3.17.

Tabla 3.17 Modelos de tráfico MTC 3GPP (TS-37.868)

Características	Modelo 1	Modelo 2
Número de dispositivos MTC	1.000,3.000,5.000, 10.000 y 30.000	1.000,3.000,5.000, 10.000 y 30.000
Distribución de llegada	Uniforme en T	Beta en T ($\alpha=3, \beta=4$)
Período de distribución (T)	60s	10s

El modelo 1 de tráfico puede ser considerado como un escenario realista donde los dispositivos MTC acceden a la red de manera uniforme durante un período de tiempo de una manera independiente (no sincronizada). En cambio, el segundo modelo puede ser considerado como un escenario extremo en el que un gran número de dispositivos MTC acceden a la red de manera sincronizada.

En ambos casos el número máximo de usuarios se ha limitado a 30.000 que es el máximo número de dispositivos actualmente gestionables dentro de una celda [101].

Para el modelo 2, asumiendo que todos los dispositivos MTC se activan entre $t = 0$ y $t = T$, que intensidad de tráfico es una variable aleatoria con distribución $p(t)$ y que el número total de dispositivos MTC es N , el número de llegadas en el la “i” oportunidad de acceso viene dada por:

$$\text{Intensidad en el acceso}(i) = \left[N \int_{t_i}^{t_{i+1}} p(t) dt \right] \quad \text{Ecuación 3. 17}$$

Donde t_i es el tiempo de la oportunidad i , y $p(t)$ sigue una distribución Beta.

$$p(t) = \frac{t^{\alpha-1}(T-t)^{\beta-1}}{t^{\alpha+\beta-1} \text{Beta}(\alpha, \beta)} \alpha > 0, \beta > 0 \quad \text{Ecuación 3. 18}$$

Beta (α, β) representa la función Beta. Los valores de $\alpha = 3$ y $\beta = 4$ se han definido en la norma.

La distribución de los intentos de acceso de los dispositivos debería estar limitada al período T .

$$\int_0^T p(t) dt = 1 \quad \text{Ecuación 3. 19}$$

Estos modelos de tráfico agregado poseen limitaciones cuando ocurre alguna de estas circunstancias:

- El número de dispositivos no es suficientemente elevado.

- El comportamiento de los dispositivos no es homogéneo (dispositivos hiperactivos).
- Existe tráfico sincronizado que afecta al tráfico aleatorio.
- La red influye en el comportamiento del dispositivo (diferentes clases de servicio, precio por hora punta y valle, etc.).

Sin embargo, soslayar las limitaciones de los modelos de tráfico agregado implica que cada dispositivo MTC sea representado como una entidad independiente [96]. Los modelos de tráfico de origen que pueden incorporar la naturaleza coordinada del tráfico MTC [96] están en general diseñados para un número limitado de fuentes y según éstas aumentan la complejidad se multiplica. Por ejemplo, para N dispositivos se requiere una multiplicación de matrices $N \times N$ para cada intervalo de tiempo.

Existen diferentes aproximaciones para disminuir esta complejidad, como la propuesta por M.Laner en [96] utilizando un modelo de tráfico de Poisson cuya tasa varía de acuerdo con un proceso de Markov (MMPP). Este tipo de aproximaciones también han sido propuestas en la literatura para modelar tráfico como el de internet [106].

3.6.3 Caracterización propuesta para una fuente MTC

La caracterización del modelo de datos para una fuente MTC ha de perseguir encontrar un equilibrio entre flexibilidad y complejidad dada la gran variedad de comportamientos posibles para una fuente de estas características. Además, sería conveniente poder modelar fuentes agregadas que siguen los modelos definidos en 3.6.2.1.

La estructura de algunos de los modelos de fuente propuestos en la literatura [103] junto con diferentes implementaciones utilizando herramientas de simulación [104, 105] pueden servir de base para la confección del modelo propuesto. En el presente análisis se ha definido un modelo de fuente de tráfico basado en tres niveles: sesión, actividad y bloque.

- El nivel de sesión modela el proceso de llegada de fuentes al sistema. Los momentos en los que llegan las fuentes se pueden describir mediante una distribución de variable aleatoria de las llegadas en el tiempo con una tasa promedio (λ), sin memoria, independiente de llegadas previas. Cada fuente tiene además un período de actividad (μ) que depende de la cantidad de información a transmitir (número de paquetes para enviar) y del ancho de banda disponible.
- El nivel de actividad representa la aplicación como una sucesión detallada de períodos de actividad e inactividad. Los períodos de actividad vienen dados por una variable aleatoria que representa la transmisión de una estructura de datos definida y otra variable aleatoria que representa el tiempo entre transmisiones.
- El nivel de bloque define las características básicas de los bloques de información, es decir el volumen de datos.

En la Figura 3.14 se muestra un esquema simplificado del modelo donde la totalidad de los parámetros de configuración pueden ser variables aleatorias con una distribución específica. Como ejemplo ilustrativo, una fuente MTC puede generar tráfico durante un número finito de sesiones separadas entre sí por un tiempo de inactividad (TON-TOFF). A su vez, cada sesión está compuesta por un número finito de “bloques” que se transmiten separados por un “intervalo de envío”. Por último, cada “bloque” está compuesto por un número variable de “textos e imágenes” que a su vez pueden tener un tamaño variable. Esto permite modular el tamaño de los bloques con mayor flexibilidad.

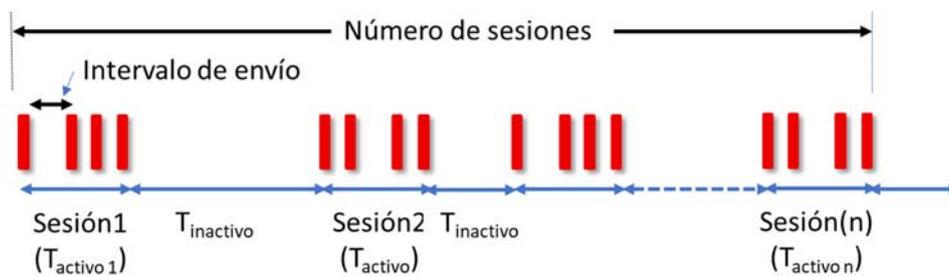


Figura 3.14 Modelado de una fuente MTC

El comportamiento de esta fuente tipo se puede modelar con 8 variables, 4 de ellas están relacionadas con el tamaño de los paquetes a transmitir, y 4 relacionadas con la distribución del tráfico. En la Tabla 3.18 se muestran los parámetros utilizados para la configuración de la fuente MTC.

Tabla 3.18 Parámetros definidos para la configuración de una fuente MTC

Parámetro	Tipo	Descripción
textResourceSize	Tamaño bloque	Tamaño (en bytes) de cada recurso de texto.
imageResourceSize		Tamaño (en bytes) de cada imagen.
numResources		Número de recursos de texto que contiene el bloque
numImages		Número de imágenes que contiene el bloque.
numPackets	Ráfagas	Número de bloques que se envían en cada sesión.
sendInterval		Tiempo que transcurre entre el envío de un fichero y el siguiente (s).
numBursts		Número de sesiones.
BurstInterval		Tiempo en el que la fuente no genera tráfico (s).

En la Figura 3.15 se muestra un ejemplo de modelado para una fuente MTC utilizando la herramienta SimulCity que se describirá en el siguiente capítulo. Para ello se ha configurado la fuente MTC con los parámetros definidos de manera arbitraria en la Tabla 3.19.

Tabla 3.19 Parámetros de configuración de una fuente MTC arbitraria.

Parámetro	Valor
-----------	-------

textResourceSize	Variable aleatoria distribución exponencial acotada, media = 20 bytes, min =10 y max =30 bytes.
imageResourceSize	Variable aleatoria distribución exponencial acotada, media = 5.000 bytes, min =2.000 y max =10.000 bytes.
numResources	Valor constante = 1.
numImages	Variable aleatoria de distribución uniforme entre 0 y 1.
numPackets	Variable aleatoria de distribución uniforme entre 0 y 5.
sendInterval	Variable aleatoria de distribución uniforme entre 1 y 2.
numBursts	Variable aleatoria de distribución uniforme entre 0 y 5.
BurstInterval	Variable aleatoria distribución exponencial acotada, media = 60 segundos, min =30 y max =300.

La simulación realizada (1.200 segundos) muestra una fuente periódica asíncrona en el entorno de los 60 segundos que genera bloques de datos según el histograma de la Figura 3.16 donde se puede comprobar que el número de bloques enviados en cada intervalo coincide con la especificación de la fuente.

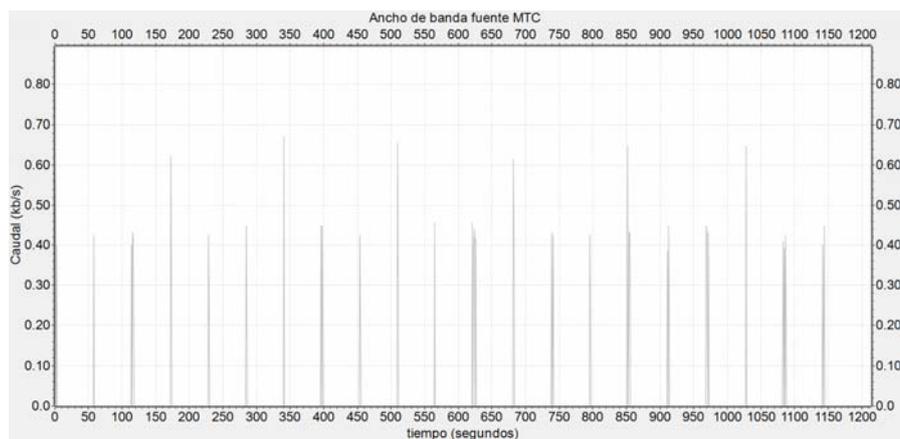


Figura 3.15 Ancho de banda de la fuente MTC definida

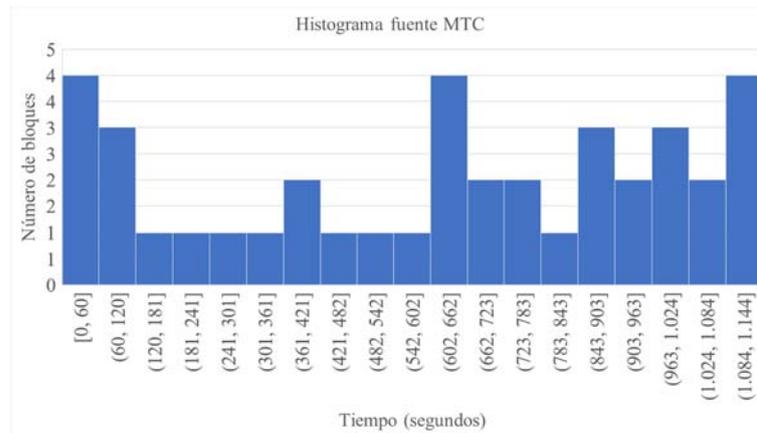


Figura 3.16 Histograma de bloques enviados en el período de simulación

3.7 Calidad de servicio (QoS)

En apartados anteriores se ha definido un modelo de red genérico municipal, se han analizado los diferentes tipos de tráfico que pueden estar presentes en una Smart City, y se ha modelado tres tipos de fuente, VoIP, vídeo y MTC.

El siguiente reto consiste en garantizar la calidad de los propios servicios y la coexistencia de diferentes tipos de clases de servicio en la red municipal. Servicios de voz, vídeo y datos poseen diferentes requerimientos de calidad como el retardo, el jitter y la pérdida de paquetes que es necesario garantizar.

La infraestructura de comunicaciones municipal está construida en muchos casos por una red óptica que interconecta los diferentes edificios municipales entre sí (ver Figura 3.2) por lo que la evaluación de la calidad de servicio tiene un claro sentido en los enlaces del operador donde existe un pago por ancho de banda y clase de servicio. En una Smart City, la conexión entre la red municipal y la Plataforma de Ciudad, normalmente alojada en un Data Center externo, es objeto del presente análisis.

Este apartado tiene como objetivo describir los objetivos de calidad de servicio extremo a extremo entre dispositivos conectados a la red corporativa municipal y una Plataforma de Ciudad en un entorno Smart City para identificar y definir los mecanismos que es necesario implementar en la herramienta de simulación SimulCity que se describirá en el siguiente capítulo.

3.7.1 Parámetros de medida de la calidad de servicio (QoS)

El término calidad de servicio (QoS) puede tener distintos significados. En primer lugar, un servicio solamente es útil si cumple unos requisitos de calidad que las aplicaciones de los usuarios finales requieren. Por lo tanto, el objetivo del servicio es maximizar la

satisfacción del usuario final con las aplicaciones que requieren de esos servicios, garantizando que funcionen correctamente.

Es posible asegurar que siempre hay más capacidad que carga de tráfico, de manera que el retardo, el jitter y las pérdidas serán mínimos, por lo que estos parámetros de calidad se cumplirán sobradamente. Sin embargo, asegurar que se cumplan todos los parámetros de calidad de todos los servicios tiene un coste muy elevado.

Se puede asociar la QoS a un problema de optimización, en el que se trata de maximizar la satisfacción del cliente final (las aplicaciones y los servicios funcionan bien), y a la vez minimizar los costes, asumiendo retardos y pérdidas en algunos servicios, pero siempre manteniendo una calidad mínima recogida en las recomendaciones internacionales [9, 10].

Los parámetros de medida de la calidad comúnmente definidos para los servicios basados en IP, como es el caso de esta tesis, son el retardo, el jitter y la pérdida de paquetes [40]. Estos parámetros forman parte de los Acuerdos de Nivel de Servicio (SLA) que utilizan los operadores de red como valores mínimos que la operadora se compromete a cumplir.

Con el fin de establecer un escenario lo más realista posible, se utilizarán las características de un servicio de red privada virtual ofrecido por Telefónica, Servicio Macrolan¹ [41], donde los parámetros definidos en los acuerdos de nivel de servicio son:

- Retardo extremo a extremo.
- Jitter.
- Pérdida de paquetes.
- Ancho de banda ocupado.

De manera complementaria, otros parámetros que forman parte de los acuerdos de nivel de servicio también contemplan:

- Disponibilidad del equipamiento de acceso.
- Tiempo de respuesta a averías.
- Tiempo de resolución de incomunicación.

¹ Macrolan es un servicio de interconexión de redes locales sobre infraestructura Metro Ethernet + IP (basada en tecnología MPLS) perteneciente a Telefónica que permite la creación de redes privadas virtuales sobre dicha infraestructura manteniendo las mismas prestaciones que una red privada.

3.7.1.1 Retardo

El retardo extremo a extremo caracteriza la diferencia de tiempo entre que se envía un paquete desde un punto de la red (generalmente la fuente), hasta otro punto de la red (generalmente el destino) y puede ser debido a diversos factores:

- **Retardo de propagación.** Es el tiempo que tarda un bit en viajar desde el puerto de salida hacia otro router. Dependerá principalmente de la longitud del enlace y del medio físico usado en el mismo. Suele tomar un valor de unos 4ms cada 1.000 km en un cable coaxial, y alrededor de 5ms. por cada 1.000 km en un cable de fibra óptica.
- **Retardo de conexión.** Es el tiempo que tarda un paquete desde que entra en un router hasta que sale de éste. Suele ser muy pequeño.
- **Retardo por colas.** Es el tiempo que tarda un paquete desde que entra en una cola hasta que sale de ésta. Este tipo de retardo se puede controlar mediante gestión de colas y mecanismos de planificación eficientes. Este es del elemento crítico a considerar.

En los servicios de tiempo real, este parámetro es especialmente importante. Algunos umbrales recomendados en las recomendaciones internacionales [108] se describen en la Tabla 3.20.

Tabla 3.20 Valores máximos de retardo por servicio (ITU-T E.800)

Retardo Tránsito	Servicio
< 50 ms.	Para servicios de voz (para evitar ecos y diafonías).
< 100 ms	Para aplicaciones interactivas que necesiten de la participación de personas.
< 150 ms	Para vídeo interactivo, conferencia multimedia de audio y vídeo, etc.
< 400 ms	En general, para cualquier aplicación en tiempo real.

En el caso del servicio Macrolan y a modo de ejemplo, en la Tabla 3.21 se incluyen los valores SLA de retardo de tránsito sobre un acceso, tomando como medida la media aritmética de los valores recogidos a lo largo de un día. Macrolan ofrece tres clases de servicio, clase multimedia para servicios de retardo crítico (voz), clase oro para servicios multimedia y clase plata para servicios best effort.

Tabla 3.21 SLA Retardo (round trip) servicio Macrolan

Retardo Tránsito	SLA escenario nacional
Clase Plata	45 ms.
Clase Oro	31 ms.
Clase Multimedia	27 ms.

3.7.1.2 Jitter

El jitter caracteriza la variación del retardo de una red. En la práctica, el jitter suele ser medido como la variación del retardo en un momento con respecto a una referencia, que suele ser el retardo mínimo o el medio (ver Figura 3.17).

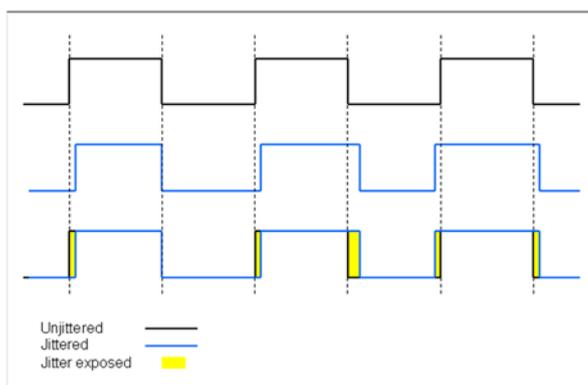


Figura 3.17 Jitter de una señal periódica

En el caso del servicio Macrolan y a modo de ejemplo en la tabla se incluyen los valores de SLO (Service Level Objective que no aplica penalización por incumplimiento) para el jitter sobre un acceso, tomando como medida los valores recogidos a lo largo de un día.

Tabla 3.22 Jitter máximo (SLO) para el servicio Macrolan

Jitter en red nacional	SLO
Clase Multimedia	<2 mseg

3.7.1.3 Pérdida de paquetes

Las pérdidas caracterizan los paquetes que son descartados entre 2 puntos de una red. Un paquete se considera perdido si no llega a su destino en un tiempo determinado (time out). Estas pérdidas de paquetes pueden ocurrir por distintos motivos:

- **Congestión.** Cuando hay congestión, las colas crecen y los paquetes acaban siendo descartados. Esto se puede evitar gestionando la carga de tráfico y aplicando unos mecanismos de gestión de colas y planificación adecuadas.
- **Errores en capas inferiores.** Si se producen errores de bits, que pueden ser causados por ruido o por una excesiva atenuación, es posible que el paquete se descarte. La mayoría de los protocolos usa mecanismos de detección de errores, como el CRC, o un checksum, y si detectan un error, descartan el paquete. La tasa de error suele ser pequeña.

- **Fallos en elementos de red.** Si algún elemento de la red falla, es posible que se pierdan paquetes hasta que se recupere el funcionamiento normal del elemento que ha fallado.
- **Pérdidas en aplicaciones.** Es posible que cuando el buffer está lleno, la aplicación descarte paquetes. Esto suele ser común en aplicaciones de tiempo real como VoIP y video streaming.

Generalmente, se pueden evitar pérdidas mediante un control de errores extremo a extremo (TCP) con reenvío de paquetes, pero en aplicaciones de tiempo real TCP no es viable. Los umbrales máximos admisibles para diferentes servicios en tiempo real [109] se describen a continuación, a partir de estos valores el servicio se convierte en indisponible.

- < 3%. Para transmisión de imágenes y de señales biomédicas.
- < 15%. Para audio y vídeo interactivo, conferencia multimedia de audio y vídeo, etc.
- < 20%. En general, para cualquier aplicación en tiempo real.

En el caso del servicio Macrolan, el SLA de pérdida de paquetes sobre un acceso, tomando como medida la media aritmética de los valores recogidos a lo largo de un día se detalla en la Tabla 3.23.

Tabla 3.23 SLA pérdida de paquetes servicio Macrolan

Pérdida de Paquetes	SLA escenario nacional
Clase Plata	0.9%
Clase Oro	0.8%
Clase Multimedia	0.7%

3.7.2 Ancho de banda disponible

El ancho de banda de una red se define como la velocidad a la que se transmite cada bit de información, es decir, el número bits que se transmiten en un determinado tiempo. El ancho de banda depende de la aplicación, de la tecnología de red usada, del códec y de los valores de retardo y pérdidas.

3.7.3 Requisitos de calidad por tipo de servicio

El impacto de los diferentes parámetros de QoS sobre las fuentes de tráfico dependerá fundamentalmente de su naturaleza y ancho de banda disponible.

3.7.3.1 VoIP

3.7.3.1.1 Impacto del Retardo

Para la VoIP, es importante el retardo en cada dirección. Si el retardo es demasiado grande, el oyente puede tener dificultades a la hora de distinguir entre pausas naturales en un discurso y retardos producidos por el sistema, y si es extremo, la conversación puede ser imposible.

La relación entre el retardo y la calidad de la llamada, medida en la escala Mos (*Mean Opinion Score*), que es una estimación con valores que oscilan entre 1 (mínima calidad) y 5 (máxima calidad) se muestra en la Tabla 3.24.

Tabla 3.24 Relación entre retardo y calidad (MOS)

Retardo	MOS
0ms - 150ms	5
150ms - 250ms	4
250ms - 325ms	3
325ms - 425ms	2
>425ms	1

3.7.3.1.2 Impacto del Jitter

Las aplicaciones susceptibles al jitter, como VoIP, usan unos buffers conocidos como “jitter buffers” para compensar el jitter a la llegada de los paquetes. Esto es importante porque la decodificación de la señal que llega es un proceso síncrono, por lo que los paquetes deben llegar al buffer a un ritmo constante. Los jitter buffers eliminan la variación del retardo convirtiendo ese retardo variable en un retardo fijo en el destino.

Si el jitter buffer provoca un retardo demasiado bajo para eliminar el jitter, es posible que el flujo de paquetes que le llegue al decodificador sea demasiado bajo y que en algún momento el buffer esté vacío cuando necesite un paquete nuevo, lo que equivale a la pérdida de un paquete.

La mayoría de los sistemas de VoIP usan jitter buffers adaptativos, que intentan solucionar esos problemas cambiando el valor del retardo fijo generado al valor más pequeño posible sin tener problemas. Estos buffers se ajustan dinámicamente al máximo valor de jitter de la red, por lo que no se genera un retardo mayor que en el peor de los casos.

3.7.3.1.3 Impacto de las Pérdidas

Cuando se pierde un paquete de voz, existen técnicas para minimizar o para eliminar el efecto de esa pérdida, como repetir el paquete anterior. Sin embargo, cuando se pierden varios paquetes consecutivos, la calidad de la llamada puede bajar drásticamente, resultando en una notable degradación del servicio.

3.7.3.1.4 *Impacto del Ancho de Banda*

Los códecs de VoIP producen un flujo constante de paquetes, por lo que el ancho de banda consumido por una fuente de VoIP será constante durante los periodos de ON. Las redes con tráfico VoIP deberían ser diseñadas para que tengan unas pérdidas de este tipo de paquetes cercanas a 0, intentando evitar congestiones en la cola de voz, que deben ser a su vez pequeñas para evitar que el retardo sea demasiado grande. Este problema ha generado reglas de diseño en las que el ancho de banda reservado para el tráfico VoIP debería ser como mínimo el máximo ancho de banda generado por las fuentes. Aun así, se pueden definir más fuentes de voz de las teóricamente aseguradas si se tienen en cuenta los periodos de inactividad de transmisor/receptor durante la llamada.

3.7.3.2 *Video*

3.7.3.2.1 *Impacto del Retardo*

El retardo en el tráfico de vídeo estará directamente relacionado con el servicio subyacente, es decir, en servicios de streaming o difusión, el impacto del retardo es irrelevante frente a su variación, no así en servicios interactivos como, por ejemplo, los servicios de videoconferencia donde aplican condicionantes similares a los definidos para el tráfico de voz.

3.7.3.2.2 *Impacto del Jitter*

Los decodificadores de vídeo usados para el streaming de vídeo necesitan recibir un flujo constante de paquetes, típicamente con tolerancias de jitter de unos 500 ms. Esa tolerancia se suele conseguir al igual que con VoIP con jitter buffers.

3.7.3.2.3 *Impacto de las Pérdidas*

El impacto de la pérdida de cuadros en un vídeo depende de la estructura del tipo de cuadro que se pierda. La pérdida de un Cuadro I es muy notable y afecta a todos los cuadros hasta que llega el siguiente cuadro I. La pérdida de un Cuadro P puede afectar a varios cuadros, mientras que la pérdida de un Cuadro B solamente afecta a ese cuadro.

Una estructura donde los GOP sean más grandes permitirá una mayor compresión, pero tendrá menos cuadros I, lo que hace que las pérdidas de paquetes alarguen el tiempo de sincronización del códec de vídeo.

3.7.3.3 *Datos y dispositivos MTC*

3.7.3.3.1 *Impacto del Retardo*

En este caso, la sensibilidad al retardo del tráfico MTC dependerá de los requerimientos específicos de la aplicación subyacente, del número de fuentes y de la ventana de sincronización con los servidores.

Como ejemplo para retardos de menos de una décima de segundo, el usuario tiene la sensación de que el sistema responde al instante. Si el retardo es del orden del segundo, el usuario notará el retardo, pero en general notará que la interacción usuario-máquina

es fluida. Sin embargo, retardos superiores afectan a la usabilidad de la aplicación, por ejemplo, a partir de los 8 segundos de espera para cargarse una página web, la mitad de los usuarios abandona [110].

3.7.3.3.2 Impacto del Jitter

El jitter no tiene mucho impacto en este tipo de tráfico.

3.7.3.3.3 Impacto de las Pérdidas

La pérdida de paquetes no acarrea efectos tan severos como en los servicios de voz y vídeo siempre que se mantenga en niveles adecuados para que la capa TCP pueda mantener las sesiones activas.

3.7.3.4 DiffServ

Para proporcionar la calidad de servicio, se crearon dos tipos de tecnologías en capa 3: IntServ y DiffServ. IntServ está basado en la reserva de recursos, de manera que cada router del trayecto debe tomar nota y efectuar una reserva de recursos mediante el protocolo RSVP. El principal inconveniente de esta solución radica en la necesidad de mantener información sobre cada flujo en todos los routers de la red, lo cual es inviable en escenarios grandes, generando problemas de escalabilidad.

DiffServ intenta evitar los problemas de escalabilidad que suponen el protocolo RSVP de IntServ. Para ello, se basa en el marcado de paquetes IP, sin reservar recursos, sin usar un protocolo de señalización y sin información de estado en los routers. De esta manera, en lugar de distinguir flujos individuales, DiffServ clasifica los paquetes en categorías, y a cada de ellas le corresponde un SLA (Acuerdo de Nivel de Servicio), el cual se contrata con el operador.

Este marcado se produce en el campo DSCP (RFC 2474), de manera que se pueden establecer una serie de clases y subclases en los que se clasifican los paquetes IP dependiendo del valor marcado.

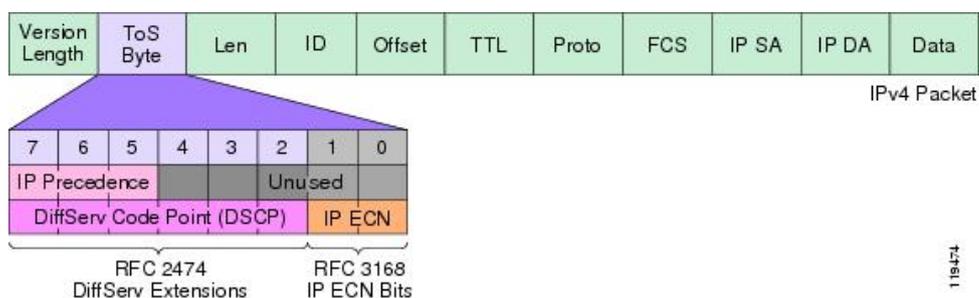


Figura 3.18 Estructura del campo DSCP dentro de la trama de IPv4

Estas clases son las siguientes:

- **Expedited Forwarding (EF).** Es la clase más prioritaria, y se utiliza para marcar el tráfico que requiera un retardo muy bajo, pocas pérdidas, y un ancho de banda asegurado.
- **Assured Forwarding (AF_{xy}).** Es la segunda clase más prioritaria, y tiene a su vez 4 subclases según la prioridad, y además a cada subclase se le puede asignar una prioridad de descarte de 1 a 3, siendo 1 la mínima posibilidad de descarte. El subíndice x representa la subclase y el subíndice y representa la probabilidad de descarte.
- **Class Selector (CS).** Originalmente, para establecer niveles de prioridad entre paquetes IP se pensó en definir un campo de 1 byte, el ToS (Type of Service), que usaba 3 bits para definir la prioridad (IP Precedence). Esta clase que se definió para ser compatible con los valores de prioridad de IP Precedence.
- **Default Forwarding (DF).** Es el tráfico best effort, y el menos prioritario.

Para la construcción de la herramienta SimulCity se ha usado como referencia la clasificación del tráfico del servicio Macrolan. Macrolan divide el tráfico en Multimedia, Oro, Plata y Gestión, aunque este último no se comercializa al usuario. La equivalencia entre DSCP y los tipos de tráfico en Macrolan se puede encontrar en la siguiente tabla:

Tabla 3.25 Correspondencia entre DSCP y tráfico equivalente en Macrolan

Servicio	Nombre DSCP	Valor DSCP	C. Macrolan
Voz	EF, CS5	46, 40	Multimedia
Vídeo	AF41, AF42, AF43, CS4	40, 28, 36, 32	Oro
Gestión / tráfico crítico	AF31, AF32, AF33, CS3	30, 28, 26, 24	Gestión
Tiempo real	AF21, AF22, AF23, CS2	22, 20, 18, 16	Oro
Alto requerimiento de ancho de banda	AF21, AF22, AF23, CS1	14, 10, 12, 8	Plata
Estándar	DF, CS0	0	Plata

El funcionamiento de Diffserv, que ha sido implementado en SimulCity asigna el DSCP según alguna característica del paquete, como puede ser la IP de origen o de destino, o el puerto de origen o de destino, o por el patrón de tráfico que general.

La función policía solamente actúa en los routers de entrada de la red de la nube MPLS del operador (que conectará los Centros Corporativos con la cloud privada donde se encuentra la Plataforma de Ciudad), y en el router de entrada a la cloud privada. El router de ingreso al dominio DiffServ (es decir, el primer router de todos), es el que se encarga de marcar el campo DSCP, los demás solamente se encargan de realizar el tratamiento que corresponde según el DSCP.

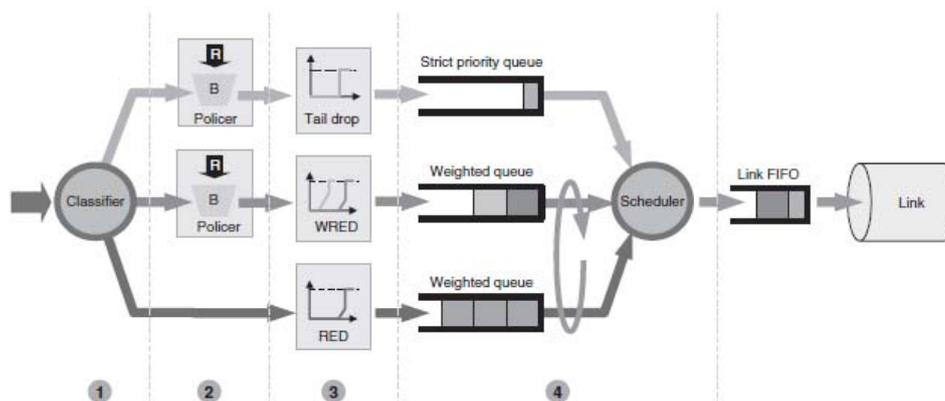


Figura 3.19 Implementación de la calidad de servicio en un router

En la Figura 3.19 se muestran los distintos mecanismos usados: la clasificación y marcado de las tramas (1), las funciones policía (2), y la gestión de colas (3) junto con el planificador que decide a qué cola se le tiene que dar servicio (4).

3.7.3.4.1 Clasificación

La clasificación es el proceso mediante el cual se identifican los distintos tipos de tráfico y se dividen en clases de servicio que, en el caso de este modelo, al estar basado en Macrolan, serán 4: oro, plata, multimedia y gestión.

Este proceso es necesario para poder tratar a cada clase de servicio de una manera distinta, pudiendo ofrecer a cada clase un caudal garantizado y un caudal máximo permitido, y por lo tanto poder establecer distintas prioridades.

Existen diversos criterios para clasificar el tráfico, que pueden ser la dirección de origen o de destino, el puerto de origen o de destino, etc. En esta herramienta, se ha elegido el puerto de destino como el elemento que se ha usado para clasificar el tráfico.

3.7.3.4.2 Marcado

El marcado de los paquetes permite que todos los nodos de la red sean más eficientes a la hora de cumplir los requisitos de calidad contratados con el operador, pues hace más sencillo el clasificado de tráfico, y permite establecer prioridades entre el tráfico, por ejemplo, asignando una mayor prioridad de descarte en caso de congestión en las colas al tráfico que no esté garantizado. El marcado de los paquetes se realiza en el campo del DSCP.

3.7.3.4.3 Función policía (Policing)

Las funciones policía miden los parámetros temporales de tráfico, y su principal objetivo es evitar que el tráfico cumpla con los parámetros contratados con el operador. Si se

cumplen las especificaciones de tráfico acordadas con el operador, el tráfico se transmite sin ningún problema. Sin embargo, en caso de no cumplirse, el paquete se marca con una mayor probabilidad de descarte baja su prioridad, o directamente se descarta.

Para conocer el perfil temporal del tráfico se han utilizado algoritmos basados en un leaky bucket o en un token bucket, concretamente el algoritmo "two rate Three Color Marker" [113], compuesto por dos tokens bucket. Para ello se ha hecho uso de cuatro parámetros:

- **CIR** (*Committed Information Rate*). Es la tasa de tráfico garantizado. (bps)
- **CBS** (*Committed Burst Size*). Es el máximo tamaño de ráfaga garantizado. (B)
- **PIR** (*Peak Information Rate*). Es la tasa máxima de tráfico. Debe ser mayor o igual que el CIR. (bps)
- **PBS** (*Peak Burst Size*). Es el máximo tamaño de ráfaga permitido. Debe ser mayor o igual que el CBS. (B)

3.7.3.4.4 Gestión de colas

Por último, la gestión de las colas es uno de los elementos más importantes a la hora de garantizar la calidad de los distintos servicios, pues va a afectar de una manera importante a los parámetros de calidad como el retardo o la pérdida de paquetes en casos de congestión.

La congestión se produce cuando los paquetes llegan a la salida del router a una tasa mayor que la del enlace, por lo que se llenan las colas de paquetes esperando salir, produciéndose retardos y pérdidas de paquetes. Para gestionar las colas, se implementa también un planificador, que es el encargado de decidir a cuál de las colas se sirve primero.

3.7.3.4.5 Cola: Drop Tail Queue

Es el mecanismo más simple de gestión de memoria en una cola, en el que sencillamente se descartan paquetes cuando supera un cierto umbral.

Las principales ventajas incluyen la facilidad de implementación, y que es sencillo garantizar un bajo retardo haciendo la cola pequeña, o minimizar la pérdida de paquetes aumentando el tamaño de los paquetes.

Sin embargo, es un algoritmo muy poco flexible, pues no empieza a descartar paquetes hasta que la cola esté llena y se hayan consumido todos sus recursos. Además, eso provoca que en caso de congestión hay que elegir entre tener poco retardo y pocas pérdidas, pues una cola de tamaño pequeño empezará a descartar paquetes muy pronto, mientras que, si tiene un tamaño grande, el retardo será también mayor.

3.7.3.4.6 Cola: RED (*Random Early Detection*)

El funcionamiento del algoritmo RED (RFC 2309) está basado en dos umbrales y una posibilidad de descarte, de manera que se monitoriza en todo momento el número de paquetes que hay en la cola.

Si el número de paquetes en la cola no supera el umbral mínimo, no se hará nada; y si se supera el umbral máximo, se descartarán todos los paquetes que lleguen hasta que baje el número de paquetes esperando en la cola. Si, por el contrario, el número de paquetes supera el primer umbral, pero no supera el segundo, se empezará a descartar paquetes con la posibilidad de descarte definida.

Existe una variante conocida como WRED (Weighted Random Early Detection) que permite asignar diferentes perfiles de descarte RED a distintos tipos de tráfico, lo que usaremos en nuestro modelo para distinguir entre tráfico conforme y tráfico con mayor posibilidad de descarte.

De este modo, se conseguirá una mayor flexibilidad en las colas frente a congestiones, a costa de una mayor complejidad de implementación y de configuración que en el caso de las colas Drop Tail.

3.7.3.4.7 Planificador: Priority Queueing (PQ)

Un planificador de colas prioritarias tiene tantos puertos de entrada como colas deba planificar, y un puerto de salida. De modo que las colas están ordenadas por prioridad, de modo que las colas menos prioritarias solamente se servirán cuando las colas más prioritarias estén vacías.

Este planificador es fácil de implementar y permite a los routers servir una clase de tráfico de modo diferente a otras, además de aumentar la estabilidad de la red durante periodos de congestión, y garantiza que la cola menos prioritaria tenga bajas pérdidas y pocas pérdidas. Sin embargo, cuando hay muchas colas poco prioritarias, éstas podrían experimentar un retardo excesivo, o incluso podrían provocar descartes de paquetes.

3.7.3.4.8 Planificador WRR (Weighted Round Robin):

En un planificador con WRR está diseñado para asegurar que cada flujo tenga un acceso definido a los recursos de la red, asignando unos pesos a cada cola. Estas colas se sirven siguiendo un orden de turnos round-robin, es decir, se servirán por turnos, y cada una lo que le toca según los pesos asignados, saltándose las colas vacías.

De esta manera se evitan los problemas del PQ con las colas menos prioritarias, y si un flujo intenta consumir más de su ancho de banda, solamente afectará a su cola, y no influirá en las demás colas.

Sin embargo, la distribución hecha por un planificador WRR no siempre es "justa", pues si una cola tiene un peso muy alto, las demás colas pueden pasar demasiado tiempo sin ser servidas. Por ello, otros algoritmos como el WFQ (Weighted Fair Queue) o SRR (Shared Round Robin) suelen ser usados, aunque su modelado es más complicado.

3.7.3.4.9 Solución escogida

En el modelo implementado, se ha decidido implementar un sistema de colas como el usado por el Switch-Router 3750 de Cisco [44], que implemente 4 colas.

Para el tráfico de voz, se requiere una cola con pocas pérdidas y poco retardo, por lo que se elegirá una cola prioritaria del tipo Drop Tail de pequeño tamaño.

Para el resto de las colas se implementará un mecanismo RED, de manera que en caso de congestión se empiecen a descartar los paquetes marcados con mayor probabilidad de descarte antes que los paquetes conformes.

Además, para evitar que el tráfico menos prioritario se quede sin servicio, se ha implementado un WRR con pesos ajustables por el usuario que sirva a las colas de gestión, oro y plata.

De esta manera, la implementación de las colas se ha diseñado de modo que hay una cola prioritaria para la clase multimedia y una cola menos prioritaria, que engloba al tráfico de oro, plata y gestión.

3.8 Conclusiones

En este capítulo se ha abordado la problemática de modelar diferentes fuentes de tráfico en el ámbito de una red convergente municipal donde existe un número prácticamente ilimitado de nuevos servicios con patrones de tráfico y requerimientos de calidad de servicio diferentes (retardo, Jitter y pérdida de paquetes) que es necesario garantizar.

En primer lugar, se ha definido una arquitectura de red simplificada donde se consideran exclusivamente funciones del plano de transporte y de la capa de aplicación, asumiendo que las funciones de la capa de control están prefijadas de antemano, como puede ser las clases de servicio y el control de las sesiones. Este modelo se ha trasladado a un entorno de comunicaciones municipal donde se han descrito diferentes mecanismos de acceso existentes y la interconexión con otras redes.

En segundo lugar, se han descrito los diferentes tipos de tráfico en las redes de telecomunicaciones y sus propiedades ya que éstas inciden directamente sobre el rendimiento y por lo tanto en la calidad de servicio que ésta ofrece. Las propiedades del tráfico se han definido de manera sucinta, describiéndose brevemente el tráfico de Poisson y una introducción al tráfico autosimilar. También se han descrito las diferentes aproximaciones para modelar el tráfico: modelos matemáticos, modelos de simulación y modelos físicos.

En tercer lugar, se han analizado las características de las fuentes de tráfico asociadas a las comunicaciones HTC. Se han identificado diferentes modelos en base a la monitorización pasiva de las fuentes de tráfico y la definición de modelos estadísticos que describen su comportamiento para fuentes de voz y vídeo. En base a las características identificadas, se han propuesto modelos parametrizables de fuentes de voz y vídeo que se podrán utilizar como generadores de fuentes de tráfico en herramientas de simulación.

En cuarto lugar, se han analizado las características de las fuentes de tráfico asociadas a las comunicaciones MTC, con especial atención al modelado de fuentes agregadas propuesto por 3GPP. En base a las características identificadas se han definido un modelo

de fuente MTC parametrizable que se podrá utilizarse como generador de fuentes de tráfico en herramientas de simulación.

Por último, se ha realizado un análisis sobre diferentes aspectos relativos a la calidad de servicio en la conexión de la red municipal con la Plataforma de Ciudad describiendo las características de un servicio de VPN llamado Macrolan del operador Telefónica. Se han definido los tipos de colas y se ha escogido la configuración más adecuada a las necesidades de la gestión de cada clase de servicio, colas pequeñas Drop Tail para tráfico multimedia, para el resto se implementará un mecanismo RED y para evitar que el tráfico menos prioritario se quede sin servicio, se ha implementado un WRR con pesos ajustables por el usuario que sirva a las colas de gestión, oro y plata.

Capítulo 4

Herramienta de Simulación SimulCity

4.1 Introducción

Este capítulo describe como contribución a esta Tesis, el diseño de la herramienta de Simulación denominada “SimulCity” que permite, como funcionalidades destacadas, evaluar el comportamiento de un acceso Metro Ethernet con fuentes de tráfico heterogéneas que compiten por un ancho de banda definido y contratado al operador.

Como se ha comentado con anterioridad, la existencia de una única infraestructura de acceso requiere de mecanismos de gestión de tráfico para mantener los objetivos de calidad de acuerdo con las recomendaciones internacionales.

En la fase de especificación funcional de SimulCity se consideró prioritario:

- Flexibilidad en el diseño de las fuentes y definición de escenarios.
- Facilidad de uso y configuración.
- Entorno gráfico.
- Uso de software de libre distribución en entornos académicos.
- Exportación de resultados a otras herramientas de análisis.

Aunque existen diferentes herramientas de simulación en el mercado como OPNET, GNS3, SSFNET, CNET, etc. SimulCity se ha diseñado utilizando el entorno de simulación OMNeT++ debido a la experiencia previa existente, su gratuidad para entornos académicos, un elevado número de módulos ya desarrollados, la programación en C++ y su flexibilidad.

Este capítulo se estructura en dos bloques, el primero describe muy sucintamente el entorno de simulación basado en OMNeT++. El segundo bloque presenta una aportación relevante de esta Tesis que es una herramienta denominada SimulCity. SimulCity está diseñada para evaluar el impacto de la introducción masiva de dispositivos en la infraestructura de comunicaciones municipal, especialmente en el dimensionado y evaluación de calidad de servicio de los accesos a la Plataforma de Ciudad alojada en un Centro de Datos externo.

4.2 OMNeT++

4.2.1 Visión general de OMNeT++

OMNeT++ [115] es un entorno de simulación de uso libre para instituciones universitarias desarrollado en C++. Su principal área de aplicación es la simulación de redes de telecomunicación, sistemas complejos de IT, redes de colas o arquitecturas hardware.

Como características principales, OMNeT++ trabaja en base a módulos y mensajes. Los modelos consisten en módulos jerárquicamente anidados que se comunican mediante mensajes. Los módulos activos se denominan módulos simples y su comportamiento se define mediante ficheros escritos en C++. Los módulos simples pueden ser agrupados dentro de otros módulos (módulos compuestos). La profundidad de anidación de módulos no está limitada. El modelo final, o "network" (red), es en sí un módulo compuesto como se muestra en la Figura 4.1. Los módulos son reutilizables (tanto los simples como los compuestos).

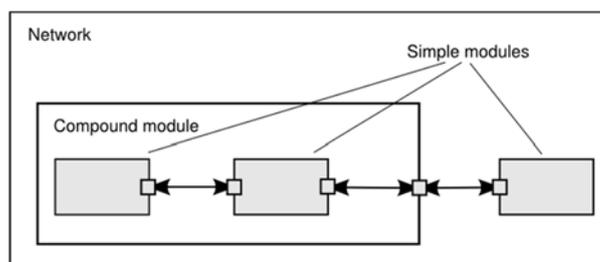


Figura 4.1 Jerarquía de módulos en OMNeT++ (<https://www.OMNeTpp.org/>)

Este modelo de jerarquía se define en archivos específicos NED (.ned), que usan un lenguaje especial (lenguaje NED).

Los módulos se conectan entre ellos intercambiando mensajes, y pueden ser enviados directamente al destino, o llegar a través de un camino predefinido, atravesando puertas y conexiones. Los mensajes pueden llegar de otros módulos, pero también del propio módulo (por ejemplo, para implementar contadores).

Las puertas (gates) son las interfaces de entrada y de salida de los módulos, y se conectan entre sí a través de enlaces (conexiones). Los mensajes se envían a través de las puertas

de salida, y se reciben por las puertas de entrada. Existe la posibilidad de definir puertas de entrada/salida y conexiones bidireccionales.

Para facilitar el modelado de redes de comunicaciones reales, las conexiones pueden modelar conexiones físicas, con parámetros como la velocidad de los datos, el retardo que provoca, y la tasa de error.

OMNeT++ también permite el uso de parámetros, que pueden ser definidos en los propios ficheros NED, o en el fichero de configuración OMNeTpp.ini, y éstos pueden ser de varios tipos: string (cadena), numérico, booleano, `simtime_t` (tiempo de simulación), o un fichero xml. Con estos parámetros, el usuario puede fácilmente cambiar la configuración y el comportamiento de un módulo.

Existe un entorno gráfico de desarrollo basado en el framework Eclipse [116] para facilitar las tareas de diseño, modelado, ejecución de la simulación y análisis de resultados.

4.2.1.1 Uso de OMNeT++

OMNeT++ dispone de un conjunto de bibliotecas con componentes predefinidos que pueden ser parametrizados. También los componentes pueden ser utilizados como plantillas para la creación de nuevos elementos personalizados.

- **Biblioteca propia de C++.** Incluye un kernel de simulación y varias clases útiles como generación de números aleatorios, colección de estadísticas, topologías..., usadas para crear la simulación de los componentes (módulos simples y canales) y de los mensajes.
- **Infraestructura para las simulaciones.** Un lenguaje propio para describir la arquitectura de la red (NED), y también ficheros para configurar la simulación (INI).
- **Interfaz gráfica para la simulación.** Con Tkenv. Permite seguir la evolución del modelo a medida que pasa el tiempo.
- **IDE basado en Eclipse.** Para diseñar, ejecutar y evaluar simulaciones de forma sencilla, ofreciendo normalmente 2 modos: uno gráfico y otro de texto (código).
- **Extensiones.** Existe un elevado número de bibliotecas de elementos preconfigurados y reutilizables. Destaca el proyecto INET [117] que contienen elementos ya programados y que son reutilizables.

El proceso de simulación consiste en construir un modelo, añadir las bibliotecas de componentes para construir el proyecto (Build Project), posteriormente editar el fichero de configuración (OMNeTpp.ini) y ejecutarlo. El interfaz gráfico de simulación Tkenv permite la ejecución paso a paso, continua o con puntos de ruptura.

Los módulos simples son en realidad clases de C++, que heredan de la clase **cSimpleModule**, y que redefinen una serie de funciones. Estas clases se registran en OMNeT++ mediante **Define Module()**.

Los módulos se comunican mediante mensajes, que se definen con la clase **cMessage**. Estos mensajes son procesados en la función *handleMessage(cMessage *msg)*, y se envían a otros módulos con *send(cMessage *msg, const char *outGateName)*. Los auto-mensajes (es decir, mensajes programados por el propio módulo para implementar por ejemplo timeouts), se programan con *scheduleAt(simtime_t time, cMessage *msg)*, y se pueden cancelar con *cancelEvent(cMessage *msg)*.

cMessage incluye variables internas como *name*, *length*, *kind*,... y también incluye algunas funciones útiles como *encapsulate(cMessage *msg)*, y *decapsulate()*.

OMNeT++ nos ofrece una manera sencilla para programar mensajes personalizados mediante los **ficheros MSG** (.msg). Estos ficheros soportan herencia, estructuras de datos, arrays...

4.2.1.2 Uso del IDE

El entorno de desarrollo integrado (Integrated Development Environment, IDE) de OMNeT++ está basado en la plataforma Eclipse. OMNeT++ añade funcionalidad para crear y configurar modelos, realizar ejecuciones por lotes y analizar los resultados de las simulaciones.

La ventana principal del IDE se divide en varias zonas: el "*Workspace*" es la zona donde se muestra todos los proyectos disponibles en el directorio de trabajo. Desde dicha zona se pueden crear o importar nuevos proyectos. Se presenta en formato de jerarquía de carpetas y se pueden explorar todos los archivos que componen un determinado proyecto. En la Figura 4.2 se puede observar el entorno de desarrollo integrado.

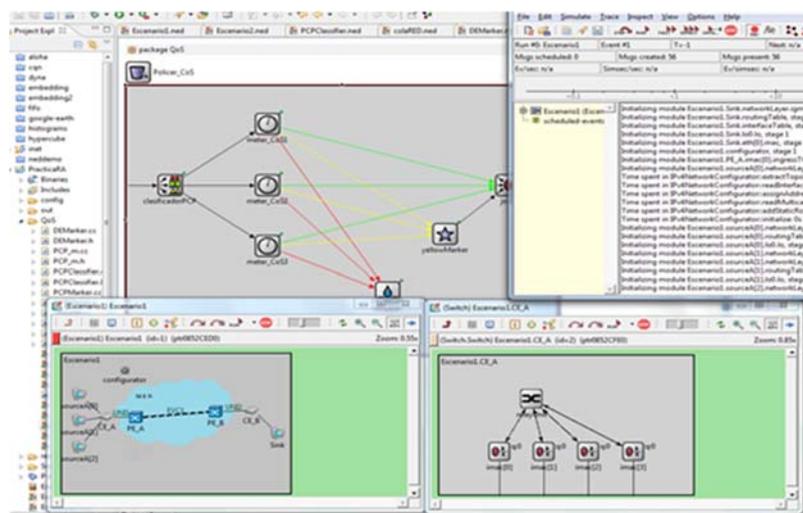


Figura 4.2 El IDE de OMNeT++

El editor es la otra zona importante de la ventana principal de IDE. En esta zona se puede ver y editar el contenido de los diferentes archivos del proyecto. Cuando un usuario hace doble click en cualquier archivo del proyecto, automáticamente se abre un editor asociado al tipo de archivo que se haya seleccionado. Normalmente presenta dos opciones para visualizar el contenido de un archivo, en modo gráfico y en modo código fuente. Estas opciones se seleccionan mediante las pestañas que aparecen en la parte inferior del editor.

Existen diferentes tipos de archivos que un proyecto puede contener:

- **Archivos ".h"**: archivos cabecera de C++ donde se definen diferentes funciones y variables.
- **Archivos ".cc o .cpp"**: archivos de código C++. Definen el comportamiento de los módulos simples haciendo uso de las funciones y variables declaradas en los archivos ".h".
- **Archivos ".msg"**: definiciones de mensajes. Permiten definir los mensajes que se van a usar en la simulación.
- **Archivos ".ned"**: archivos donde se define la topología de módulos y redes.
- **Archivos ".ini"**: archivos de configuración de parámetros para la simulación.
- **Archivos ".anf"**: archivos de análisis de resultados. Reconocen, tratan y representan los resultados generados por una simulación.
- **Archivos ".xml"**: OMNeT++ permite la asignación de parámetros en los módulos mediante este tipo de archivos. Son útiles cuando se necesitan estructuras de datos complejas para asignar un valor a un parámetro (por ejemplo, el valor de una dirección MAC en una interfaz Ethernet).

Para simular un proyecto con OMNeT++ siempre hay que seguir los mismos pasos:

- Definir la arquitectura que tendrá la red en el lenguaje NED (con una jerarquía de módulos y submódulos).
- Los módulos simples del modelo deben ser programados en C++ (o bien reusar módulos simples ya programados).
- Crear el fichero OMNeTpp.ini para establecer la configuración de la simulación, y para definir los parámetros. Un fichero de configuración puede describir distintas simulaciones con distintos parámetros.
- Simular el modelo.
- Analizar los resultados de la simulación.

4.3 SimulCity

Como ya se apuntó en la introducción, SimulCity responde a una necesidad operativa para la evaluación de prestaciones y dimensionado de la red de acceso, basada en los estándares Metro-Ethernet, ante la implantación de proyectos que incorporan un elevado número de dispositivos que inyectan tráfico a la Plataforma de Ciudad.

En la Figura 4.3 se describe de manera esquemática, no exhaustiva, la arquitectura de comunicaciones típica de una corporación municipal (ya descrita en el Capítulo 3) donde la mayor parte de edificios están conectados con una infraestructura propia basada en diferentes tecnologías, principalmente fibra óptica.

Por razones económicas, edificios pequeños, empresas municipales o participadas se conectan con la corporación a través de redes privadas virtuales contratadas con un operador utilizando tecnologías FTTH, xDSL, etc.

La conectividad con dispositivos IoT se realiza a través de internet o de soluciones M2M de operador.

La corporación municipal dispone de accesos internet para la relación con los ciudadanos, acceso remoto de miembros de la corporación y otros dispositivos o sistemas externos.

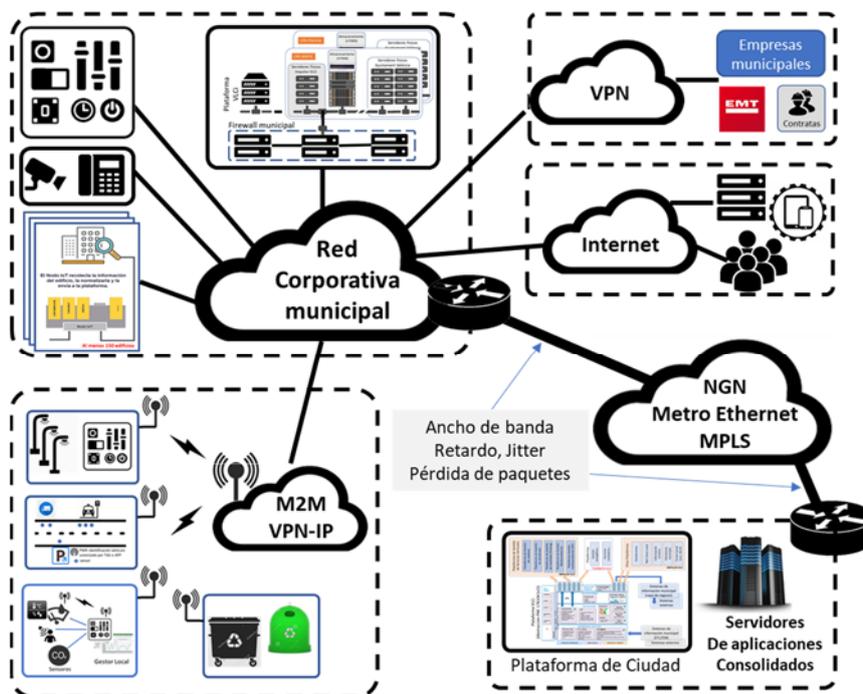


Figura 4.3 Arquitectura simplificada de las comunicaciones de una corporación municipal.

Por último, la infraestructura tecnológica de la Plataforma de Ciudad reside en servidores externos a la Corporación Municipal conectados por un acceso Metro-Ethernet.

Como funcionalidades más destacadas, SimulCity permite:

- Definir flujos de tráfico arbitrarios (fuentes) hacia la Plataforma de Ciudad tanto en número como en tipo y comportamiento.
- Asignar cada tipo de fuente a una calidad de servicio determinada.
- Configurar los parámetros contratables al operador de los enlaces Metro Ethernet, permitiendo evaluar el incremento de costes en comunicaciones derivado de la introducción de nuevos proyectos municipales.
- Configurar los mecanismos de gestión de colas y descarte de tráfico.
- Configurar el comportamiento de la nube MPLS de acuerdo con diferentes criterios de retardo y pérdida de paquetes.
- Construir un escenario de red y simular su funcionamiento durante un tiempo arbitrario con diferentes condiciones de contorno.
- Obtener resultados de ancho de banda utilizado, retardos, Jitter, pérdida de paquetes para cada una de las fuentes y clase de servicio definidas.
- Los resultados pueden utilizarse, mediante un proceso iterativo para conseguir el dimensionado adecuado de los accesos y minimizar los costes.

A continuación, se describirán con mayor detalle los módulos principales con los que cuenta SimulCity:

- Configurador de fuentes de tráfico.
- Elemento de Red e interfaz UNI Metro Ethernet de acuerdo con la especificación del servicio Macrolan de Telefónica.
- Módulos diseñados utilizando la biblioteca INET.
- Módulos complementarios.
- Implementación del escenario
- Configurador de escenarios
- Módulo de estadísticas

4.3.1 Fuentes de Tráfico implementadas en SimulCity

4.3.1.1 Fuentes de VoIP

Se ha utilizado como base la aplicación básica de UDP adaptándola a un modelo de ráfagas On-Off, e implementando los atributos necesarios para que sea lo más realista posible, y además que sea totalmente configurable dependiendo de las necesidades del usuario. Para ello se ha utilizado como base el artículo de A.D. Shaikh et al. sobre el modelado de fuentes de VoIP [19].

El modelo se basa en dos estados generales que definen cuándo la fuente está generando tráfico y cuándo no (cuya duración es definida por el usuario mediante los parámetros de duración de llamada y de tiempo inactivo), y otros dos estados de On-Off que definen

los períodos de On y de Off cuando la fuente está activa, de tal manera que envía paquetes de voz cuando la fuente está activa y su estado es On.

La fuente funciona con auto mensajes, que indican a la fuente que tiene que empezar a funcionar (tipo START), que tiene que enviar un nuevo paquete de voz al destino (tipo SEND), que tiene que cambiar de estado (tipo STATE), y cuándo debe acabar la ejecución (tipo END).

Es necesario que antes de programar el envío del siguiente paquete de voz (justo al enviar uno) se compruebe que no haya ningún cambio de estado antes del instante previsto para el envío de ese mensaje. En caso de que haya algún cambio de estado, se enviará un auto mensaje en el instante del cambio de estado para avisar a la fuente de ese cambio de estado.

Las variables que controlarán el funcionamiento de la fuente serán las siguientes:

Tabla 4.1 Variables usadas en el modelado de una fuente de voz en OMNeT++

<i>Variable</i>	<i>Tipo</i>	<i>Función</i>
tMsg	cMessage	Automensaje de control
startTime	simtime t	Tiempo mínimo de inicio
stopTime	simtime t	Instante en el que se termina la ejecución
nextPkt	simtime t	Instante en el que se enviará el siguiente paquete
nextOnOff	simtime_t	Instante en el que se produce el siguiente cambio de estado OnOff
nextActive	simtime_t	Instante en el que se produce el que la fuente pasa a estar activa o inactiva
tOnOff	simtime t	Duración del estado actual de OnOff
tActive	simtime t	Tiempo durante el cual la fuente permanece activa o inactiva
Tam	double	Tamaño de cada paquete enviado (depende del códec)
wtime	double	Separación entre cada paquete que se envía
isSending	bool	Indica si la fuente está activa y el estado de la misma es On
isActive	bool	Indica si la fuente está activa
transit	bool	Indica si la fuente está en régimen transitorio

Si el valor de transit es false (régimen permanente) y la fuente está configurada para empezar en $t=0s$ ($startTime=0s$), se calculará la probabilidad de que la fuente ya estuviera activa, y una variable aleatoria decidirá si la fuente empieza en $t=0s$ (ya estaba activa) o en un tiempo aleatorio entre $0s$ y el tiempo máximo de inactividad de la fuente.

En un principio la fuente está inactiva, y se configura el primer automensaje, que será del tipo START. La lógica del programa con la llegada de cada mensaje está descrita en las siguientes figuras:

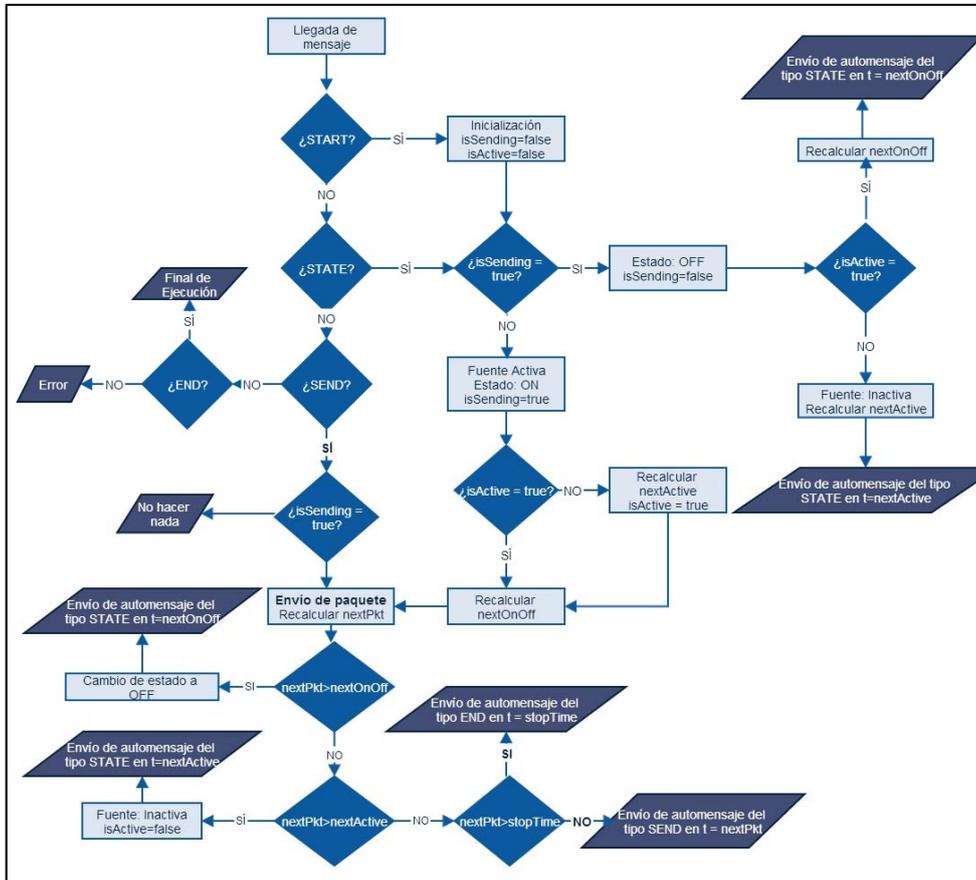


Figura 4.4 Diagrama de funcionamiento de una fuente de voz

4.3.1.2 Fuentes de Vídeo

Al igual que las fuentes de voz, para el desarrollo de las fuentes de vídeo se ha utilizado como base la aplicación básica de UDP, implementando atributos que hagan que la fuente se comporte de la manera más realista posible.

En la fuente de vídeo se han implementado dos estados: uno en el que se envían cuadros de vídeo, 25 por segundo por defecto siguiendo el estándar europeo, y otro en el que no se envía tráfico. Al igual que en el caso anterior, la fuente funcionará mediante automensajes, que pueden ser del tipo START, STATE, SEND y END.

En la realidad, los cuadros no siempre se distribuyen uniformemente en el tiempo, y el tamaño de estos cambia dependiendo del tipo de cuadro y de la cantidad de información que contenga, lo que implica que el bitrate generado sea variable.

El algoritmo de Kush [93] se ha utilizado para limitar el ancho de banda de vídeo, si se usa el códec H.264 [89], a partir de la resolución en píxeles de los cuadros ($ResX * ResY$), de la cantidad de cuadros por segundos (FPS) y de un factor que indica el movimiento del vídeo (Mov) según se describe en la Ecuación 3.17.

Se pueden considerar variaciones aceptables si están comprendidas entre el 75% y el 150% de este valor óptimo.

Por lo tanto, el programa generado calcula el bitrate mínimo y máximo, y a partir de esos valores, calculará el tamaño mínimo y máximo que puede tener cada cuadro para estar dentro de ese rango. Para conseguir el bitrate variable, el tamaño de cada cuadro de vídeo que enviaremos será aleatorio siguiendo una distribución uniforme entre el tamaño mínimo calculado y el máximo, y enviamos los cuadros cada $1/FPS$ segundos.

Debido a un bug de OMNeT++ relacionado con la fragmentación en la capa IP de paquetes grandes, se ha optado por forzar una fragmentación en la propia aplicación si el cuadro a enviar es demasiado grande (más de 1400B si se quiere fragmentar para no superar el tamaño máximo de un paquete IP, o más de 64KB para no superar el tamaño máximo de un paquete UDP. La primera opción es más segura, pero se genera una gran cantidad de fragmentos, lo que puede dificultar el análisis del escenario).

Las variables que controlarán el funcionamiento del programa serán las siguientes:

Tabla 4.2 Variables usadas en el modelado de una fuente de vídeo en OMNeT++

Variable	Tipo	Función
tMsg	cMessage	Automensaje de control
startTime	simtime t	Tiempo de inicio
stopTime	simtime t	Instante en el que se termina la ejecución
nextPkt	simtime t	Instante en el que se enviará el siguiente paquete
nextActive	simtime_t	Instante en el que se produce el que la fuente pasa a estar activa o inactiva
tActive	simtime t	Tiempo durante el cual la fuente permanece activa
TamMin	double	Tamaño mínimo del cuadro. Calculado automáticamente a partir de los parámetros.
TamMax	double	Tamaño máximo del cuadro. Calculado automáticamente a partir de los parámetros.
Tam	double	Tamaño del cuadro a enviar
wtime	double	Separación entre cada paquete que se envía
isActive	bool	Indica si la fuente está activa
isFirst	bool	Indica si se va a enviar un cuadro nuevo o un fragmento

En un principio la fuente está inactiva, y se configura el primer automensaje, que será del tipo START. La lógica del programa con la llegada de cada mensaje está descrita en la siguiente figura:

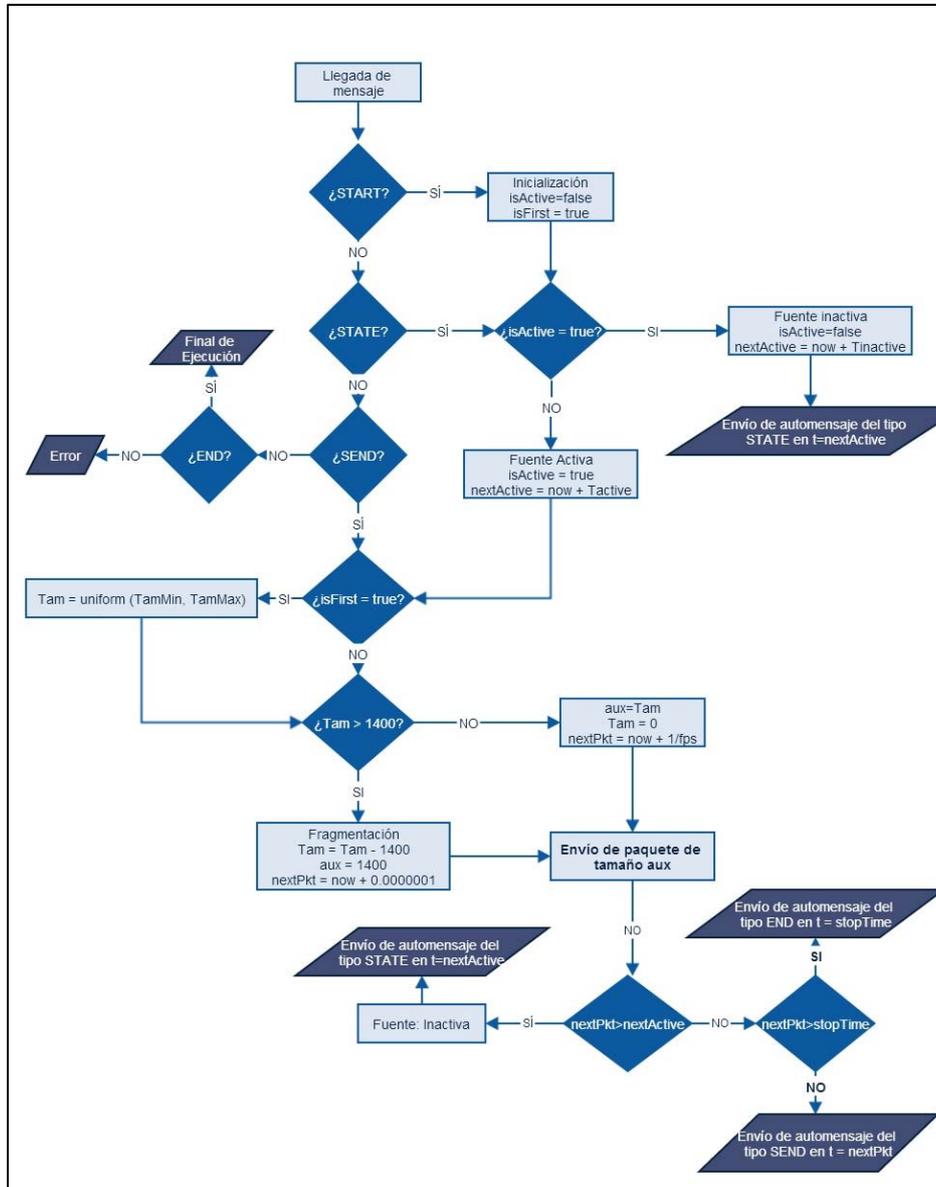


Figura 4.5 Diagrama de funcionamiento de una fuente de vídeo

4.3.1.3 Fuente de Datos

Como en el caso de las dos fuentes anteriores, se ha usado la aplicación de UDP básica como base, y además funciona con automensajes, que pueden ser del mismo tipo que en los dos casos anteriores: START, STATE, SEND y END.

La fuente de datos funciona como un generador de ráfagas en las que se envían ficheros que pueden contener tanto texto como imágenes. Cada ráfaga se considera una sesión, en las que se envían un número de ficheros configurable, de tamaño configurable.

Para la definición del modelo, y ante la heterogeneidad de estructuras y formatos de datos se ha considerado adecuado utilizar como referencia los requerimientos de Servicio para comunicaciones MTC del 3GPP TS 22.368 [15].

Por defecto, el tamaño del fichero a enviar se recalcula cada vez que se produce un envío, pero el usuario puede configurar la fuente para que el tamaño del fichero se mantenga constante durante toda la sesión, o que el tamaño del fichero sea siempre constante.

Al igual que en la implementación de la fuente de vídeo, debido a un bug de OMNeT++ con la fragmentación en la capa IP, se fuerza en la propia aplicación si el fichero tiene un tamaño mayor que 1400B (o de 64KB).

Las variables que controlarán el funcionamiento del programa serán las siguientes:

Tabla 4.3 Variables usadas en el modelado de una fuente de datos en OMNeT++

<i>Variable</i>	<i>Tipo</i>	<i>Función</i>
tMsg	cMessage	Automensaje de control
startTime	simtime t	Tiempo de inicio
stopTime	simtime t	Instante en el que se termina la ejecución
nextPkt	simtime t	Instante en el que se enviará el siguiente paquete
numPackets	int	Número de ficheros que faltan por enviar en la sesión actual
numBursts	int	Número de sesiones restantes.
Tam	double	Tamaño del fichero a enviar
WTime	double	Separación entre cada fichero que se envía dentro de una sesión
WBurstTime	double	Separación entre cada sesión
isActive	bool	Indica si la fuente está activa
isFirst	bool	Indica si se va a enviar un cuadro nuevo o un fragmento

En un principio la fuente está inactiva, y se configura el primer automensaje, que será del tipo START. La lógica del programa con la llegada de cada mensaje está descrita en la siguiente figura:

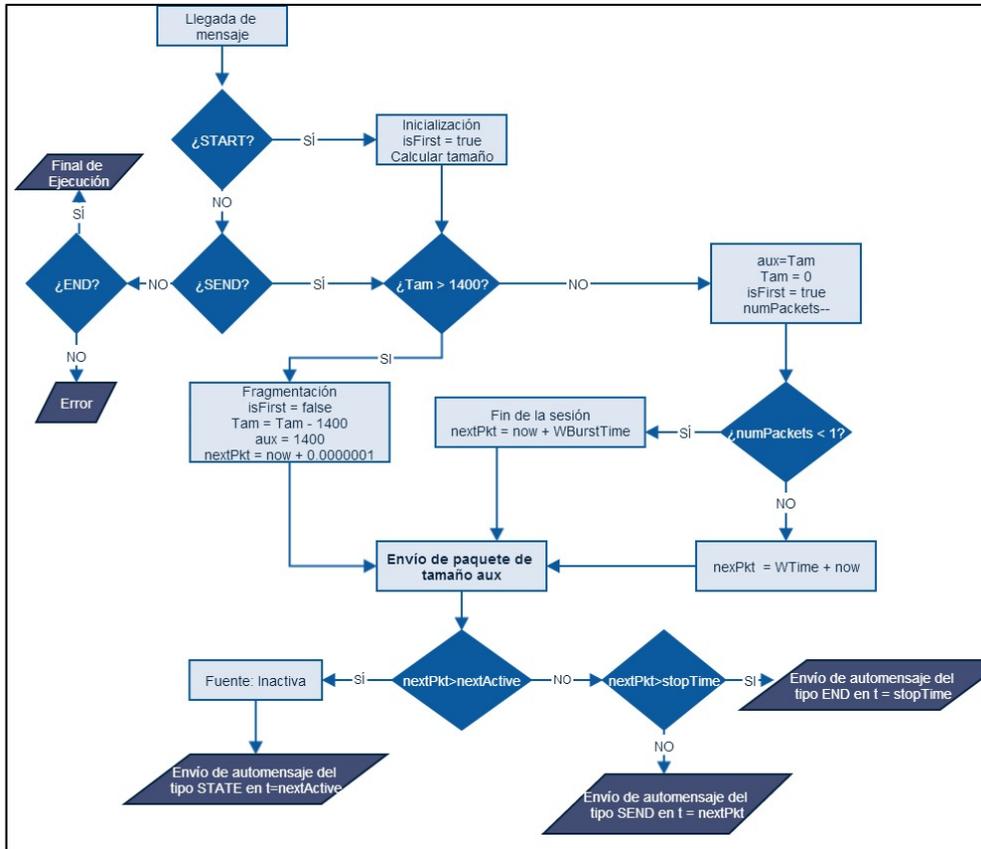


Figura 4.6 Diagrama de funcionamiento de una fuente de datos

4.3.2. Diseño del Elemento de Red e interfaz UNI Metro Ethernet de acuerdo con especificaciones servicio Macrolan de Telefónica

En este apartado se describen los elementos constitutivos y la funcionalidad del Elemento de Red que se ha definido (funciones básicas) para simular el comportamiento del interfaz de acceso proporcionado por el operador.

En este caso se ha tratado de simular el Servicio Macrolan de Telefónica [107] que utiliza como router-switch terminal Catalyst 3650 [24]. La utilización de esta solución no resta generalidad a la herramienta dado que no se utilizan parámetros de configuración específicos del fabricante. Lo que sí es imprescindible es que el número de colas, su gestión y la capacidad de proceso sean iguales o superiores al utilizado. Los principales routers utilizados por los operadores en este tipo de accesos cumplen las especificaciones marcadas.

Respecto a los parámetros de configuración del acceso, éstos son independientes del operador y están sujetos a las recomendaciones de los estándares del Metro Ethernet Forum [112].

4.3.1.4 Implementación de QoS

Se ha definido un router 3650 a la salida de cada Centro Corporativo y a la entrada de la Cloud Privada que se encargan de las tareas de QoS (DiffServ), con el objetivo de garantizar una calidad mínima a cada tipo de tráfico.

Con el fin de conseguir una simulación lo más parecida posible a un entorno real, se ha tomado como referencia el servicio Macrolan de Telefónica, donde se comercializan tres categorías de tráfico: Multimedia, Oro y Plata. El tráfico multimedia incluye el tráfico de VoIP, el tráfico Oro incluye el tráfico de vídeo y el de datos en tiempo real, mientras que el de plata incluye el tráfico best effort. El servicio Macrolan incluye también un cuarto tipo de tráfico, que es el tráfico crítico o de gestión, que es usado internamente y no se comercializa.

En la entrada del router se clasifica el tráfico filtrando según dirección de origen o de destino, puerto de origen o de destino, etc. Una vez clasificado el tráfico, se marca en el campo DSCP del paquete IP como EF (tráfico de voz), AF41 (tráfico de gestión), AF21 (tráfico oro) o AF11 (tráfico de plata)

Una vez marcadas las tramas, se implementan los mecanismos de policing asociados a cada tipo de tráfico, con los parámetros CIR, CBS, EIR y EBS.

Estos parámetros definen cuándo marcar las tramas como amarillas (descartables) , o cuándo descartar directamente las tramas.

En el caso de marcarse como amarillas, el usuario puede escoger entre cuatro maneras de tratar la trama: marcarla para que tenga una mayor posibilidad de descarte en las colas, degradarla a best effort (lo que puede implicar grandes retardos), tratarla como si fuera una trama verde (lo que podría comprometer parte del tráfico asegurado en caso de congestión), o descartarla directamente. Las tramas amarillas del tráfico de plata se configurarían automáticamente como best effort.

El tráfico plata (AF11) y el tráfico degradado a best effort técnicamente se tratan de la misma manera en el modelo, pero éste último empezará a sufrir descartes por el mecanismo RED antes que el tráfico asegurado de calidad plata.

De esta manera se garantiza un ancho de banda distinto para cada tipo de tráfico, con un posible ancho de banda de exceso.

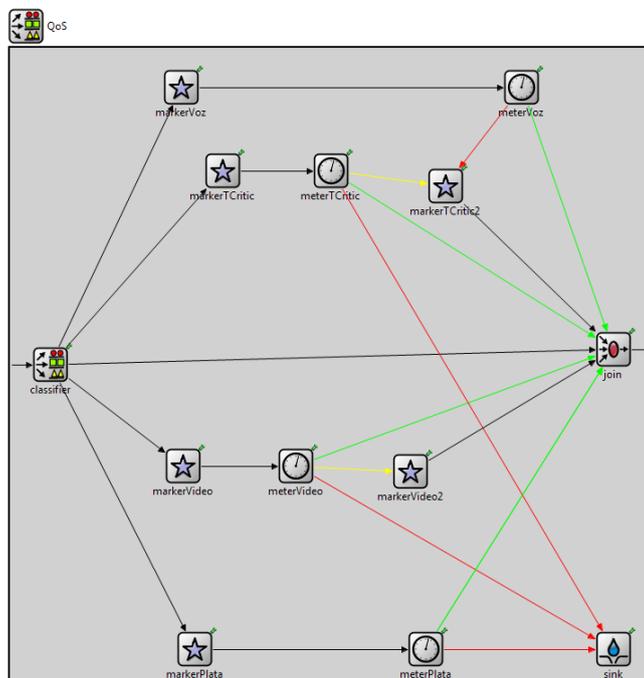


Figura 4.7 Implementación de QoS (clasificado, marcado y policing) en OMNeT++

La herramienta utiliza los clasificadores y los marcadores ya implementados en INET, y para la función de policing se creará un nuevo módulo, debido a que el módulo ya implementado falla para valores grandes.

4.3.1.5 Gestión de las colas

También se ha desarrollado un modelo básico de simulación de colas para el Switch-Router utilizado en el servicio Macrolan, el Catalyst 3650 de Cisco, que dispone de 4 colas: una para tráfico multimedia, otra para tráfico crítico (que incluye el tráfico de gestión, de control, etc.), otra para el tráfico oro, y una última para el tráfico plata.

Las colas encargadas del tráfico en tiempo real son pequeñas (10-100 paquetes) para minimizar la generación de retardos y de jitter, mientras que la cola dedicada al tráfico que no es en tiempo real es mucho mayor (500-1000 paquetes), para minimizar la pérdida de información. Sin embargo, el tamaño de las colas también es completamente configurable por el usuario.

La cola multimedia será prioritaria, mientras que las otras tres colas serán gestionadas por un WRR, cuyos pesos son completamente configurables por el usuario, de manera que se servirán por orden, y en la proporción definida por sus pesos.

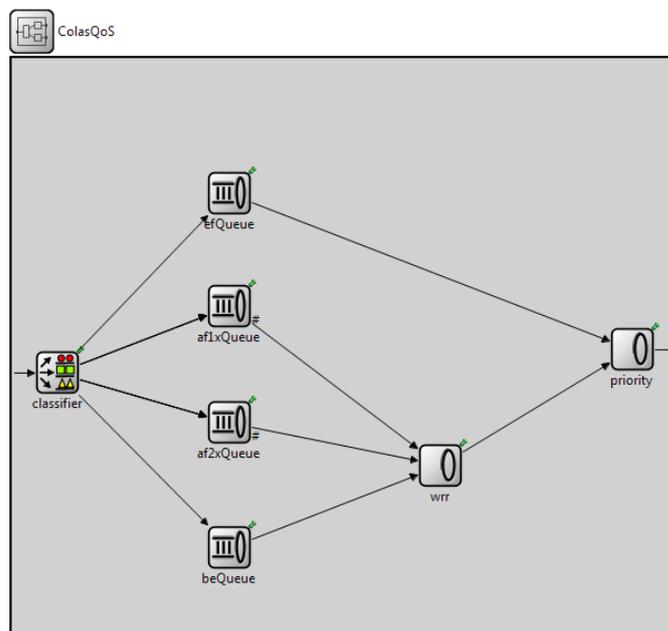


Figura 4.8 Implementación de la gestión de colas en OMNeT++

Se han usado módulos ya existentes en la implementación de la gestión de colas, incluyendo una cola DropTail, una cola FIFO + una función de descarte Random Early Detection (RED) de acuerdo con la funcionalidad descrita en la RFC 2309 [114], un planificador de prioridad basado en el algoritmo “two-rate three-color marker” descrito en [113]

Se ha modificado la función de descarte RED, de manera que el usuario puede introducir los dos valores umbrales que determinan su funcionamiento y dos posibilidades de descarte: una para tramas verdes y otra para tramas amarillas, y que genere automáticamente valores umbrales que serán usados para las tramas amarillas (mayor posibilidad de descarte), de modo que las tramas amarillas sean descartadas antes de empezar a descartar tramas verdes.

La implementación de las colas por defecto es la siguiente:

Tabla 4.4 Configuración por defecto de las colas

Tráfico	Cola	Tamaño	Scheduler
Multimedia	Droptail Queue	10	Priority
Gestión	FIFO Queue + RED	100	WRR (peso 3)
Oro	FIFO Queue + RED	100	WRR (peso 2)
Plata	FIFO Queue + RED	1000	WRR (peso 1)

4.3.1.6 Meter

Para realizar las tareas de policing se ha implementado un módulo simple, que acepta como parámetros los valores de CIR, PIR, CBS y PBS, y cuyo funcionamiento es el de un trTCM según se define en la recomendación RFC 4115 [113].

Se da la opción de que los parámetros CBS y PBS se inicialicen automáticamente al tamaño equivalente de llenado durante 1 segundo.

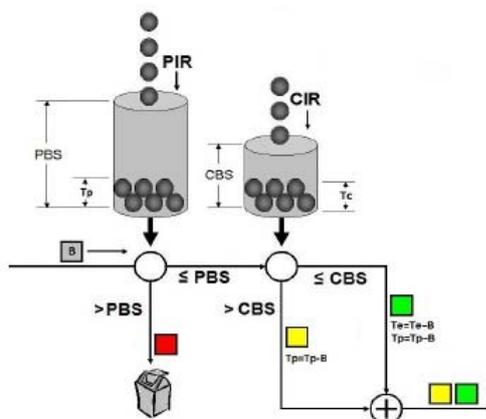


Figura 4.9 Descripción del algoritmo trTCM implementado

En un principio el cubo P se llena $PBS \cdot 8$ bits, y el cubo C con $CBS \cdot 8$ bits, es decir se llenan a la máxima capacidad posible. Cuando llega un paquete nuevo, se calcula cuántos tokens se han generado desde el anterior ($PIR \cdot t$ para el cubo P y $CIR \cdot t$ para el cubo C, donde t es el tiempo que ha pasado desde la llegada del anterior paquete, o el tiempo de simulación si es el primero).

Posteriormente se comprueba si hay suficientes tokens, siguiendo el diagrama de la Figura 4.9.

4.3.2 Módulos diseñados en base a la biblioteca INET

4.3.2.1 StandardHost

El módulo Standard Host es el usado como base tanto para las fuentes de tráfico como para los servidores.

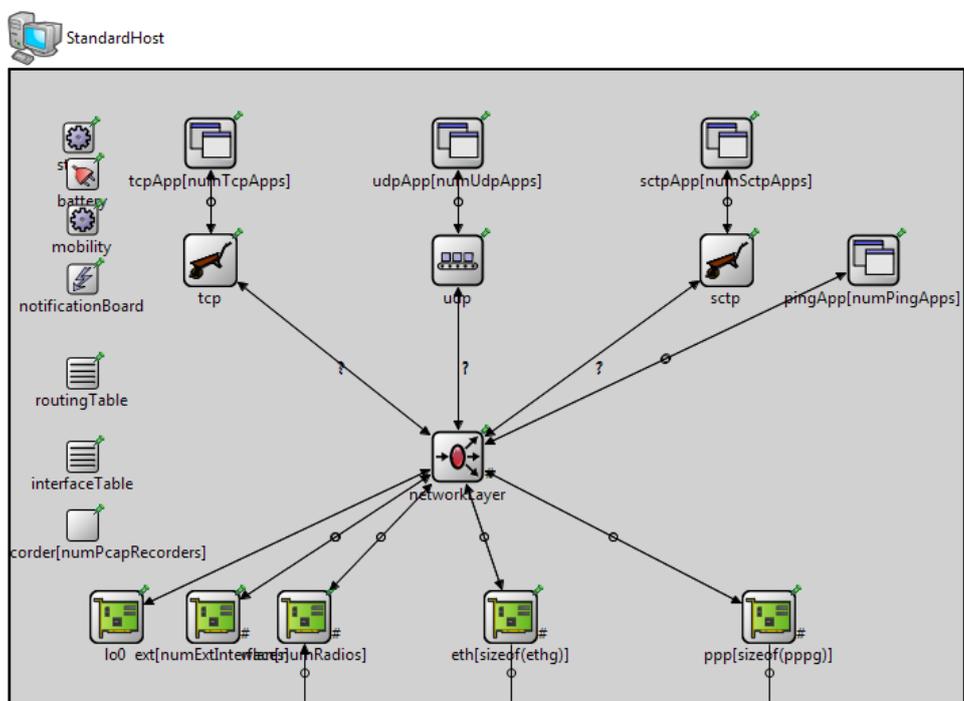


Figura 4.10 Arquitectura del módulo StandardHost

Tal y como se puede observar en la Figura 4.10, el módulo StandardHost admite aplicaciones TCP, UDP y SCTP, las cuales se deben implementar aparte. Además, un mismo Host puede implementar un número de aplicaciones configurable por el usuario, sin ninguna limitación.

En esta tesis, se van a usar aplicaciones de UDP. Concretamente 4, de las cuales 3 de ellas se han programado y se han descrito en los apartados 4.2.1 (fuentes de voz), 4.2.2 (fuentes de vídeo) y 4.2.3 (fuentes de datos); y la cuarta es una aplicación del tipo UDP-Sink, ya implementada, y cuya función es la de recopilar información de los paquetes que le llegan, y será la usada para los servidores.

En cuanto a los puertos de entrada y de salida, estos pueden ser usados para insertar el medidor de ancho de banda.

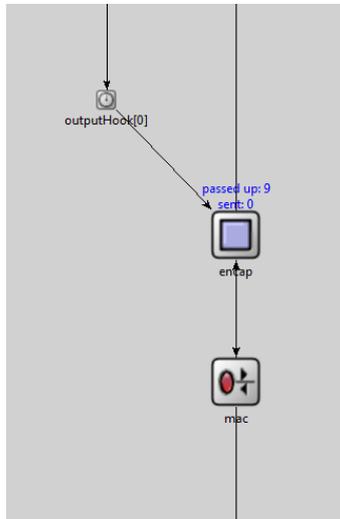


Figura 4.11 Arquitectura del puerto de Ethernet de un StandardHost

4.3.2.2 Router

Otro de los principales elementos ya existentes en INET es el router, cuya estructura es mostrada en la Figura 4.12.

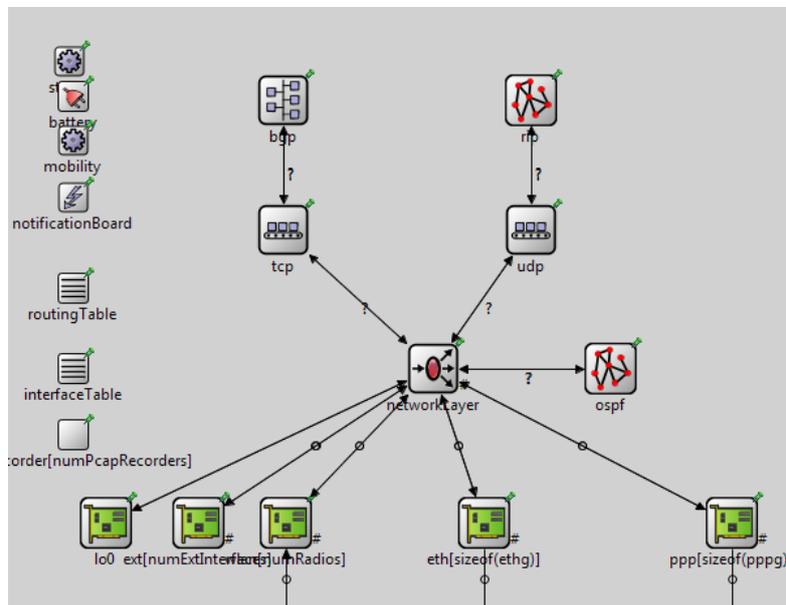


Figura 4.12 Arquitectura del módulo Router

El principal uso que se le da a los routers en esta tesis es la implementación de las funciones de QoS, las cuales se realizarán en los puertos de entrada y de salida.

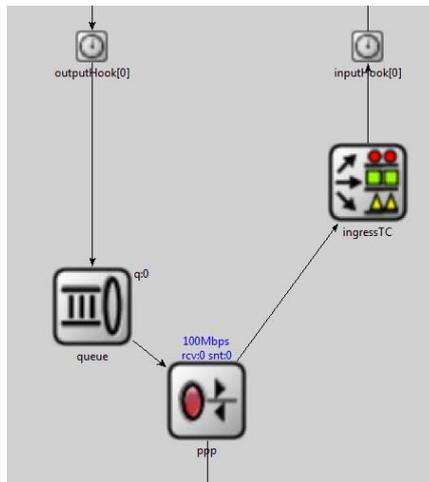


Figura 4.13 Arquitectura de un puerto PPP de entrada de un router

En la Figura 4.13 se puede observar a modo de ejemplo la entrada por un puerto PPP a un router, en el que se ha implementado el mecanismo de QoS descrito en el apartado 4.2.5 en el módulo `ingressTC`.

Las colas descritas en el apartado 4.2.6 se implementan en el módulo “queue” en los puertos de salida.

Como antes se pueden observar los medidores de ancho de banda tanto de entrada (`inputHook`) como a la salida (`outputHook`)

4.3.2.3 Internet Cloud

Para simular el retardo producido por la red MPLS junto con la tasa de pérdida de paquetes en la red, según los SLA’s del servicio, entre las fuentes de tráfico y la plataforma de ciudad. Para ello, se ha usado un módulo ya existente en INET llamado Internet Cloud, el cual usaremos para simular la Nube MPLS.

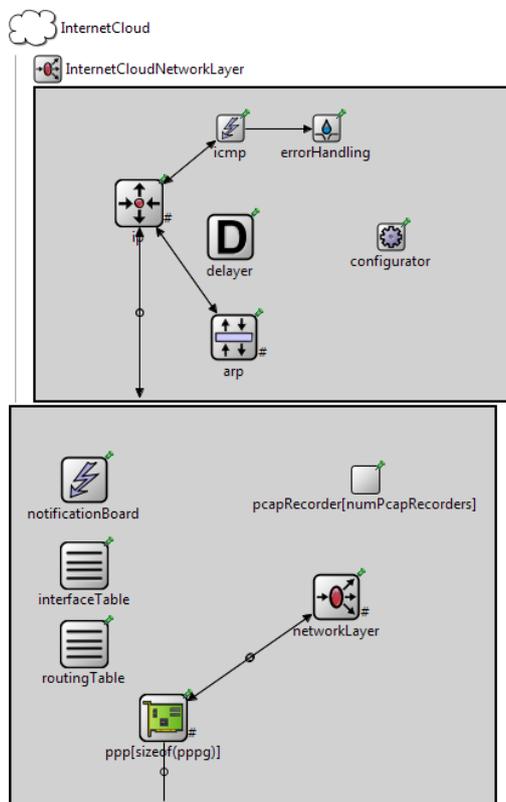


Figura 4.14 Arquitectura del módulo Internet Cloud

4.3.3 Módulos adicionales utilizados en SimulCity

4.3.3.1 Medidor de ancho de banda

La principal funcionalidad de esta herramienta es optimizar escenarios en los que hay múltiples fuentes de tráfico heterogéneo, y para ello la medida de estadísticas como el retardo, el jitter, las pérdidas o el ancho de banda es vital.

De las medidas del retardo, jitter y pérdidas de paquetes se encargan las propias aplicaciones que simulan las fuentes de tráfico y los servidores mediante **estadísticas** (paquetes enviados, paquetes recibidos, diferencia de tiempo entre salida y recepción de paquetes, etc.).

Sin embargo, para la medida del ancho de banda se ha tenido que modificar un módulo ya existente de OMNeT++, que se coloca en la entrada y en la salida de los distintos routers, y permite medir tanto número de bits recibidos, como número de paquetes recibidos, como el ancho de banda.

Para medir el ancho de banda, se define un intervalo de tiempo, de manera que durante cada intervalo se cuentan los bits que llegan, y al final del intervalo calcula el ancho de banda dividiendo el número de bits recibidos entre el tamaño del intervalo. De esta manera, con intervalos pequeños es posible observar los picos con una mayor resolución, mientras que, con un intervalo mayor, es posible ver mejor la evolución del ancho de banda medio. Por ejemplo, en el caso de una fuente de voz, con intervalos de 20 milisegundos (en el caso de G711 coincide con la separación entre paquetes), es posible observar perfectamente la distribución de On y de Off de la fuente. El código usado para programar este medidor de ancho de banda se puede encontrar en el anexo “Código”

En este escenario tendremos por defecto intervalos de 20 milisegundos para las fuentes de voz, 40 milisegundos para las fuentes de vídeo, 1 segundo para las fuentes de datos, y 1 segundo para ver el tráfico combinado a lo largo de distintas fases del escenario (a la salida de cada Centro Corporativo, o a la entrada de la cloud privada).

Las variables que controlarán el funcionamiento del programa serán las siguientes:

Tabla 4.5 Variables usadas en el medidor de ancho de banda

<i>Variable</i>	<i>Tipo</i>	<i>Función</i>
tMsg	cMessage	Automensaje de control
maxInterval	double	Duración máxima de cada intervalo
intvlStartTime	simtime t	Instante en el que acabó el último intervalo
batchSize	simtime t	Número máximo de paquetes por intervalo
intvlNumBits	long	Número de bits que han pasado en el intervalo actual
OnlyIP	bool	Número de sesiones restantes.

En t=0 se configura el automensaje para que se envíe en t=Interval, y se inicializan todas las variables a 0. La lógica del programa con la llegada de cada mensaje está descrita en la siguiente figura:

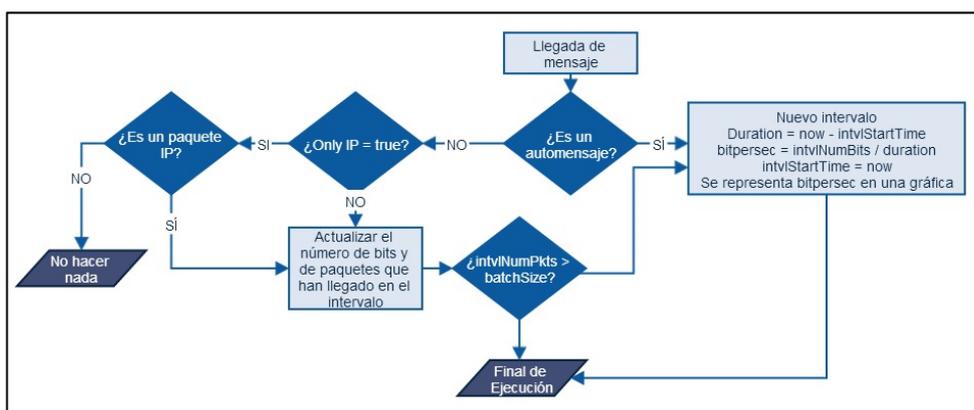


Figura 4.15 Diagrama de funcionamiento del medidor de ancho de banda

4.3.3.2 *Función Random*

Se ha creado un programa que permite obtener valores aleatorios siguiendo distribuciones estadísticas, según unos parámetros definidos en los ficheros XML de configuración de las fuentes.

En el programa se han definido distribuciones uniformes, normales y exponenciales, además de permitir valores constantes.

En el caso de una distribución normal, el usuario puede definir una media (mean), una desviación estándar (sd), un valor mínimo (min), y la posibilidad de usar una distribución normal truncada que no pueda tomar valores negativos (nonNegative=true).

En una distribución uniforme el usuario puede elegir un límite inferior y un límite superior, de manera que se elegirá un valor aleatorio entre esos límites.

Por último, para una distribución exponencial el usuario tendrá la posibilidad de elegir una media (mean), un valor mínimo (min), y un valor máximo (max), de manera que el valor aleatorio estará en el rango definido por el usuario.

4.3.3. **Implementación del escenario**

4.3.3.3 *Escenario global*

El escenario global, como se describió en el punto 4.3 está compuesto por un Centro Corporativo Municipal que se conecta a una cloud privada alojada en un Data Center que alberga la Plataforma de Ciudad y otros servidores consolidados, según se puede observar su implementación en OMNeT++ en la Figura 4.16:

De esta manera, el intercambio de información entre Centros Corporativos se puede hacer de manera sencilla, al igual que el acceso a información como pueda ser los resultados de unas pruebas médicas, el historial de un paciente, etc.

Para modelar este escenario, se ha usado un vector de módulos que representan un Centro Corporativo, un módulo que representa la nube MPLS, y un módulo que representa la cloud privada que incluye a los servidores consolidados.

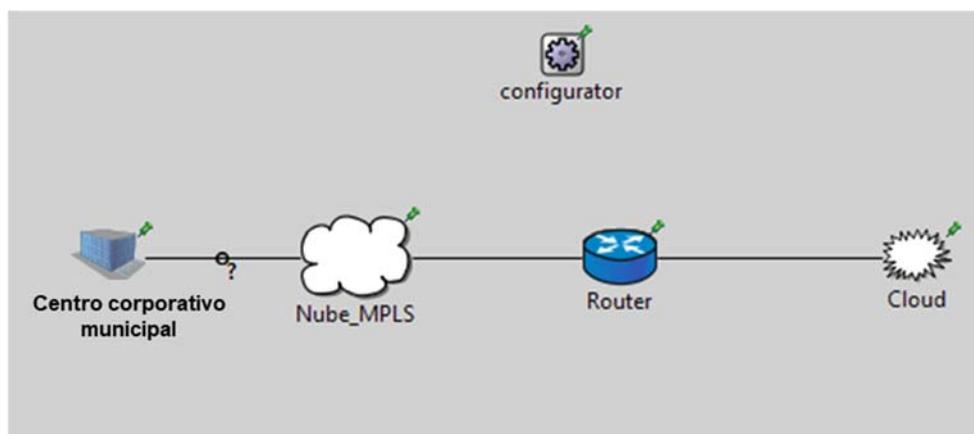


Figura 4.16 Implementación en OMNeT++ del escenario global

El router a la salida de la nube MPLS es usado para configurar la QoS a la entrada de la Cloud, donde el tráfico de todos los Centros Corporativos se une, y es posible que se produzca congestión. Los módulos del Centro Corporativo, de la nube MPLS y de la Cloud serán explicados más adelante.

La velocidad del enlace entre el módulo Centro Corporativo y la nube MPLS es configurable, y permite al usuario elegir entre un enlace de 10Mbps, de 100Mbps o de 1Gbps. La velocidad del enlace entre la nube MPLS y la cloud es también configurable, y puede adoptar cualquier valor definido por el usuario.

4.3.3.4 Módulo Centro Corporativo Municipal

El módulo Centro Corporativo será el encargado de dos funciones principales: la caracterización de las fuentes de tráfico y del router Cisco 3650, que será el encargado de realizar las funciones de QoS a la salida del Centro Corporativo, donde se provisiona un enlace Metro Ethernet, que puede ser de 10Mbps, 100Mbps o de 1Gbps, configurable por el usuario.

En la Figura 4.17 se describe el Módulo Centro Corporativo donde se dibuja un conjunto arbitrario de fuentes.

- Fuentes de voz para los servicios de telefonía corporativa y Comunicaciones Unificadas.
- Fuentes de vídeo para los servicios de control de accesos, gestión de tráfico, y video vigilancia entre otros.
- Fuentes de Datos para dispositivos IoT, sistemas de información municipal, acceso de usuarios, control de edificios, etc.

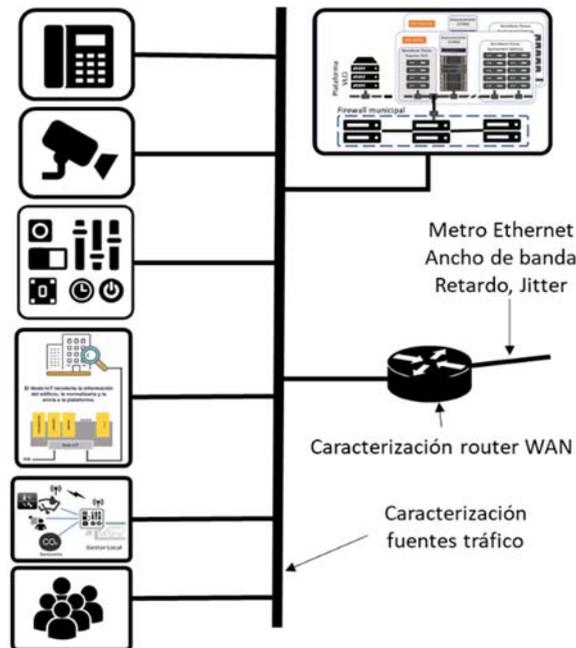


Figura 4.17 Módulo Centro Corporativo

Con el fin de facilitar la implementación en OMNeT++, facilitar la obtención de estadísticas se ha definido un módulo específico para cada tipo de fuente de tráfico.

- Fuentes de voz.
- Fuentes de vídeo.
- Fuentes de datos / IoT.

Dentro de este módulo tendremos a su vez tres módulos que representan los tipos de servicios, un switch que une todo el tráfico generado por el Centro Corporativo, y el router de salida.

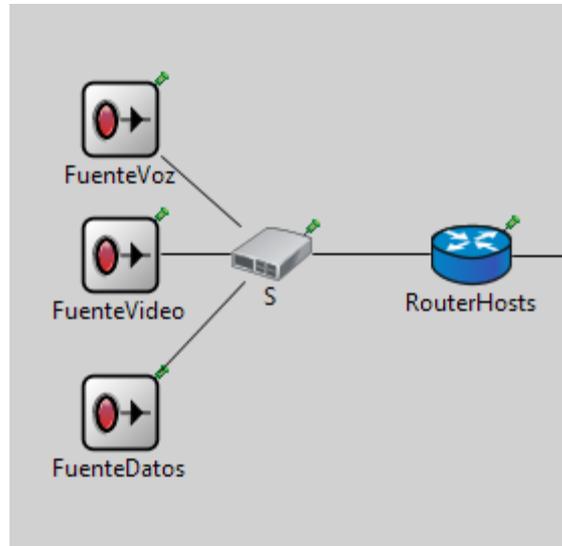


Figura 4.18 Implementación del módulo Centro Corporativo en OMNeT++

Dentro de los módulos de las fuentes, se encuentra un vector de hosts que representan los servicios ofrecidos por el Centro Corporativo, tal y como se muestra en la Figura 4.19. Habrá tantos hosts como servicios ofrezca el Centro Corporativo, y cada host simulará el tráfico generado por un número configurable de fuentes reales que ofrezcan ese servicio. De esta manera, se puede configurar en un primer lugar los servicios que ofreceremos en los Centros Corporativos de nuestro escenario, clasificados según el tipo de tráfico, y posteriormente podremos escalar el escenario, indicando el número de fuentes de tráfico para cada servicio y para cada Centro Corporativo.

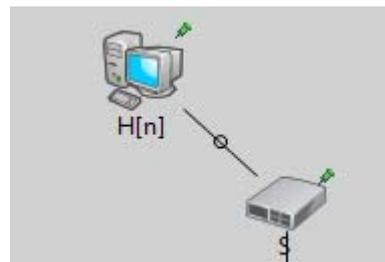


Figura 4.19 Implementación del módulo Fuente en OMNeT++

Tanto el host como el switch presentes en la Figura 4.19 y el router presente en la Figura 4.18 son módulos definidos en la librería INET.

El switch se encarga de juntar todo el tráfico, al igual que en el módulo anterior.

4.3.3.5 Módulo Nube MPLS

El módulo de la nube MPLS es el que simula todo el proceso desde que el tráfico sale del Centro Corporativo hasta que llega a la cloud privada.

El principal objetivo de este módulo es simular el retardo que se pueda producir durante el proceso de MPLS, y las pérdidas de paquetes que se puedan producir. Para ello, se hará uso del módulo de la librería INET de OMNeT++ “Internet Cloud”, que funciona como un router simplificado que tendrá tantas entradas como Centros Corporativos tenga el escenario, y una salida hacia la cloud privada. Además, se incluye un Delayer y un Dropper.

El módulo “Internet Cloud” permite además construir escenarios con varios centros corporativos, este sería el caso de una arquitectura no centralizada donde cada edificio municipal dispone de un acceso a la cloud privada.

En la Figura 4.20 se dibuja un escenario general con esta arquitectura.

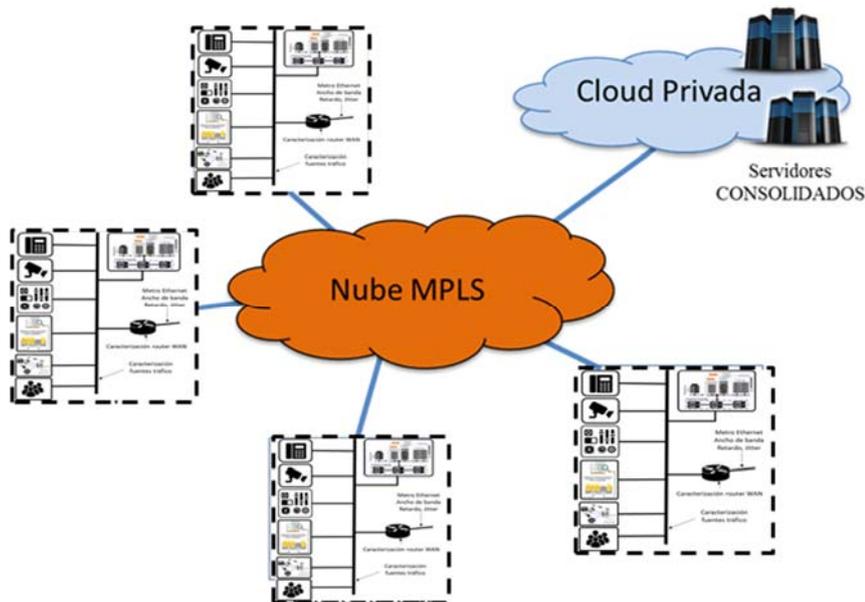


Figura 4.20 Nube MPLS

De esta manera, el usuario puede definir en un fichero XML las características de cada uno de los enlaces entre cada Centro Corporativo y la cloud privada. A partir de una dirección de origen y una dirección destino, es posible definir la velocidad del enlace, el retardo (el tiempo que tarda en atravesar la nube MPLS), y una posibilidad de descartar la trama. Estos parámetros se comprueban cada vez que se envía un nuevo paquete por

el enlace, por lo que pueden tomar valores fijos o valores aleatorios siguiendo distribuciones estadísticas. En el caso de la posibilidad de pérdida de paquetes, crearemos un valor aleatorio, y definiremos un umbral: si el valor aleatorio es menor que el umbral, se descartará el paquete.

Además, al estar definido en un fichero XML, es posible crear varias configuraciones que definan distintos estados de la red. Por ejemplo, una configuración que simule el retardo producido por el MPLS, y otra sin retardo ni posibilidad de descarte de paquetes que nos permita medir los resultados como si la nube estuviera a la salida del Centro Corporativo.

4.3.3.6 Módulo Cloud

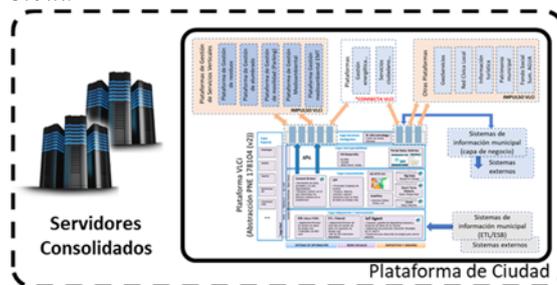


Figura 4.21 Cloud privada con los servidores remotos

El módulo Cloud será el encargado de almacenar los servidores consolidados. Estos servidores tendrán en esta simulación el papel de recolectores de estadísticas, para calcular retardos, pérdidas de paquetes y el jitter en cada servicio.

La implementación en OMNeT++ será la mostrada en la Figura 4.22

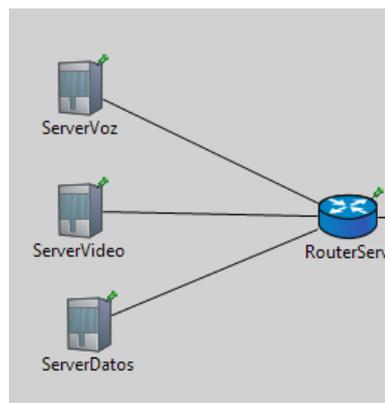


Figura 4.22 Implementación del módulo Cloud en OMNeT++

La implementación es muy parecida a la del módulo Centro Corporativo, clasificando los servidores según el tipo de tráfico que reciben: voz, vídeo o datos. Como antes, el motivo de esta clasificación es para facilitar la creación de las gráficas de retardo, jitter y paquetes recibidos y enviados.

La Figura 4.23 representa el contenido de los módulos ServerVoz, ServerVideo y ServerDatos.

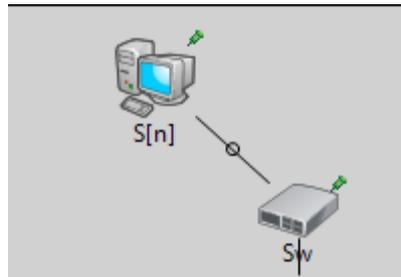


Figura 4.23 Implementación del módulo Server en OMNeT++

Por cada aplicación tendremos un host que hará las funciones de servidor para ese servicio.

4.3.4 Configuración del escenario

4.3.4.1 Configuración del códec de voz: CodecVoz.xml

Para configurar las fuentes de tráfico de voz, en primer lugar, el usuario debe seleccionar el códec a emplear.

Como el objetivo de esta simulación es caracterizar el tráfico en la red, sin tener en cuenta los retardos producidos por el códec al codificar o decodificar la voz, se define el códec mediante el tamaño de cada paquete de voz y el intervalo de tiempo entre paquetes en un fichero XML, donde hay varios códecs definidos por defecto.

En concreto, serán los siguientes:

Tabla 4.6 Códecs de voz implementados

Códec	Tamaño de la carga útil (B)	Intervalo entre paquetes (ms)
G.711	160	20
G.729	20	20
G.723.1	24	30
G.726	80	20
G.728	60	30

A los tamaños de la carga útil hay que añadirles 12 bytes de RTP.

La estructura del fichero XML es la siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Definición de los atributos de los distintos códecs de voz so-
portados. -->
<root>
  <voice-codec id="g.711">
    <PacketSize type="constant" value="172" />
    <SendInterval type="constant" value="0.02" />
  </voice-codec>
  <voice-codec id="g.729">
    <PacketSize type="constant" value="32" />
    <SendInterval type="constant" value="0.02" />
  </voice-codec>
  <voice-codec id="g.723.1">
    <PacketSize type="constant" value="36" />
    <SendInterval type="constant" value="0.03" />
  </voice-codec>
  <voice-codec id="g.726">
    <PacketSize type="constant" value="92" />
    <SendInterval type="constant" value="0.02" />
  </voice-codec>
  <voice-codec id="g.728">
    <PacketSize type="constant" value="72" />
    <SendInterval type="constant" value="0.03" />
  </voice-codec>
  <voice-codec id="Custom">
    <PacketSize type="constant" value="172" />
    <SendInterval type="constant" value="0.02" />
  </voice-codec>
</root>
```

Figura 4.24 Configuración del códec de voz

4.3.4.2 Configuración de la calidad de vídeo: QualityVideo.xml

Para configurar el tráfico de vídeo, en primer lugar, el usuario debe indicar en un fichero XML la resolución horizontal y vertical del cuadro (X, Y). En la Figura 4.25, se muestran 4 calidades definidas. SimulCity no impone ninguna restricción a estos valores.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Resolución para distintas calidades. -->
<root>
  <video-quality id="SD">
    <ResX type="constant" value="320" />
    <ResY type="constant" value="180" />
  </video-quality>

  <video-quality id="HQ">
    <ResX type="constant" value="640" />
    <ResY type="constant" value="360" />
  </video-quality>

  <video-quality id="HD">
    <ResX type="constant" value="1280" />
    <ResY type="constant" value="720" />
  </video-quality>

  <video-quality id="HD+">
    <ResX type="constant" value="1920" />
    <ResY type="constant" value="1080" />
  </video-quality>
</root>
```

Figura 4.25 Configuración de las calidades de vídeo

4.3.4.3 Configuración del tamaño de los ficheros de las fuentes de datos: Data-Size.xml

Se ha creado el fichero DataSize.xml para la configuración de los parámetros relacionados con el tamaño de los ficheros generados por una fuente de datos, como son el número de recursos de texto, el número de imágenes, y el tamaño de cada uno de ellos.

El formato del fichero XML es el mostrado en la Figura 4.26:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Definición del tamaño de los ficheros de datos a enviar. -->
<root>
  <data-size id="residuos">
    <textResourceSize type='exponential' mean='5000' min='100'
max='10000' />
    <imageResourceSize type='exponential' mean='300000'
min='100000' max='500000' />
    <numResources type='uniform' beginning='0' end='10' />
    <numImages type='uniform' beginning='0' end='2' />
  </data-size>
  <data-size id="ilum">
    <textResourceSize type='exponential' mean='100000'
min='10000' max='1000000' />
    <imageResourceSize type='exponential' mean='300000'
min='100000' max='500000' />
    <numResources type='uniform' beginning='0' end='10' />
    <numImages type='uniform' beginning='0' end='2' />
  </data-size>
  <data-size id="medioamb">
    <textResourceSize type='exponential' mean='1000000'
min='100000' max='3000000' />
    <imageResourceSize type='exponential' mean='10000000'
min='1000000' max='30000000' />
    <numResources type='uniform' beginning='0' end='2' />
    <numImages type='uniform' beginning='0' end='2' />
  </data-size>
  <data-size id="cloud">
    <textResourceSize type='exponential' mean='1000000'
min='100000' max='1000000' />
    <imageResourceSize type='exponential' mean='200000'
min='100000' max='500000' />
    <numResources type='uniform' beginning='0' end='5' />
    <numImages type='uniform' beginning='0' end='5' />
  </data-size>
  <data-size id="database">
    <textResourceSize type='exponential' mean='1000' min='100'
max='5000' />
    <imageResourceSize type='exponential' mean='2000' min='1000'
max='20000' />
    <numResources type="constant" value="1" />
    <numImages type="constant" value="0" />
  </data-size>
  <data-size id="servidores">
    <textResourceSize type='exponential' mean='1000' min='100'
max='1000' />
    <imageResourceSize type='exponential' mean='200000'
min='10000' max='500000' />
    <numResources type='uniform' beginning='0' end='5' />
    <numImages type='uniform' beginning='0' end='2' />
  </data-size>
</root>
```

Figura 4.26 Contenido del fichero DataSize.xml

4.3.4.4 Configuración de los perfiles de fuentes de tráfico: ConfigTraffic.xml

Por último, para la configuración que definen el comportamiento del tráfico (tiempo de actividad y de inactividad para las fuentes de voz y vídeo, cantidad de movimiento y cuadros por segundo para fuentes de vídeo, y la caracterización de las ráfagas en las fuentes de datos), se ha creado el fichero ConfigTraffic.xml.

Este fichero tiene el siguiente formato:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Traffic profiles configuration. -->
<root>
  <voice-profile id="low">
    <Duration type='exponential' mean='2' min='1' max='3' />
    <TimeInactive type='exponential' mean='60' min='0' max='120' />
  </voice-profile>
  <voice-profile id="high">
    <Duration type='exponential' mean='3' min='1' max='6' />
    <TimeInactive type='exponential' mean='2' min='0' max='10' />
  </voice-profile>
  <video-profile id="vigilancia">
    <Duration type='exponential' mean='7' min='3' max='15' />
    <TimeInactive type='exponential' mean='5' min='0' max='20' />
    <Movement type="constant" value="2" />
    <fps type="constant" value="25" />
  </video-profile>
  <data-profile id="60x60">
    <numPackets type="constant" value="1" />
    <SendInterval type="constant" value="1" />
    <numBursts type="constant" value="60" />
    <BurstInterval type="constant" value="60" />
  </data-profile>
  <data-profile id="30x120">
    <numPackets type="constant" value="1" />
    <SendInterval type="constant" value="1" />
    <numBursts type="constant" value="30" />
    <BurstInterval type="constant" value="120" />
  </data-profile>
</root>
```

Figura 4.27 Configuración de los perfiles de tráfico

Los parámetros de tiempo de las fuentes de voz y de vídeo se definen en minutos, mientras que los parámetros de tiempo de datos se definen en segundos.

De esta manera, mediante la combinación de ficheros XML se pueden definir estructuras arbitrarias con un elevado grado de flexibilidad. Por ejemplo, se puede generar una fuente de tráfico de una videoconferencia de distintas calidades, o una fuente de datos que envíe avisos de estado de luminarias.

4.3.4.5 Configuración del filtrado: filters.xml

Como se ha explicado en el apartado 3, la clasificación del tráfico se hace filtrando por los puertos de llegada.

La equivalencia entre el puerto de salida del clasificador que corresponde a un tipo de tráfico concreto y el puerto o los puertos usados para identificarlo se realiza en el fichero filters.xml

```
<filters>
  <filter destPort='1000' gate="0"/>
  <filter destPort='1001' gate="1"/>
  <filter destPort='1002' gate="2"/>
  <filter destPort='1003' gate="3"/>
</filters>
```

Figura 4.28 Configuración del filtrado

4.3.4.6 Configuración del escenario: OMNeTpp.ini

Por último, en el fichero OMNeTpp.ini, el usuario puede configurar todos los detalles del escenario, asignando valores a las distintas variables.

La configuración del escenario se ha dividido en cinco grandes bloques:

- **Configuración general del escenario.** En primer lugar, el usuario define unos parámetros de configuración generales del escenario. Estos parámetros incluyen la duración de la simulación, el número de centros corporativos, la velocidad del enlace de entrada a la cloud privada, y opcionalmente se pueden indicar valores globales que van a afectar a todas las fuentes, por ejemplo, si se va a usar el régimen transitorio o el régimen permanente en el escenario.

En la Figura 4.29 se puede observar un ejemplo de configuración general, donde se ha definido una duración de 3600 segundos (1 hora), una velocidad del enlace de la cloud de 10 Gbps, y se ha indicado que el escenario estará compuesto por un centro corporativo, y que el escenario comenzará en régimen transitorio..Como se puede observar, en esta sección del fichero de configuración, se implementan además los mecanismos de QoS y de colas al escenario, y se le asigna a la nube MPLS su fichero de configuración.

```

#####
# #
# CONFIGURACIÓN GENERAL #
# #
#####

##### Duración #####
sim-time-limit = 3600s#1800s

##### Implementación de Los módulos de QoS #####
**.RouterHosts.eth[*].ingressTCType = "QoS"
**.Router.ppp[0].ingressTCType = "QoS"
**.RouterHosts.ppp[*].queueType = "ColasQoS"
**.Router.ppp[1].queueType = "ColasQoS"
**.Nube_MPLS.ppp[*].queueType = "ColasQoS"

##### Fichero configuración de La nube MPLS #####
**.delayer.config = xmldoc("internetCloud.xml")

## Configuración de Los servidores y de parámetros de fuentes ##
**.Server*.**.numUdpApps = 1
**.Server*.**.udpApp[*].typename = "UDPSink"
**.Fuente*.**.localPort=-1 #Puerto efímero

##### Comportamiento general #####
#Régimen transitorio o permanente
**.transit = true #Verdadero indica que se arrancará el escenario de
0. Falso simula un régimen permanente.
#Enlace de salida
**.CDatarate = 10Gbps #Velocidad del enlace de entrada a La cloud
privada
#Número de centros corporativos
**.NumCentros=1 #Número total de centros corporativos

```

Figura 4.29 Configuración general del escenario

- **Definición de aplicaciones.** En segundo lugar, se definen las aplicaciones que van a ser usadas en el escenario por cualquiera de los Centros Corporativos, tanto de voz, como de imágenes como de datos.

Este bloque se divide a su vez en tres sub bloques, uno para las fuentes de voz, otro para las de vídeo, y otro para las de datos.

Primero se define el número de aplicaciones que se van a implementar, y posteriormente se configuran los parámetros de cada una de las aplicaciones, los cuales serán los siguientes:

- destAddresses. El nombre del servidor al que irá dirigido el tráfico.
- startTime. El tiempo de inicio de la aplicación.
- stopTime. El instante donde acaba la ejecución de la fuente
- destPort. El puerto de destino. Es usado para identificar cada tipo de tráfico.
- rndStart. Parámetro opcional que indica si el tiempo de inicio es aleatorio o no.
- Transit. Parámetro opcional que indica si la fuente está en régimen transitorio o permanente.
- Codec. En el caso de las fuentes de voz, selecciona el códec.
- Quality. En el caso de las fuentes de vídeo, selecciona la calidad.
- DataSize. En el caso de las fuentes de datos, selecciona el tamaño del fichero.
- conFigura Selecciona la configuración del perfil de tráfico.

```
##### Fuentes de tráfico de voz #####
**. *Voz.n=2
**. FuenteVoz.H[*].udpApp[*].typename = "UDPVoz"

# Fuente 1 (Voz)
**. FuenteVoz.H[0].udpApp[*].destAddresses = "Cloud.ServerVoz.S[0]"
**. FuenteVoz.H[0].udpApp[*].chooseDestAddrMode = "once"
**. FuenteVoz.H[0].udpApp[*].startTime = uniform(0s,900s)
**. *Voz.*[0].udpApp[*].*Port = 1000 #Tráfico Multimedia (1000 = mul-
timedia, 1001 = oro, 1002 = crítico, 1003 = plata)
**. FuenteVoz.H[0].udpApp[*].codec = xmldoc("VoiceCo-
dec.xml", "//voice-codec[@id='g.711']")
**. FuenteVoz.H[0].udpApp[*].config = xmldoc("ConfigTraf-
fic.xml", "//voice-profile[@id='low']")
```

Figura 4.30 Definición de una fuente de voz

En la Figura 4.30 se puede apreciar un ejemplo de configuración de una fuente de voz. En primer lugar, se define el número de aplicaciones de voz que definiremos en el escenario que en este caso es 2, y después se define el tipo de fuentes que son aplicaciones del tipo UDPVoz.

Posteriormente, se ha definido una de las aplicaciones, definiendo el destino (Cloud indica que está en la nube Cloud, ServerVoz indica que es un servidor de aplicaciones de voz, y S[0] indica que es la primera aplicación de voz definida), el tiempo de inicio (en este caso es aleatorio con probabilidad uniforme entre 0 y 2400 segundos), el instante en el que se desconecta la fuente (t=3600s), el puerto de destino (1000, indicando que es tráfico multimedia), el códec (g.711), y la configuración del tráfico (perfil de voz del tipo low).

```
##### Fuentes de vídeo #####
**.*Video.n=3
**.*FuenteVideo.H[*].udpApp[*].typename = "UDPVideo"

# Fuente 1 (Vídeo)
**.*FuenteVideo.H[0].udpApp[*].destAddresses = "Cloud.ServerVi-
deo.S[0]"
**.*FuenteVideo.H[0].udpApp[*].chooseDestAddrMode = "once"
**.*FuenteVideo.H[0].udpApp[*].startTime = uniform(0s,900s)
**.*Video.*[0].udpApp[*].*Port = 1001 #Tráfico Oro (1000 = multime-
dia, 1001 = oro, 1002 = crítico, 1003 = plata)
**.*FuenteVideo.H[0].udpApp[*].quality = xmldoc("VideoQual-
ity.xml", "//video-quality[@id='HD']")
**.*FuenteVideo.H[0].udpApp[*].config = xmldoc("ConfigTraf-
fic.xml", "//video-profile[@id='vigilancia']")
```

Figura 4.31 Definición de una fuente de vídeo

En la Figura 4.31 se puede observar un ejemplo de configuración de una fuente de vídeo. La diferencia con respecto al caso anterior es que, en lugar de definir el códec, definimos la calidad del vídeo.

```
##### Fuentes de tráfico de tiempo real #####
**.*Datos.n=4
**.*FuenteDatos.H[*].udpApp[*].typename = "UDPDatos"
**.*FuenteDatos.H[*].udpApp[*].chooseDestAddrMode = "once"

# Fuente 1 (Datos)
**.*FuenteDatos.H[0].udpApp[*].destAddresses = "Cloud.ServerDatos.S[0]"
**.*FuenteDatos.H[0].udpApp[*].startTime = uniform(0s,180s)
**.*Datos.*[0].udpApp[*].*Port = 1001 #Tráfico Oro (1000 = multimedia,
1001 = oro, 1002 = crítico, 1003 = plata)
**.*FuenteDatos.H[0].udpApp[*].data_size =
xmldoc("DataSize.xml", "//data-size[@id='cronicos']")
**.*FuenteDatos.H[0].udpApp[*].config = xmldoc("ConfigTraf-
fic.xml", "//data-profile[@id='60x60']")
```

Figura 4.32 Definición de una fuente de datos

En la Figura 4.32 se ha configurado una fuente de datos. De nuevo, la configuración es muy parecida a los casos anteriores.

- **Escalado de Centros Corporativos.** Una vez definidas todas las aplicaciones que se van a modelar en el escenario a simular, el siguiente paso es el escalado de los Centros Corporativos. Para ello, para cada uno de ellos se define la capacidad del enlace de salida (10Mbps, 100Mbps o 1Gbps), el número de fuentes de tráfico que implementan cada aplicación, y se configuran los parámetros de QoS del router de salida del Centro Corporativo.

```

#####
# #
# ESCALADO DE Centros #
# #
#####

##### Centro Corporativo GRANDE #####
####Tipo de Centro
**.CentroCor[0].Datarate = 2 # 0 = 10Mbps, 1 = 100Mbps, 2 = 1Gbps

####Número de fuentes
**.CentroCor[0].FuenteVoz.H[0].numUdpApps = 300 #200
**.CentroCor[0].FuenteVoz.H[1].numUdpApps = 20 #20
**.CentroCor[0].FuenteVideo.H[0].numUdpApps = 10 #5
**.CentroCor[0].FuenteVideo.H[1].numUdpApps = 20 #1
**.CentroCor[0].FuenteVideo.H[2].numUdpApps = 1 #10
**.CentroCor[0].FuenteDatos.H[0].numUdpApps = 500 #500
**.CentroCor[0].FuenteDatos.H[1].numUdpApps = 50 #100
**.CentroCor[0].FuenteDatos.H[2].numUdpApps = 5 #10
**.CentroCor[0].FuenteDatos.H[3].numUdpApps = 100 #5
**.CentroCor[0].FuenteDatos.H[4].numUdpApps = 0 #50
**.CentroCor[0].FuenteDatos.H[5].numUdpApps = 0 #100

####QoS en el router de salida del Centro Corporativo
#Multimedia
**.CentroCor[0].RouterHosts.**.ingressTC.meterVoz.cir = "2Mbps"
**.CentroCor[0].RouterHosts.**.ingressTC.meterVoz.cbs = 250KiB
**.CentroCor[0].RouterHosts.**.ingressTC.meterVoz.pir = "4Mbps"
**.CentroCor[0].RouterHosts.**.ingressTC.meterVoz.pbs = 500KiB
#Oro
**.CentroCor[0].RouterHosts.**.ingressTC.meterOro.cir = "40Mbps"
**.CentroCor[0].RouterHosts.**.ingressTC.meterOro.cbs = 5MiB
**.CentroCor[0].RouterHosts.**.ingressTC.meterOro.pir = "80Mbps"
**.CentroCor[0].RouterHosts.**.ingressTC.meterOro.pbs = 10MiB
#Gestión
**.CentroCor[0].RouterHosts.**.ingressTC.meterGestion.cir = "0Mbps"
**.CentroCor[0].RouterHosts.**.ingressTC.meterGestion.cbs = 0KiB
**.CentroCor[0].RouterHosts.**.ingressTC.meterGestion.pir = "2Mbps"
**.CentroCor[0].RouterHosts.**.ingressTC.meterGestion.pbs = 12.5MiB
#Plata
**.CentroCor[0].RouterHosts.**.ingressTC.meterPlata.cir = "20Mbps"
**.CentroCor[0].RouterHosts.**.ingressTC.meterPlata.cbs = 2.5MiB
**.CentroCor[0].RouterHosts.**.ingressTC.meterPlata.pir = "100Mbps"
**.CentroCor[0].RouterHosts.**.ingressTC.meterPlata.pbs = 12.5MiB

```

Figura 4.33 Escalado del escenario

En la Figura 4.33 se puede observar un ejemplo en el que se define un escenario con un único Centro Corporativo.

Se configura el Centro Corporativo con un enlace de salida de 1 Gbps, y posteriormente se indica el número de fuentes que será implementadas de las definidas en la parte anterior en el Centro Corporativo 0 (el único que hay). Finalmente, se configuran los parámetros de QoS.

- **Medidor de Ancho de Banda.** Posteriormente se configuran los medidores de anchos de banda, asignando valores a los parámetros. Normalmente esta parte no se modifica por el usuario, pero es totalmente configurable.

```

#####
# #
# MEDIDOR DE ANCHO DE BANDA #
# #
#####
##### Asignación de Los medidores #####
**.numOutputHooks = 1 #A La salida de todos Los hosts y routers
**.Router.ppp[0].numInputHooks = 1 #A La entrada del router de La
cloud
**.Router.ppp[0].inputHook[0].typename = "ThruputMeter"
**.Route*.*.outputHook[0].typename = "ThruputMeter"
**.Nube_MPLS.ppp[0].numInputHooks = 1 #A La entrada del router de La
cloud
**.Nube_MPLS.ppp[0].*putHook[0].typename = "ThruputMeter"
**.Server*.*.numInputHooks = 1 #A La entrada de Los servers
**.Server*.*.inputHook[0].typename = "ThruputMeter"
**.H[*].eth[*].outputHook[0].typename = "ThruputMeter"

```

Figura 4.34 Configuración del medidor de ancho de banda

En la Figura 4.34 se puede observar la configuración del medidor de ancho de banda. En primer lugar se colocan en los puntos estratégicos que nos interesan: a la salida de todos los routers y hosts, y a la entrada del router de entrada a la cloud y de los servers.

Posteriormente se configuran los distintos parámetros relacionados con el medidor de ancho de banda.

- **QoS General.** Se configuran los parámetros de QoS del router de entrada a la cloud, que es el que recibe el tráfico de todos los Centros Corporativos.

```

#####
# QoS GENERAL #
#####
##### Policing #####
#Multimedia
**.Route***.ingressTC.meterVoz.cir = "200Mbps" #35% total
**.Route***.ingressTC.meterVoz.cbs = 25MiB
**.Route***.ingressTC.meterVoz.pir = "400Mbps" #35% total
**.Route***.ingressTC.meterVoz.pbs = 50MiB
#Oro
**.Route***.ingressTC.meterOro.cir = "200Mbps" #60% total
**.Route***.ingressTC.meterOro.cbs = 25MiB
**.Route***.ingressTC.meterOro.pir = "400Mbps" #100% total
**.Route***.ingressTC.meterOro.pbs = 50MiB
#Gestión
**.Route***.ingressTC.meterGestion.cir = "200Mbps" #5% total
**.Route***.ingressTC.meterGestion.cbs = 25MiB
**.Route***.ingressTC.meterGestion.pir = "400Mbps"
**.Route***.ingressTC.meterGestion.pbs = 50MiB
#Plata
**.Route***.ingressTC.meterPlata.cir = "200Mbps" #0% total
**.Route***.ingressTC.meterPlata.cbs = 25MiB
**.Route***.ingressTC.meterPlata.pir = "400Mbps" #100% total
**.Route***.ingressTC.meterPlata.pbs = 50MiB
#Tratamiento de tramas amarillas
**.Route*.*.ActionMultimedia = 1 # 0 = Se trata como una trama verde, 1
= Se degrada se calidad, 2 = Se trata como best effort, 3 = Se descarta
**.Route*.*.ActionOro = 2
**.Route*.*.ActionGestion = 1
##### Gestión de colas #####
#Multimedia
**.CMult = 30 #Tamaño
#Oro
**.COroMin = 100 #Tamaño para empezar a descartar tramas verdes
**.COroDC1 = 0.25 #Probabilidad de descarte de tramas verdes
**.COroDC2 = 0.5 #Probabilidad de descarte de tramas amarillas
**.COroMax = 150 #Tamaño para descartar todas Las tramas verdes
#Gestion
**.CGestionMin = 100 #Tamaño para empezar a descartar tramas verdes
**.CGestionDC1 = 0.25 #Probabilidad de descarte de tramas verdes
**.CGestionDC2 = 0.5 #Probabilidad de descarte de tramas amarillas
**.CGestionMax = 150 #Tamaño para descartar todas Las tramas verdes
#Plata
**.CPlataMin = 700 #Tamaño para empezar a descartar tramas verdes
**.CPlataDC1 = 0.25 #Probabilidad de descarte de tramas verdes
**.CPlataDC2 = 0.5 #Probabilidad de descarte de tramas amarillas
**.CPlataMax = 1000 #Tamaño para descartar todas Las tramas verdes
#WRR
**.weights = "1 2 1" #Pesos para tráfico de gestión, oro y plata

```

Figura 4.35 Configuración de los parámetros de QoS generales

En el ejemplo de la Figura 4.35, se han definido los parámetros de la función policía del router de entrada a la cloud, y también se han configurado las políticas para el tratamiento de las tramas amarillas. Por último, se han definido los tamaños de las colas, las probabilidades de descarte, y los pesos del WRR.

4.3.4.7 Configuración de la Nube MPLS: internetCloud.xml

En la Figura 4.24 podemos ver un ejemplo de configuración de una nube MPLS:

```
<internetCloud symmetric="true">
  <parameters name="good">
    <traffic src="CentroCor[0].*" dest="*" delay="0ms" data-
rate="1000Mbps" drop="false" />
    <traffic src="CentroCor[1].*" dest="*" delay="15ms+truncnor-
mal(10ms,2ms)" datarate="150Mbps" drop="false" />
    <traffic src="CentroCor[2].*" dest="*" delay="6ms+truncnor-
mal(10ms,2ms)" datarate="50Mbps" drop="false" />
    <traffic src="CentroCor[3].*" dest="*" delay="17ms+truncnor-
mal(10ms,2ms)" datarate="50Mbps" drop="false" />
  <!--
  <traffic src="*" dest="*" delay="10ms+truncnormal(100ms,20ms)" da-
tarate="uniform(100kbps,500kbps)" drop="uniform(0,100) &lt; uni-
form(1, 5)" />
  -->
  </parameters>
</internetCloud>
```

Figura 4.36 Parámetros usados para la configuración de la nube MPLS

En este ejemplo se han definido 4 Centros Corporativos, lo que equivale a definir 4 enlaces Centro Corporativo-Cloud. Además, se ha configurado para cada uno de ellos un retardo aleatorio, un datarate, y una posibilidad de pérdida de paquetes.

El datarate debería coincidir con el del enlace de salida del Centro Corporativo, o ser mayor. Si se configura un datarate inferior, todo el proceso de QoS a la salida del centro corporativo no será válido.

4.3.5 Recopilación de estadísticas

Una de las principales funcionalidades de esta herramienta es la medida de los distintos parámetros de calidad del escenario simulado. Por ello, se hace necesario el uso de herramientas que nos permitan recopilar estadísticas sobre parámetros como el número de paquetes enviados, el flujo de tráfico en un momento determinado, el número de paquetes recibidos, el instante en el que se envía un paquete, etc.

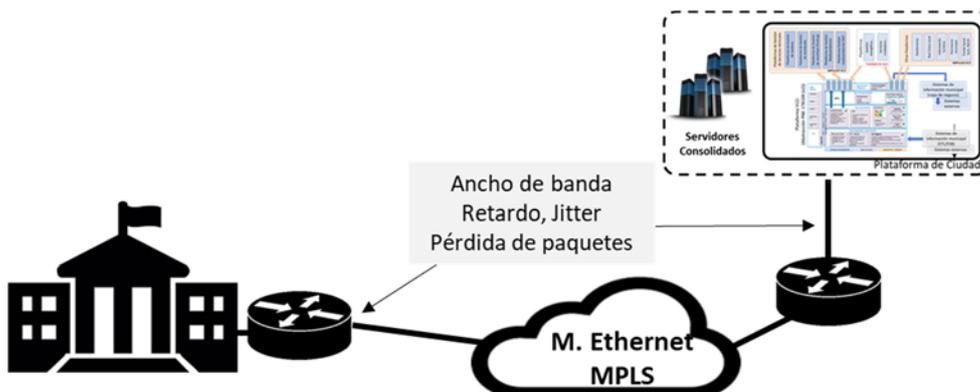


Figura 4.37 Recopilación de estadísticas

4.3.5.1 Ancho de banda

Para la medida del ancho de banda se ha usado el medidor programado en C++ explicado anteriormente.

Tanto en los puertos PPP como en los puertos de Ethernet de los routers y de los hosts, es posible configurar un “gancho” de entrada (inputHook) o de salida (outputHook), donde es posible instalar el medidor de ancho de banda, de manera que podamos medir el caudal a la entrada y a la salida de todos los routers y hosts del escenario, para posteriormente mostrar el ancho de banda en el punto que nos interese del escenario.

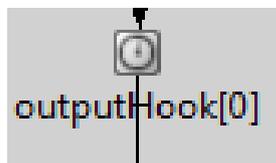


Figura 4.38 Módulo dentro de un router donde se implementa un medidor de ancho de banda

4.3.5.2 Paquetes recibidos y enviados: Pérdidas de paquetes

Todas las fuentes de tráfico y todos los servidores son en realidad instancias de standardHost, unos módulos ya creados en la librería INET que simulan el comportamiento de un host, y que por efecto acumulan estadísticas como el número de paquetes total enviados y el número total de paquetes recibidos.

Por la arquitectura jerárquica que se ha elegido para el desarrollo del modelo, es posible diferenciar fácilmente los hosts que corresponden a cada aplicación, lo que hace posible la creación de unas gráficas que muestran el número total de paquetes enviados y recibidos para cada aplicación.

En el caso del tráfico de vídeo y del tráfico de datos, hay que tener en cuenta que un paquete no siempre equivale a un cuadro o a un fichero, pues forzamos una fragmentación para que el tamaño del paquete no supere el máximo permitido por el protocolo IP.

4.3.5.3 Retardos

Cuando la fuente de tráfico envía un paquete, se guarda en él el instante en el que se produce el envío. De esa manera, cuando el servidor recibe el paquete, puede calcular automáticamente el retardo que ha sufrido el paquete con una simple resta.

La arquitectura elegida nos permite crear gráficas del retardo de todas las aplicaciones de voz, de todas las aplicaciones de vídeo y de todas las aplicaciones de datos por separado, pudiendo analizar el comportamiento según el tipo de tráfico por separado.

4.3.5.4 Jitter

OMNeT++ permite operar sobre todas las estadísticas, tanto vectores como escalares, lo cual usamos para calcular el jitter a partir del retardo. Para ello, se ha usado la operación diferencia sobre las estadísticas que contienen los retardos.

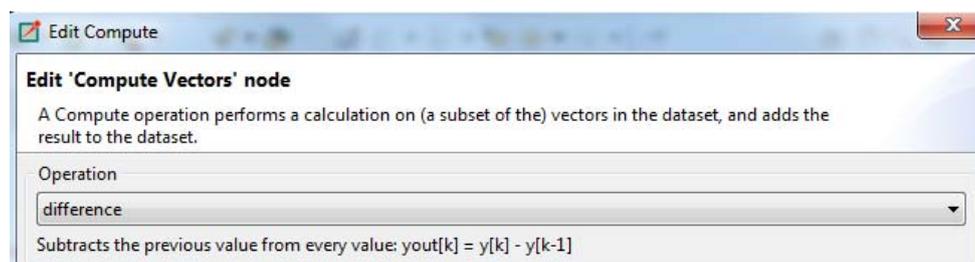


Figura 4.39 Operador de vectores para calcular el jitter a partir del vector de retardos

4.3.5.5 Dataset: Creación de gráficas

En la simulación del escenario, se generan un número muy grande de estadísticas, lo que hace muy complicado el estudio de las estadísticas que realmente nos interesan. Por eso, en OMNeT++ se permite la creación de unos Datasets, que permiten seleccionar las estadísticas que nos puedan interesar y generar nuestras gráficas con la información que nos interese. En la Figura 4.40 se puede apreciar el Dataset seleccionado.

Se han escogido las estadísticas escalares de paquetes enviados por las fuentes de tráfico y recibidos por los servidores, y las estadísticas vectoriales que representan el retardo de los servidores y el ancho de banda de todos los hosts y routers.

Con esas estadísticas SimulCity es capaz de proporcionar un total de 14 gráficas diferentes que incluyen los paquetes enviados y recibidos, el retardo, el jitter, y el ancho de banda de todas las aplicaciones definidas por el usuario, separando entre aplicaciones de voz, de vídeo y de datos. Además, se han generado 2 gráficas extras que muestran el ancho de banda a la salida de un Centro Corporativo y a la entrada del cloud.

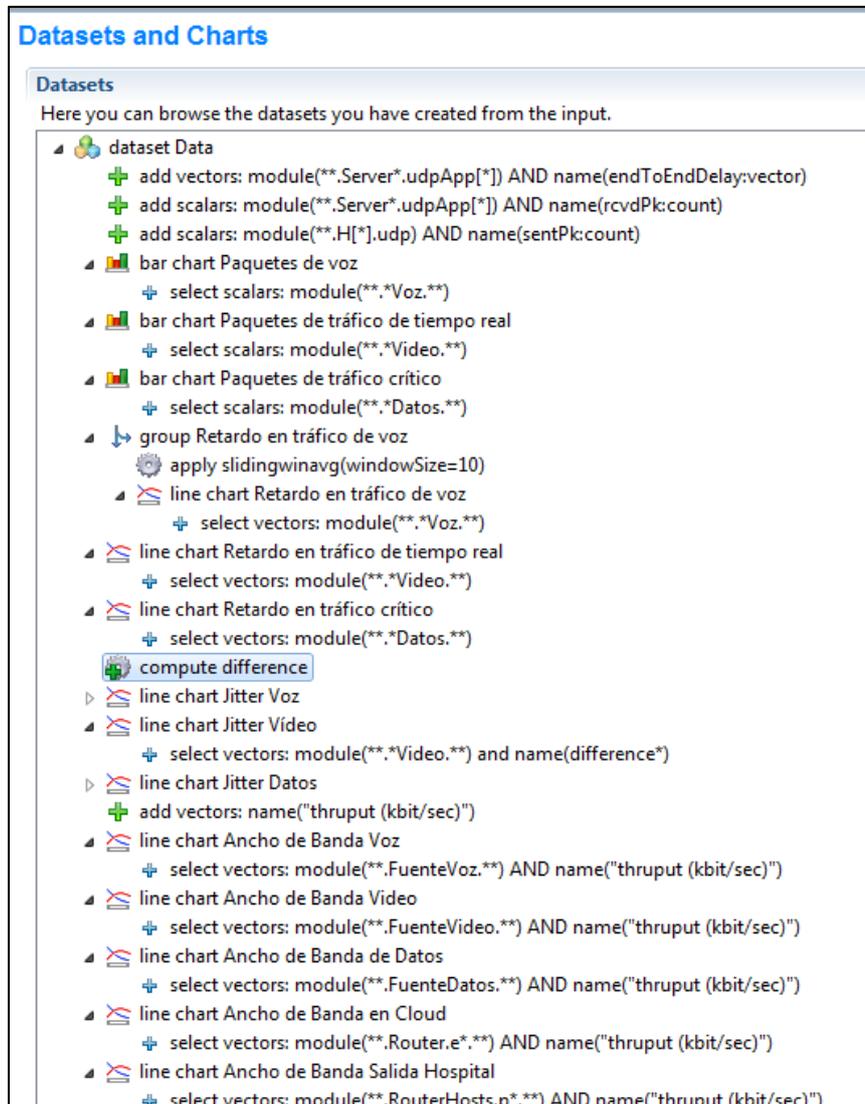


Figura 4.40 Dataset y gráficas empleadas en el modelo

En las figuras 4.41, 4.42 y 4.43 se muestran, a título ilustrativo, el ancho de banda de 5 fuentes de VoIP, el retardo extremo a extremo y el ancho de banda a la salida del interfaz Metro-Ethernet de la totalidad del tráfico corporativo.

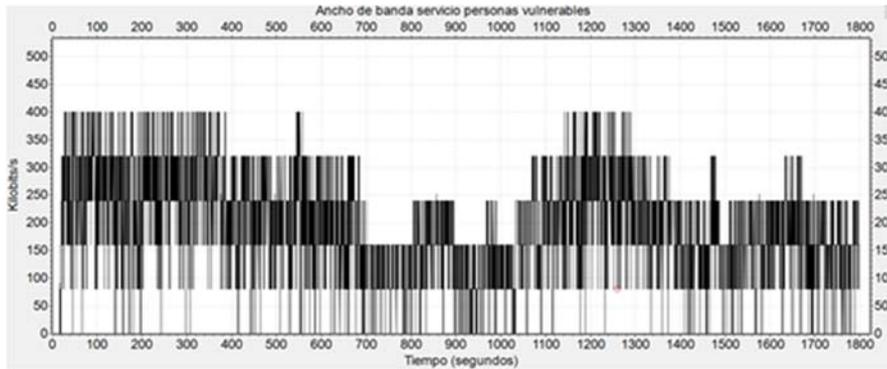


Figura 4.41 Ancho de banda de 5 fuentes de VoIP (perfil de tráfico "low")

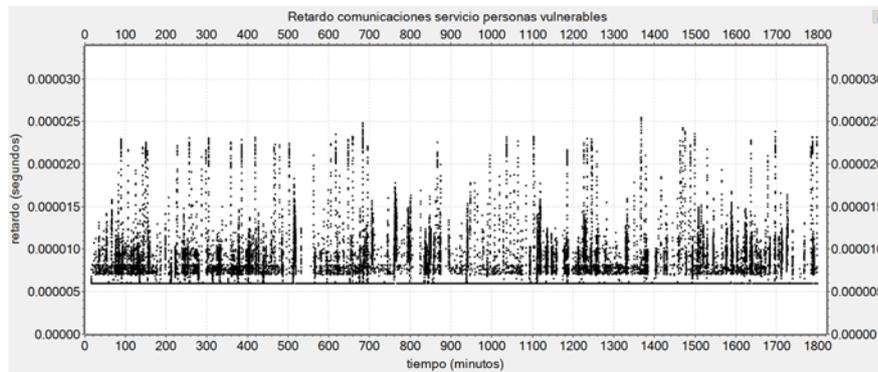


Figura 4.42 Retardo extremo a extremo de 5 fuentes de VoIP (perfil de tráfico "low")

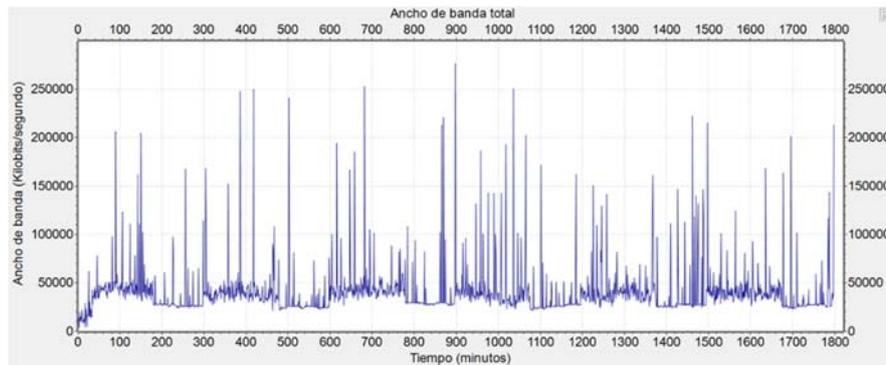


Figura 4.43 Ancho de banda total en el interfaz Metro-Ethernet

4.4 Evolución de SimulCity

De la experiencia de uso de SimulCity se han identificado diversas funcionalidades que son susceptibles de mejora y la necesidad de ampliar sus capacidades para ampliar su ámbito de uso. Dentro de las futuras versiones se incluirían mejoras en prestaciones, modelado de fuentes y usabilidad.

- SimulCity genera un volumen de información que crece potencialmente con el número de fuentes a simular y que en gran parte de los casos no es estrictamente necesaria. La consecuencia es el consumo de una capacidad de proceso significativa que resta velocidad a la simulación. Una posible línea de mejora sería configurar previamente a la simulación los vectores y escalares a evaluar.
- La flexibilidad de la herramienta permite simular una gran variedad de fuentes, pero sería conveniente que SimulCity pudiera procesar fuentes modeladas por la extracción de tráfico real de las fuentes. Esta facilidad permitiría estudiar aspectos como el grado de autosimilitud del tráfico
- La introducción de los parámetros de configuración en SimulCity se realiza actualmente mediante ficheros XML que es una tarea tediosa y sujeta a errores. En este sentido, se desarrollará un interfaz gráfico que permita la configuración de las fuentes de tráfico y parámetros de dimensionado de la red y calidad de servicio bajo un entorno gráfico. Además, se realizará la comprobación automática de la coherencia de los valores introducidos en la herramienta, evitando de esta manera errores en la simulación.

En la Figura 4.44 se puede observar el interfaz de configuración de un escenario con múltiples sedes corporativas que se conectan a la plataforma de ciudad.

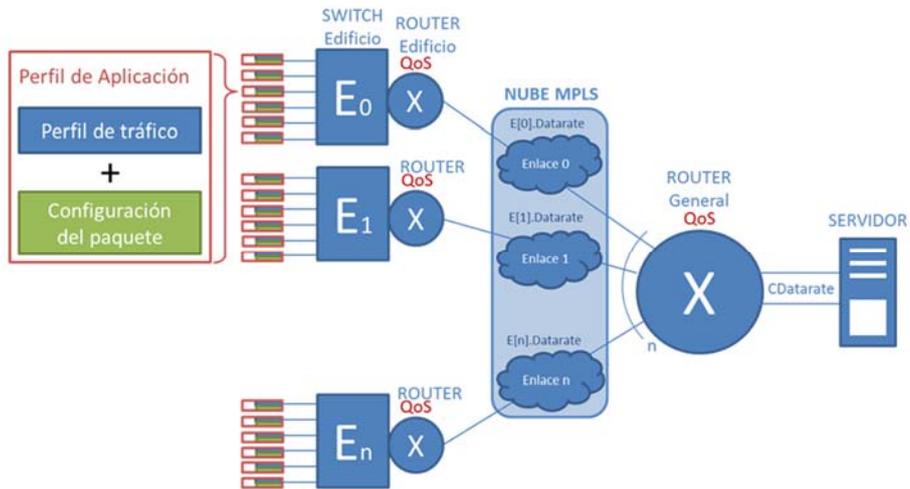


Figura 4.44 Nuevo interfaz SimulCity múltiples dependencias

En la Figura 4.46 se describe el nuevo interfaz de SimulCity para configurar cada sede que simplificará drásticamente las tareas de simulación. La Tabla 4.7 describe los elementos configurables de SimulCity a través del interfaz gráfico.

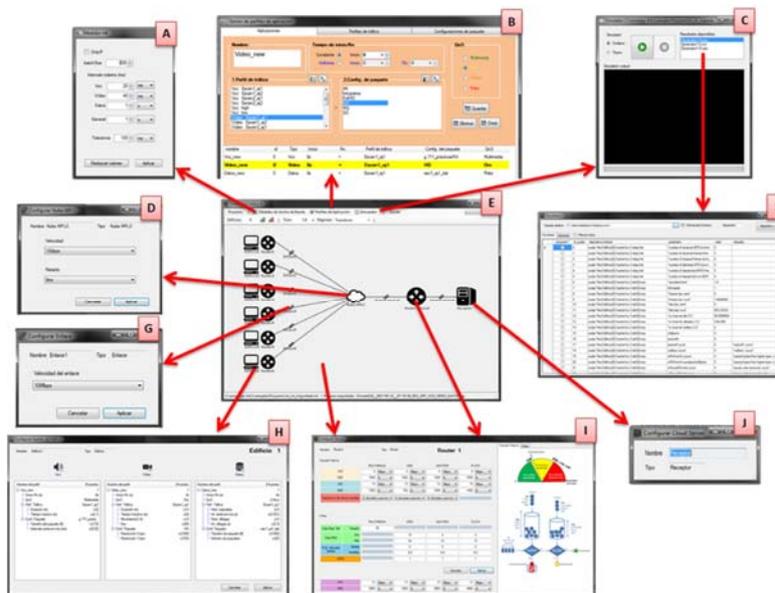


Figura 4.45 Nuevo interfaz SimulCity

Tabla 4.7 Descripción de los elementos configurables en SimulCity

ID	Descripción breve
A	Permite configurar los parámetros del medidor de ancho de banda. Al ajustarlo la medida del ancho de banda es más certera.
B	Se encarga de gestionar los Perfiles de Aplicación, CoS, los Perfiles de Tráfico y las Configuraciones de los Paquetes.
C	Maneja el proceso de simulación. Permite seleccionar el entorno de simulación, arrancar y detener el proceso, y acceder a los resultados.
D	Permite configurar la nube MPLS
E	Es el formulario de inicio de la aplicación.
F	Se accede desde el FrmSimulador. Permite ver los resultados de escalares y vectores en formato tabla y exportarlos a .csv.
G	Permite modificar el valor de la velocidad del enlace.
H	Se encarga de gestionar las fuentes que generan tráfico en cada edificio. También permite ver las características de cada fuente.
I	Gestiona los parámetros de QoS de los routers de salida de los edificios y del router de entrada del receptor. Función policía y colas.
J	Formulario preparado para, en un futuro, poder modificar parámetros en el receptor.
K	Si se abre un fichero omnetpp.ini no válido la aplicación salta a este formulario y fuerza al usuario a reparar el fichero.
L	Permite cambiar el Directorio de Trabajo
M	Muestra la versión actual, un registro de cambios, agradecimientos y menciones

Por último, en la Figura 4.45 se muestra el nuevo configurador de caudales y otros elementos relacionados con la calidad del servicio.

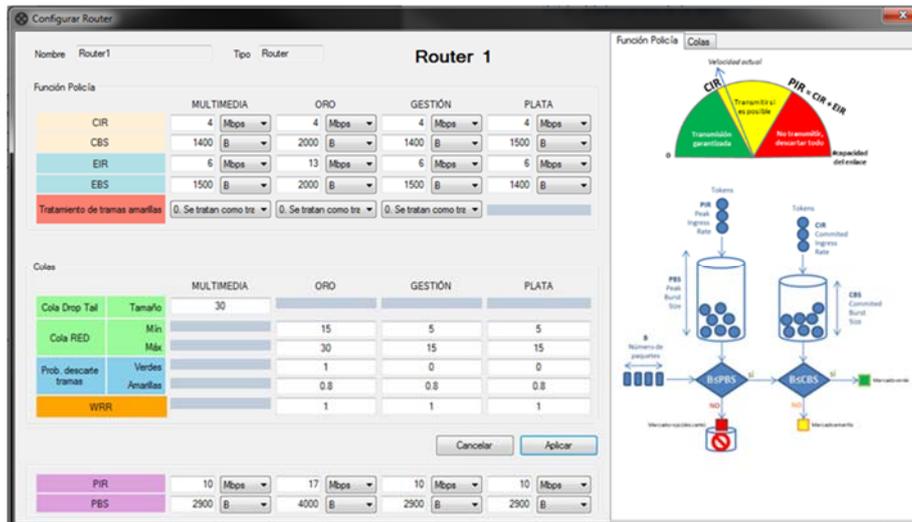


Figura 4.46 Nuevo interfaz SimulCity parámetros de control de la calidad de servicio

Capítulo 5

Casos de uso de SimulCity en proyectos VLCi del Ayuntamiento de València

5.1 Introducción

Este capítulo tiene como objetivo demostrar las funcionalidades de la herramienta Simulcity en la tarea de evaluar el impacto sobre la infraestructura de comunicaciones municipal de la ejecución de diferentes proyectos enmarcados dentro de la Estrategia de Ciudad Inteligente (VLCi) del Ayuntamiento de València.

Actualmente existen diferentes iniciativas municipales en el ámbito de las Ciudades Inteligentes que implican el despliegue de dispositivos IoT y la modernización de infraestructuras y servicios de comunicaciones en el entorno municipal. Destacan especialmente los proyectos derivados de la iniciativa “Impulso VLCi” y el estudio sobre la modernización de la red corporativa de comunicaciones que incluye la posible migración de la red de telefonía actual basada en TDM a una solución de comunicaciones unificadas [118] sobre servidores centralizados compartiendo la misma arquitectura de alojamiento que la Plataforma de Ciudad.

Dado que estos proyectos aún no están en fase de explotación, se propondrá una línea de investigación que profundice en la comparativa entre los resultados proporcionados por SimulCity y valores reales, así como la solución de conectividad, mejoras en la herramienta o el análisis del tráfico agregado real.

En los próximos apartados se realizará una breve descripción de los proyectos que el autor considera más alineados con el objetivo de esta tesis, se analizará su impacto en términos de consumo de ancho de banda y calidad de servicio, y por último se evaluará

el agregado del ancho de banda necesario entre la red corporativa municipal y la Plataforma de Ciudad para garantizar los niveles de calidad de servicio comprometidos.

Una posible infraestructura genérica municipal se detalló en el capítulo 3, donde las posibles fuentes de tráfico que suministrarán información a la plataforma de ciudad serán aquellas que dispongan de conectividad a través de alguno de los mecanismos de acceso disponibles:

- Usuarios y dispositivos conectados directamente a la Red Corporativa Municipal.
- Usuarios y dispositivos conectados a través de servicios móviles M2M VPN-IP.
- Usuarios y dispositivos conectados a través de servicios VPN-IP
- Usuarios y dispositivos conectados a través de internet.

El acceso a la Plataforma de Ciudad ubicada en el CPD de Telefónica utiliza el servicio Macrolan proporcionado por el operador y se materializa en la red de acceso a través de un enlace de 100 Mb/s basado en el estándar Metro Ethernet.

5.2 Iniciativa Impulso VLCi

El Ayuntamiento de València fue seleccionado el 25 de agosto de 2016 como beneficiario de la Convocatoria de concesión de ayudas para el desarrollo del Programa de Ciudades Inteligentes de la Agenda Digital para España de la Entidad Pública Empresarial Red.es.

Esta iniciativa que comprende 17 proyectos (expedientes 113/17-SP y 127/17-SP de la Entidad Pública Red.es) [6]. En la fecha de la redacción de esta tesis los expedientes están en fase de adjudicación.

Impulso VLCi propone un conjunto coordinado de proyectos que tienen como fin la mejora sustancial de la calidad de vida de los ciudadanos, la mejora de la eficiencia de los servicios municipales y la compartición de soluciones con otras corporaciones.

La Iniciativa Impulso VLCi afecta a prácticamente la totalidad de las dimensiones de una Ciudad Inteligente definidas en la Guía Metodología de Ciudades Inteligentes de Red.es [119] y en la norma AENOR PNE 178201: “Definición, atributos y requisitos de una Ciudad Inteligente “[49].

En la Figura 5.1 se describen los 17 proyectos que componen la Iniciativa Impulso VLCi en relación con las Dimensiones de una Ciudad Inteligente. Los proyectos se centran fundamentalmente en áreas de mejora de la eficiencia, sostenibilidad, datos abiertos, reutilización de soluciones y el bienestar ciudadano.

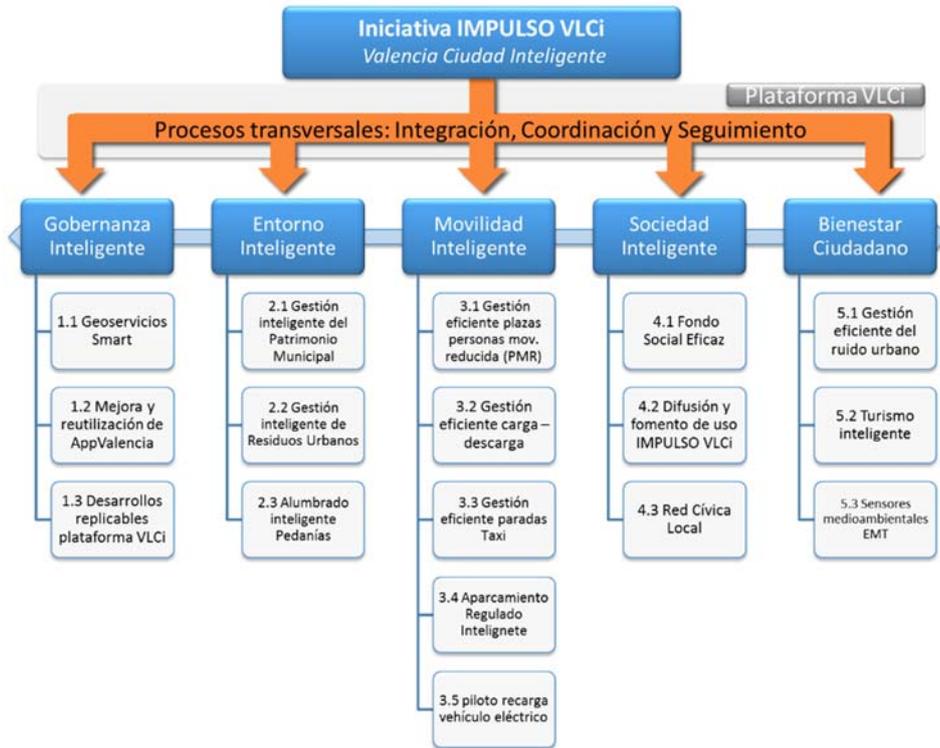


Figura 5.1 Proyectos Iniciativa Impulso VLCi

De esta iniciativa no existen proyectos que generen volúmenes de información elevados, pero se han seleccionado tres proyectos significativos que se han considerado ilustrativos para el objeto de esta tesis.

- Gestión de residuos sólidos urbanos no perecederos.
- Gestión de la red de alumbrado municipal.
- Gestión eficiente de plazas de aparcamiento para personas con movilidad reducida, servicios de carga y descarga y paradas de taxi.

En las siguientes tablas se describen brevemente los objetivos cualitativos de cada proyecto seleccionado, los resultados esperados, y el grado de impacto relacionado con la mejora de la ciudad y calidad de vida de los ciudadanos.

Tabla 5.1 Proyectos en el ámbito del Entorno Inteligente

1. Ámbito del Entorno Inteligente
Alumbrado inteligente en Pedanías dependientes del Ayuntamiento de Valencia
Descripción y objetivos: <ul style="list-style-type: none">• Dotar de una red basada en sensores y actuadores que gestione y supervise el correcto funcionamiento de las luminarias instaladas en las pedanías dependientes del Ayuntamiento de Valencia dado que su gestión y operación es muy costosa por la dispersión geográfica y los continuos robos que se producen.• No se contempla la sustitución de luminarias por no adaptarse a los objetivos de la convocatoria y ser objeto de otras líneas de financiación.
Resultados cualitativos: <ul style="list-style-type: none">• Mejorar la disponibilidad del Servicio de Alumbrado en las pedanías del Ayuntamiento de Valencia y disminuir costes dotando de inteligencia a las luminarias. La dispersión geográfica hace muy costoso su mantenimiento y supervisión incidiendo seriamente en los tiempos de reparación que son elevados.• Mejorar la seguridad de las personas dado que los accesos a las pedanías discurren normalmente entre acequias con un tráfico elevado de vehículos de dos ruedas.• Eliminar los robos de cobre que asolan a las pedanías del Ayuntamiento coordinando la información en tiempo real obtenida de la red de gestión, con avisos a las áreas de mantenimiento y a los Cuerpos de Seguridad Correspondientes a través de la Plataforma VLCi.
Grado de impacto: <ul style="list-style-type: none">• Incide en mejora de la eficiencia, sostenibilidad, innovación y automatización de métricas en la ciudad.

Tabla 5.2 Proyectos en el ámbito de la Movilidad Inteligente

2. Ámbito de la Movilidad Inteligente
Gestión eficiente plazas personas movilidad reducida (PMR)
<p>Descripción y objetivos:</p> <ul style="list-style-type: none"> • Extender el proyecto piloto existente a otras áreas de la ciudad mediante un sistema extremo a extremo que permita la sensorización de plazas de aparcamiento para personas con movilidad reducida en tiempo real mediante la identificación de vehículo a través de un App + etiqueta (Tag beacon). • Conectividad e integración de la información en la plataforma VLCI del Ayuntamiento. <p>Resultados cualitativos:</p> <ul style="list-style-type: none"> • Facilitar la movilidad de este colectivo mediante información en tiempo real de la disponibilidad de plazas con posibilidad de reserva. • Disminuir el fraude. • Disponibilidad de información del servicio en la plataforma VLCi para su explotación. <p>Grado de impacto:</p> <ul style="list-style-type: none"> • Incide en mejora de la eficiencia, innovación, inclusión social y automatización de métricas de la ciudad.
Gestión eficiente de los servicios de carga-descarga
<p>Descripción y objetivos:</p> <ul style="list-style-type: none"> • Extender el proyecto piloto existente de carga-descarga a otras áreas de la ciudad con un sistema extremo a extremo que permita sensorizar las plazas existentes de carga-descarga e integrarlas en los sistemas de información del Ayuntamiento. Valencia necesita optimizar este recurso dado que es deficitaria en zonas de carga-descarga y existe una dificultad objetiva para disminuir la ratio por comercio (de 10 a 8 comercios por vado) especialmente en las áreas más congestionadas. • Conectividad e integración de la información en la plataforma VLCI del Ayuntamiento. <p>Resultados cualitativos:</p> <ul style="list-style-type: none"> • Mejora de los flujos de tráfico y la contaminación disminuyendo el tiempo de reparto. • App integrada en AppValencia para búsqueda y guiado de plazas libres para personal autorizado. Control y visualización en tiempo real de la ocupación de plazas. • App para empresas del sector • Disponibilidad de información del servicio en la plataforma VLCi para su explotación. <p>Grado de impacto:</p> <ul style="list-style-type: none"> • Incide en mejora de la eficiencia, datos abiertos, innovación, reutilización y automatización de métricas en la ciudad.
Gestión eficiente paradas de Taxi
<p>Descripción y objetivos:</p> <ul style="list-style-type: none"> • Facilitar al ciudadano información en tiempo real de la disponibilidad de taxi (2 vehículos) en paradas de alto impacto para el turismo o negocios en la ciudad. <p>Resultados cualitativos:</p> <ul style="list-style-type: none"> • Facilitar la movilidad de este colectivo mediante información en tiempo real de la disponibilidad de plazas con posibilidad de reserva. • Disminuir el fraude. • Disponibilidad de información del servicio en la plataforma VLCi para su explotación. • Incide en mejora de la eficiencia, innovación, inclusión social y automatización de métricas de la ciudad.

5.2.1 Proyecto Gestión de Residuos Urbanos

El Ayuntamiento de Valencia dispone en total de 5.154 contenedores de residuos no perecederos de los siguientes tipos:

Residuos sólidos inorgánicos:

- Cartón: 1.783 contenedores.
- Envases: 1.599 contenedores.
- Vidrio: 1.663 contenedores.

Aceite:

- 109 contenedores

Con la implantación de un sistema de gestión inteligente de residuos urbanos se persigue una mejora en la operación del servicio basada en:

- Rutas óptimas de recogida en función del nivel de llenado y optimización de recursos.
- Disminución del impacto medioambiental: detección del desbordamiento de contenedores y optimización de la frecuencia de recogida.
- Disponibilidad de información del servicio en la plataforma VLCi para su explotación.

El objetivo de este proyecto es dotar de una solución integral de recogida selectiva para 237 contenedores de residuos sólidos inorgánicos en los distritos 17 (Pobles del Sud) y 19 (Pobles del Nord):

- 127 contenedores de vidrio geográficamente dispersos en los distritos de Pobles del Sud (100 en distrito 19) y Pobles del Nord (27 en distrito 17).
- 110 contenedores de envases ligeros geográficamente dispersos en los distritos de Pobles del Sud (87 en distrito 19) y Pobles del Nord (23 en distrito 17).

La actuación, contempla el suministro e instalación de 237 sensores para gestión de residuos urbanos, la plataforma de gestión y administración de dichos sensores (ASV de residuos urbanos) y la conectividad entre los sensores, plataforma de gestión y plataforma VLCi.

Las actuaciones se realizarán sobre los siguientes tipos de contenedores:

- Contenedores de vidrio tipo iglú de descarga inferior de 3000 litros y doble gancho operados mediante camión de 2 ejes y 18 toneladas de MMA dotado de grúa-pluma (ligera).
- Contenedores de envases ligeros y carga lateral de 3.200 litros gestionados mediante camiones de 3 ejes (MMA 26 Tm).

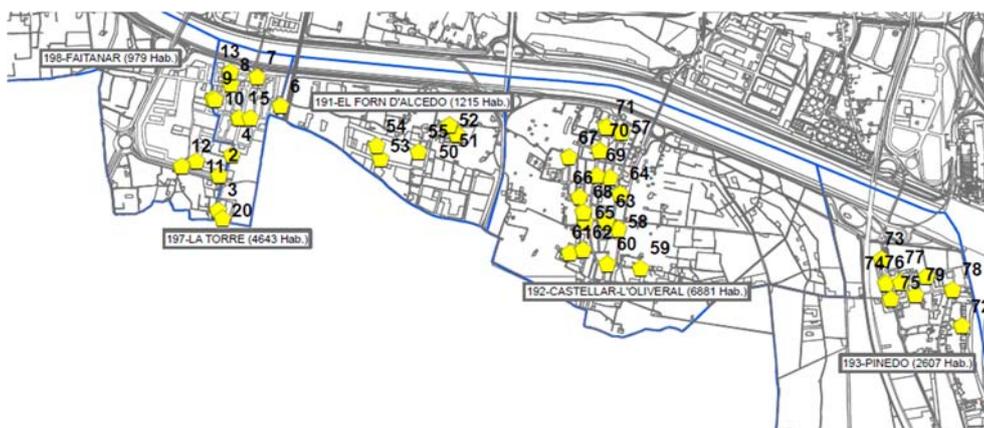


Figura 5.2 Plano parcial con ubicación de los contenedores

La información de cada sensor transmite a la plataforma se estima que será la siguiente:

- Última medición de cada sensor volumétrico
- Histórico llenado contenedor con volumétrico
- Porcentaje de llenado contenedor.
- Temperatura.
- Estado de las baterías.
- Cobertura de comunicación vía radio.
- Datos del contenedor: código tag, tipo
- Dirección ubicación del contenedor.

5.2.2 *Arquitectura de la solución*

En la Figura 5.3 se muestra una hipotética arquitectura de comunicaciones propuesta en esta tesis para este proyecto, aunque puede no coincidir con la que finalmente implemente el ayuntamiento.

También de manera arbitraria se ha escogido el sensor de residuos del fabricante IoTsens (IoTSens Waste) que va equipado con un sensor de ultrasonidos para la recolección en tiempo real de datos sobre el nivel de llenado (%), temperatura y vibración del contenedor con el fin de optimizar la gestión y detectar posibles acciones vandálicas.

Otros fabricantes de características similares pueden ser TSwasTe del fabricante TST y U-DUMP del fabricante Urbíotica.

Tabla 5.3 Características de conectividad del dispositivo IoTSENS

Comunicaciones	Características
GPRS	Cuatribanda (850/900/1800/1900Mhz) Protocolo TCP / UDP
LoRa	434 / 470Mhz 868/915 MHz, modos de enlace OTAA / ABP
Wifi	2.4GHz, estándar IEEE 802.11 b / g / n, seguridad WPA / WPA2
Sigfox	868/915 MHz, Zonas: RCZ1/RCZ2/RCZ4
Frecuencia envío	Cada dos horas (configurable)
Protocolos soportados	MQTT, API REST sobre HTTP (JSON, XML)

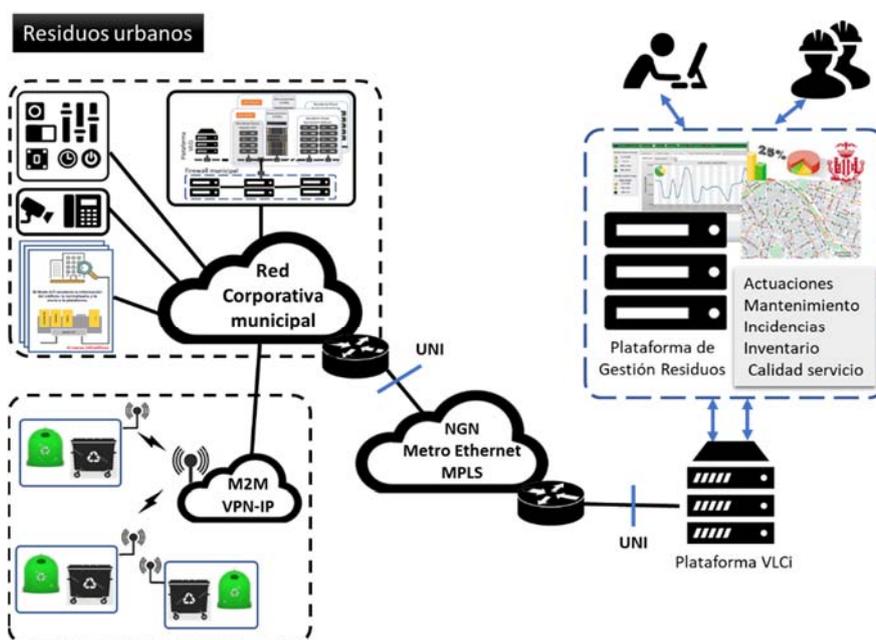


Figura 5.3 Arquitectura solución gestión inteligente de residuos urbanos

En este caso, y por los escasos requerimientos de la solución en cuanto a ancho de banda y calidad de servicio necesaria se ha considerado la utilización de accesos VPN-IP M2M contratados a un operador de red celular con un APN privado para garantizar la seguridad de la solución. La tecnología de acceso podrá ser GPRS, 3G, 4G o NB-IoT en función del tipo de sensor y disponibilidad de red.

Soluciones LoRa o Sigfox podrían también ser consideradas en el caso de existencia de cobertura en estas zonas periféricas de la ciudad.

La conectividad entre la red del operador a la red corporativa municipal se realizará a través de un acceso del servicio VPN de operador. Por ejemplo, si se utilizase el servicio VPN-IP de Telefónica, el acceso estará basado en Metro-ethernet o FTTH. Este tráfico

se encaminará hacia los módulos IoT Agents [120] de la plataforma de ciudad a través del acceso Metro Ethernet correspondiente.

Los protocolos más utilizados en los IoT Agents de la plataforma de ciudad son Ultralight 2.0 (SensorML) / JSON y HTTP / MQTT como protocolo de transporte, siendo este último el preferido por su mayor presencia en el mercado y prestaciones. MQTT (MQ Telemetry Transport) es un protocolo open source simple y muy ligero para la transmisión de mensajes cortos de telemetría y de control, desde/hacia una red de sensores/actuadores, que tenga limitaciones evidentes en cuanto al consumo, velocidad de transmisión y procesamiento. La estructura de la trama de datos junto con la carga (payload) tiene la siguiente estructura [121]:

- **i (device ID):** Device ID (unique for the API Key).
- **k (API Key):** API Key for the service the device is registered on.
- **t (timestamp):** Timestamp of the measure. Will override the automatic IoTAgent timestamp (optional).
- **d (Data):** Ultralight 2.0 payload.

5.2.2.1 Simulación del escenario con SimulCity

Con el fin de evaluar el tráfico generado por los contenedores de este proyecto hacia la Plataforma de Ciudad se calculará el ancho de banda, pérdida de paquetes, retardo y Jitter desde el interfaz de la red corporativa (UNI) con la Plataforma que reside en un Data Center. Sin pérdida de generalidad, SimulCity podría utilizarse para la evaluación del acceso VPN M2M del operador a la red corporativa municipal. La configuración es ilustrativa y se han incluido las siguientes condiciones de contorno.

- Número de contenedores = 237.
- Cada sensor enviará información de su estado cada dos horas.
- El inicio de transmisión de cada sensor seguirá una variable aleatoria con distribución uniforme en un intervalo de dos horas. De esta manera se maximiza la distribución del tráfico y se evita la posible aparición de picos por acumulación de activación simultánea de fuentes.
- No existen otras fuentes de tráfico en el interfaz.
- El caudal seleccionado es “Plata” con un CIR = 1Mb/s. (mínimo contratable).
- El protocolo de transporte seleccionado es UDP.
- El tamaño de la fuente se ha definido como una sola trama de longitud aleatoria que sigue una distribución exponencial acotada entre 120 y 210 bytes. Este valor se ha calculado en base a la suma de la información de contexto más las cabeceras de la pila de protocolos N2, N3, N4 y MQTT. Se ha utilizado una distribución exponencial para disminuir la probabilidad de generar un número elevado de paquetes grandes. La configuración de la fuente en SimulCity se realiza en cinco pasos:

1. Identificación del Servidor Destino.
 - a. `**FuenteDatos.H[0].udpApp[*].destAddresses = "Cloud.ServerDatos.S[0]"`
2. Distribución de inicio de activación de la fuente.
 - a. `**FuenteDatos.H[0].udpApp[*].startTime = uniform(0s,7200s)`
3. Calidad de servicio asignada
 - a. `**Datos.*[0].udpApp[*].*Port = 1003 #Tráfico Oro (1000 = multimedia, 1001 = oro, 1002 = critico, 1003 = plata)`
4. Tamaño de la fuente de datos.
 - a. `**FuenteDatos.H[0].udpApp[*].data_size = xmldoc("DataSize.xml", "///data-size[@id='sensor_waste']")`
5. Perfil de comportamiento temporal.
 - a. `**FuenteDatos.H[0].udpApp[*].config = xmldoc("ConfigTraffic.xml", "///data-profile[@id='1x7200']")`

El tamaño de la fuente de datos se define a través de un fichero de perfil XML (datasize). La combinación de los parámetros “textResourceSize”, “imageResourceSize”, “numResources” y “numImages” permiten construir estructuras de datos flexibles.

El comportamiento de la fuente se define a través del perfil XML (data-profile). La combinación de los parámetros “numPackets”, “SendInterval”, “numBursts” y “BurstInterval” permite construir distribuciones temporales flexibles. En este caso, se envía un solo bloque de datos cada dos horas (7200 segundos).

La Figura 5.4 muestra la configuración utilizada a través del perfil XML y en la Figura 5.5 la configuración de los parámetros de caudal en el interfaz Metro Ethernet.

```
<data-size id="sensor_waste">
  <textResourceSize type='exponential' mean='160' min='120' max='210' />
  <imageResourceSize type='exponential' mean='0' min='0' max='0' />
  <numResources type='constant' value='1' />
  <numImages type='uniform' beginning='0' end='0' />
</data-size>

<data-profile id="1x7200">
  <numPackets type="constant" value="1" />
  <SendInterval type="constant" value="1" />
  <numBursts type="constant" value="1" />
  <BurstInterval type="constant" value="7200" />
</data-profile>
```

Figura 5.4 Configuración de perfiles utilizados en SimulCity

```

**.Site[0].RouterHosts.**.ingressTC.meterPlata.cir = "1Mbps"
**.Site[0].RouterHosts.**.ingressTC.meterPlata.cbs = 0.25MiB
**.Site[0].RouterHosts.**.ingressTC.meterPlata.pir = "2Mbps"
**.Site[0].RouterHosts.**.ingressTC.meterPlata.pbs = 0.5MiB
    
```

Figura 5.5 Configuración de caudal en el interfaz Metro Ethernet

Con estos parámetros, y una vez ejecutada la simulación, SimulCity proporciona información a nivel de capa de transporte para cada fuente, y a nivel de interfaz agregado, funcionamiento de los mecanismos de calidad de servicio, longitud de las colas de los routers, etc. La información se agrupa en tres categorías:

- Vectores que almacenan el comportamiento de las fuentes a lo largo de la simulación como, por ejemplo, el ancho de banda a lo largo del tiempo consumido por una fuente y el agregado, etc. (436 vectores).
- Escalares que recogen parámetros agregados de la simulación como, por ejemplo, el número de paquetes enviados por fuente, la longitud media de las colas, los paquetes por clase de servicio, etc. (6009 escalares).
- Histogramas que recogen la distribución de determinadas variables vectoriales como, por ejemplo, la distribución de la longitud del paquete de cada fuente o la distribución del retardo a lo largo de la simulación.

La Figura 5.6 y la Figura 5.7 ilustran la información generada por esta herramienta.

Red.Cloud.RouterServ.eth[2].outputHook[0]	
> avg packets/s (scalar)	0.0393055555555556
> avg throughput (kbit/s) (scalar)	0.0509523333333333
> duration (scalar)	72000.0
> packet/sec (vector)	0.03930610147363158 (71999)
> thrupt (kbit/sec) (vector)	0.050953041014458515 (71999)
> total bits (scalar)	3668568.0
> total packets (scalar)	2830.0

Figura 5.6 Resultados de la simulación (entrada Plataforma Ciudad)

▼ Red.Solid wasteFuenteDatos.S.eth[1].mac	
> backoff:count (scalar)	0.0
> backoffs (scalar)	0.0
> bits/sec rcvd (scalar)	3.271111111111111
> bits/sec sent (scalar)	57.53233333333333
> collision:count (scalar)	0.0
> collisions (scalar)	0.0
> droppedPkBitError:count (scalar)	0.0
> droppedPkBitError:sum(packetBytes) (scalar)	0.0
> droppedPkfaceDown:count (scalar)	0.0
> droppedPkfaceDown:sum(packetBytes) (scalar)	0.0
> droppedPkNotForUs:count (scalar)	0.0
> droppedPkNotForUs:sum(packetBytes) (scalar)	0.0
> frames/sec rcvd (scalar)	0.00638888888888889
> frames/sec sent (scalar)	0.03930555555555556
> full-duplex (scalar)	1.0
> passedUpPk:count (scalar)	460.0
> passedUpPk:sum(packetBytes) (scalar)	29440.0
> passedUpPk:vector(packetBytes) (vector)	64.0 (460)
> rx channel collision (%) (scalar)	0.0
> rx channel idle (%) (scalar)	99.9999999632
> rx channel utilization (%) (scalar)	3.68E-8
> rxPausePkUnits:count (scalar)	0.0
> rxPausePkUnits:sum (scalar)	0.0
> rxPkFromHL:count (scalar)	2830.0
> rxPkFromHL:sum(packetBytes) (scalar)	517791.0
> rxPkFromHL:vector(packetBytes) (vector)	182.96501766784453 (2830)
> rxPkOk:count (scalar)	460.0
> rxPkOk:sum(packetBytes) (scalar)	29440.0
> rxPkOk:vector(packetBytes) (vector)	64.0 (460)
> simulated time (scalar)	72000.0
> txPausePkUnits:count (scalar)	0.0
> txPausePkUnits:sum (scalar)	0.0
> txPk:count (scalar)	2830.0
> txPk:sum(packetBytes) (scalar)	517791.0
> txPk:vector(packetBytes) (vector)	182.96501766784453 (2830)
> RedSolid waste.FuenteDatos.S.relayUnit	

Figura 5.7 Resultados de la simulación (fuentes de tráfico)

Una vez ejecutada la simulación para un periodo de 20 horas (arbitrario), el ancho de banda consumido por las fuentes se muestra en la Figura 5.8 donde se puede observar una distribución casi homogénea del tráfico y un caudal prácticamente despreciable, y que en ningún caso supera los 4 Kb/s. Como se apuntó en párrafos anteriores, la distribución homogénea del tráfico tiene como objetivo evitar la concurrencia de múltiples fuentes en el mismo instante de tiempo.

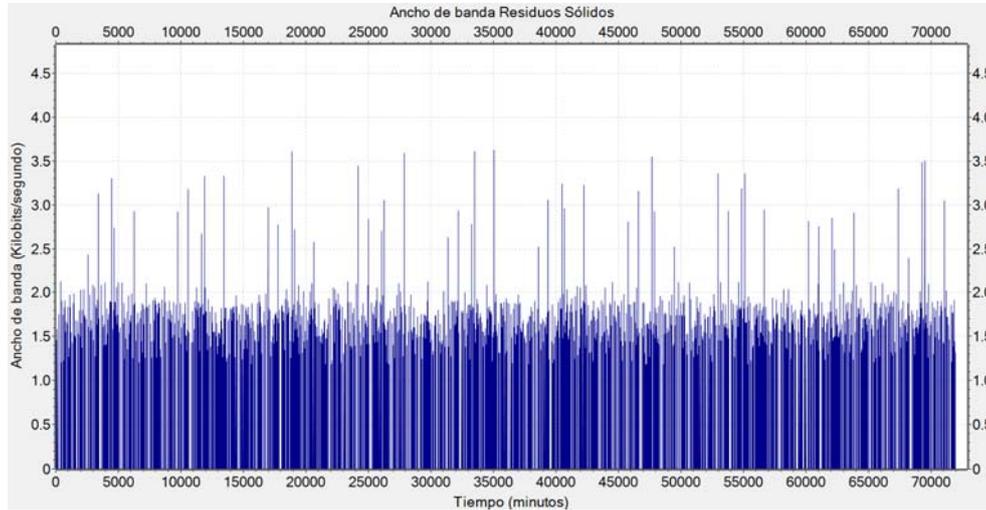


Figura 5.8 Ancho de banda consumido por los sensores de residuos

En la Figura 5.9 se muestra el retardo desde el interfaz UNI (entrada del router del servicio de operadora) y la plataforma de ciudad. Como en el caso del caudal, el retardo es prácticamente despreciable dado que es el único servicio disponible en el interfaz.

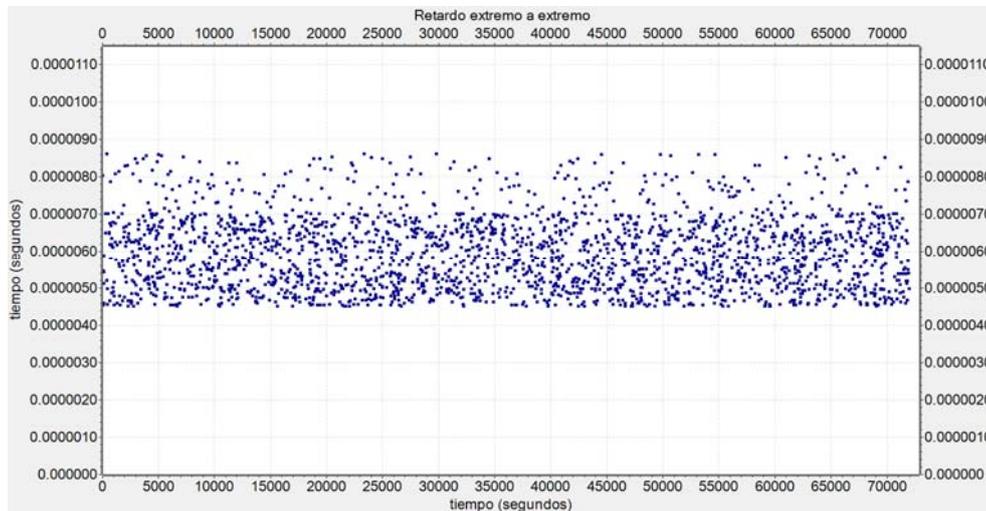


Figura 5.9 Retardo interfaz UNI Metro Ethernet - Plataforma de Ciudad

Como era de esperar, no existe pérdida de paquetes como se observa en la Figura 5.10

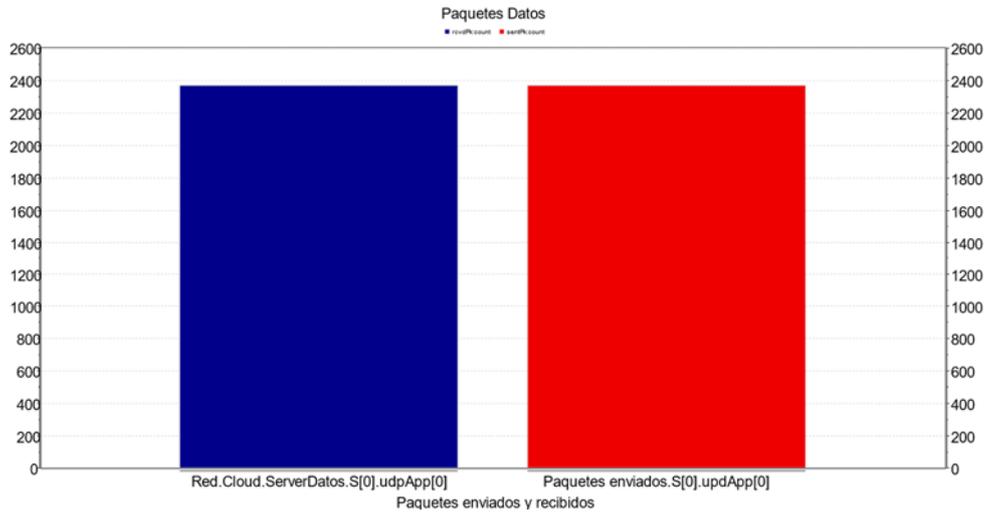


Figura 5.10 Paquetes enviados / recibidos

Respecto al Jitter, el resultado es el esperado (Figura 5.20) dado que al no actuar los mecanismos de control de flujo ni colas en los routers extremos solamente aparece el generado por el módulo de simulación de la red MPLS.

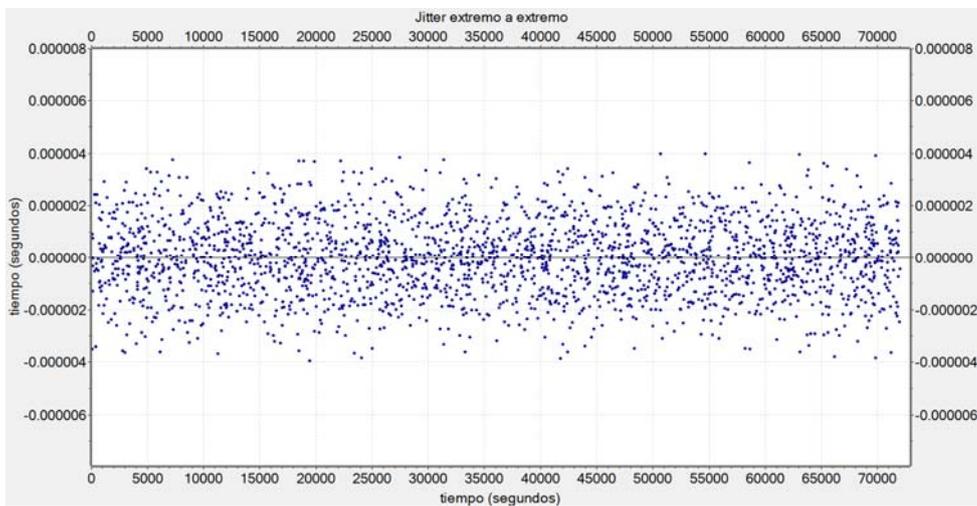


Figura 5.11 Jitter extremo a extremo

En la Figura 5.12 y como complemento de la figura anterior, se muestra la distribución del número de paquetes recibidos y retardo confirmándose de nuevo la ausencia de

cualquier problema de congestión. Este histograma es de interés en el diseño de la conectividad en situaciones de congestión porque muestra el uso de las colas de los routers por servicio.



Figura 5.12 Histograma número de paquetes - retardo

Como conclusiones de este proyecto, se ha podido comprobar el escaso impacto que este tipo de actuaciones imponen sobre la red de comunicaciones municipal siempre que se realice una adecuada configuración de los dispositivos involucrados. Aunque los parámetros de configuración se han escogido de manera arbitraria, la flexibilidad de la herramienta permite evaluar el impacto de posibles modificaciones en los parámetros de configuración, por ejemplo, en período invernal, lecturas cada 4 u 8 horas podrían ser suficientes por la baja tasa de llenado de los contenedores.

5.2.3 Proyecto Gestión Alumbrado Público

El sistema de alumbrado público de Valencia está constituido por un total de 744 Centros de Mando en vía pública, que gestionan más de 107.000 luminarias.

El objetivo de este proyecto es dotar de inteligencia a determinados centros de mando y luminarias y proporcionar un sistema de gestión de alumbrado que permita al Ayuntamiento de Valencia gestionar adecuadamente el servicio, especialmente en las pedanías que de él dependen, minimizando los tiempos de resolución de averías y los robos de cobre.

El fallo simultáneo de un número significativo de luminarias, junto con su posición en la línea de alumbrado (si particularmente afecta a las últimas) puede señalar, con un elevado grado de certeza, un intento de robo de cable de cobre, que se detectará a través de los correspondientes controladores de luminaria (SLC).

Esta solución permitirá conocer en tiempo real los parámetros eléctricos de las instalaciones tele gestionadas para su análisis, el funcionamiento a nivel de luminaria, centro de mando y realizar las operaciones de gestión típicas del servicio (activación, desactivación, teledatada, regulación lumínica, así como la detección de averías, sabotajes y robos en tiempo real).

La actuación, contempla el suministro e instalación de 60 controladores de Centro de Mando y 2.000 controladores de luminaria individual.

Las funcionalidades de la solución permitirán, entre otras:

- Encendido, apagado remoto para todo tipo de luminarias.
- Regulación independiente de la intensidad de cada punto de luz.
- Instalación de sensores que permiten adaptar de forma automática la iluminación pública a las condiciones meteorológicas, de tráfico, viandantes, etc.
- Reducción de costes en consumos y tareas de mantenimiento mediante la gestión de alarmas en tiempo real.
- Detección de incidencias en la línea (robo de cobre).

5.2.3.1 *Arquitectura de la solución*

En la Figura 5.13 se muestra una hipotética arquitectura de comunicaciones propuesta en esta tesis para este proyecto, aunque puede no coincidir con la que finalmente implemente el ayuntamiento

En general, la arquitectura de la solución propuesta se basa en una red a dos niveles (outdoor lighting network). Cada Centro de Mando dispone de un “Concentrador de Segmento” que es el encargado de gestionar las luminarias que de él dependen, aproximadamente 150 por término medio, y transferir esta información a la Plataforma de la Ciudad.

La conexión entre el Concentrador de Segmento y las luminarias se realiza a través de una arquitectura WSN, típicamente basada en los estándares 802.15.4 o PLC. En cada luminaria se instala un dispositivo denominado “controlador de luminaria” encargado de gestionar los parámetros de iluminación, consumo de energía y envío de alarmas.

Para la conexión del Centro de Mando con la Plataforma de Ciudad se ha considerado la utilización de accesos VPN-IP M2M contratados a un operador de red celular con un APN privado para garantizar la seguridad de la solución. La tecnología de acceso será 3G, 4G por razones de coste.

La conectividad entre la red del operador a la red corporativa municipal se realizará a través de un acceso del servicio VPN de operador como en el proyecto anterior. Por ejemplo, si se utilizase el servicio VPN-IP de Telefónica, el acceso estará basado en Metro-ethernet o FTTH. Este tráfico se encaminará hacia la plataforma de ciudad a través del acceso Metro Ethernet correspondiente.

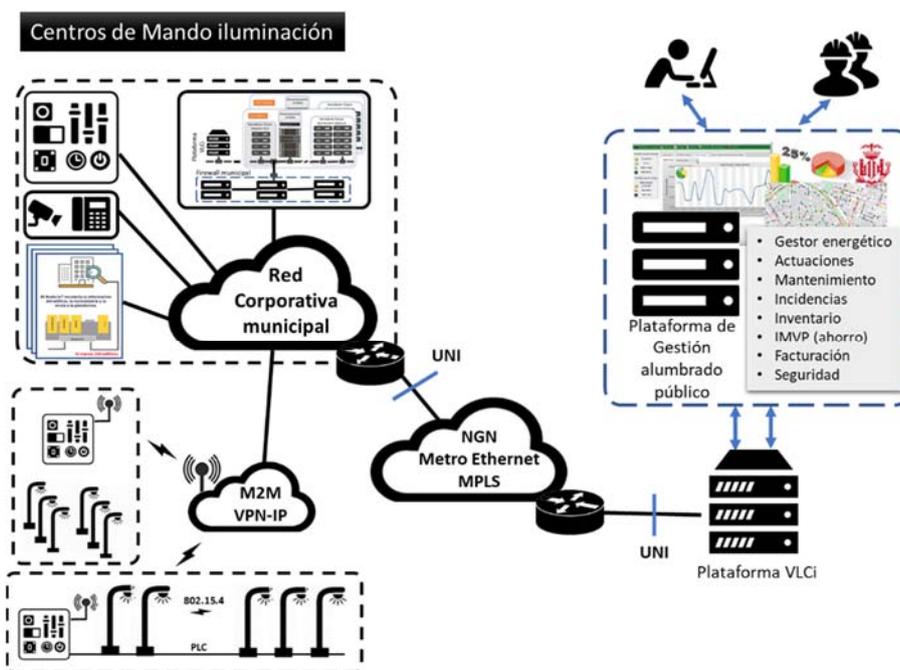


Figura 5.13 Arquitectura solución alumbrado

También, de manera arbitraria se ha escogido un controlador de luminaria que permite además de gestionar la propia luminaria, conectar dispositivos externos, como sensores medioambientales o cámaras de videovigilancia. En el caso de que se conecte un sistema de videovigilancia, la conectividad entre el centro de mando y la red corporativa municipal se realizará conectando el Centro de Mando a la red corporativa municipal, bien por una fibra óptica directa utilizando la infraestructura semafórica o de alumbrado o bien mediante una VPN cableada contratada al operador.

En la Tabla 5.4 se muestran los parámetros de comunicaciones de los posibles dispositivos controladores de luminaria y de segmento.

Tabla 5.4 Características de conectividad del dispositivo Controlador de Luminaria

Comunicaciones	Características
PLC*	Propietario
IEE 802.15.4*	Red mallada ZigBee Pro, 2.4 GHz.
3G, 4G	Cuatribanda (850/900/1800/1900Mhz) Protocolo TCP / UDP
Frecuencia envío	Cada dos segundos (configurable)
Protocolos soportados	MQTT, API REST sobre HTTP (JSON, XML)

*Excluyentes

5.2.3.2 Simulación del escenario con SimulCity

Con el fin de evaluar el tráfico generado por los Centros de Mando de este proyecto hacia la Plataforma de Ciudad, se evaluará el ancho de banda, pérdida de paquetes, retardo y Jitter desde el interfaz de la red corporativa (UNI) con la Plataforma que reside en un Data Center. Sin pérdida de generalidad, SimulCity podría utilizarse para la evaluación del acceso VPN M2M del operador a la red corporativa municipal.

Dado que en la actualidad existen otros dos proyectos en marcha de renovación de luminarias y telecontrol se ha optado por simular la red a dimensionado final, es decir, 744 Centros de Mando y 107.000 luminarias.

- Número de centros de mando: 744.
- El número de luminarias por centro de mando se modela como una variable aleatoria de distribución uniforme entre 100 y 200 que se corresponde con la planta actualmente existente.
- Cada sensor enviará información de su estado cada diez segundos al controlador. Éste consolida los diferentes contextos para minimizar el consumo de ancho de banda y genera una trama al interfaz IoT Agent de la Plataforma.
- El inicio de transmisión de cada centro de mando seguirá una variable aleatoria con distribución uniforme en un intervalo de diez segundos. De esta manera se maximiza la distribución del tráfico.
- No existen otras fuentes de tráfico en el interfaz.
- El caudal seleccionado es “Plata” con un CIR = 2 Mb/s.
- El tiempo de simulación se programará para 30 minutos.

La configuración de la fuente en SimulCity se realiza en cinco pasos:

1. Identificación del Servidor Destino.
 - a. `**FuenteDatos.H[0].udpApp[*].destAddresses = "Cloud.ServerDatos.S[0]"`
2. Distribución de inicio de activación de la fuente.
 - a. `**FuenteDatos.H[0].udpApp[*].startTime = uniform(0s,10s)`
3. Calidad de servicio asignada
 - a. `**.*Datos.*[0].udpApp[*].*Port = 1003 #Tráfico Oro (1000 = multimedia, 1001 = oro, 1002 = crítico, 1003 = plata)`
4. Tamaño de la fuente de datos.
 - a. `**FuenteDatos.H[0].udpApp[*].data_size = xmldoc("DataSize.xml", "///data-size[@id='seco_lum']")`
5. Perfil de comportamiento temporal.
 - a. `**FuenteDatos.H[0].udpApp[*].config = xmldoc("ConfigTraffic.xml", "///data-profile[@id='seco']")`

El tamaño de la fuente de datos se define a través de la definición de un perfil XML (datasize) como en casos anteriores. En este caso, es necesario simular el número variable de luminarias que dependen de cada centro de mando. Para ello el tamaño de la trama

“texResourceSize” se ha definido como una variable aleatoria de distribución uniforme de longitud comprendida entre 6.000 y 12.000 bytes. Este tamaño se corresponde con la estructura de carga, contexto de datos de los sensores dependientes del centro de mando.

El comportamiento de la fuente se define a través del perfil XML (data-profile). En este caso, se envía un solo bloque de datos cada 10 segundos.

La Figura 5.14 muestra la configuración utilizada ya descrita en párrafos anteriores y en la Figura 5.15 la configuración de caudales en el interfaz que define un caudal plata garantizado (CIR) de 5Mbs y un caudal de pico (PIR) de 10 Mbs.

```
<data-size id="seco_lum">
  <textResourceSize type='uniform' beginning='6000' end='12000' />
  <numResources type="constant" value="1" />
  <imageResourceSize type="constant" value="1" />
  <numImages type='uniform' beginning='0' end='0' />
</data-size>

</data-size><data-profile id="seco">
  <numPackets type="constant" value="1"/>
  <SendInterval type="constant" value="1" />
  <numBursts type="constant" value="1" />
  <BurstInterval type="constant" value="10" />
</data-profile>
```

Figura 5.14 Configuración de perfiles utilizados en SimulCity para c. luminarias

```
**Site[0].RouterHosts.**.ingressTC.meterPlata.cir = "2Mbps"
**Site[0].RouterHosts.**.ingressTC.meterPlata.cbs = "0.25MiB"
**Site[0].RouterHosts.**.ingressTC.meterPlata.pir = "4Mbps"
**Site[0].RouterHosts.**.ingressTC.meterPlata.pbs = "0.5MiB"
```

Figura 5.15 Configuración de caudal en el interfaz Metro Ethernet

Una vez ejecutada la simulación para un periodo de 30 minutos, el ancho de banda consumido por las fuentes se muestra en la Figura 5.16 donde se puede observar un caudal elevado con picos que llegan a los 6,7 Mb/s. y una tasa media de 5,44 Mb/s.

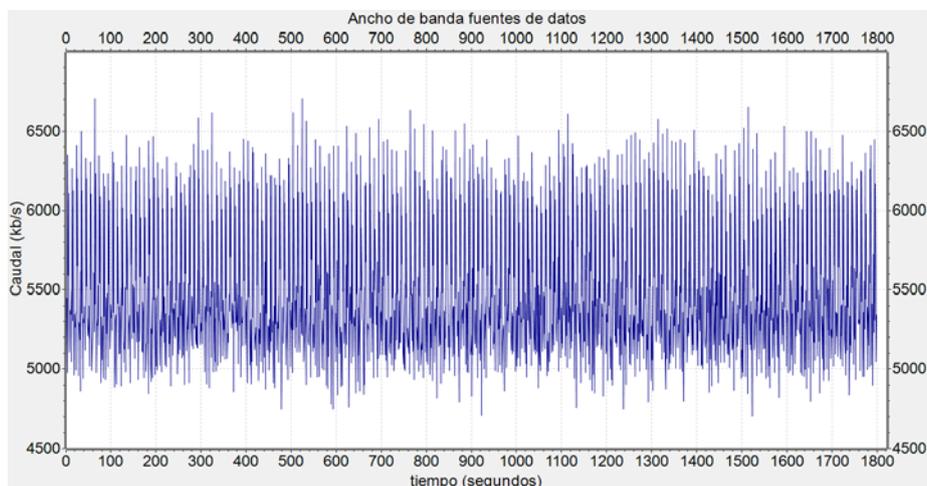


Figura 5.16 Ancho de banda Centros de Mando Luminarias

En el interfaz de salida, el tráfico se limita a 4Mb/s. (EIR) por la actuación de los mecanismos de gestión de calidad de servicio. Todo el tráfico superior al EIR se descarta. Analizando la tasa de paquetes enviados y recibidos se observa que existe una pérdida de paquetes de un 17,39%. Una buena práctica consiste en contratar caudal plata sobre la media del tráfico cursado, aunque la red puede descartar paquetes en cualquier momento. En la Figura 5.17 se detalla la nueva configuración de caudales seleccionada tomando en base esta premisa.

```
**Site[0].RouterHosts.**.ingressTC.meterPlata.cir = "6Mbps"  
**Site[0].RouterHosts.**.ingressTC.meterPlata.cbs = "0.75MiB"  
**Site[0].RouterHosts.**.ingressTC.meterPlata.pir = "12Mbps"  
**Site[0].RouterHosts.**.ingressTC.meterPlata.pbs = "1.5MiB"
```

Figura 5.17 Configuración de caudal en el interfaz Metro Ethernet

En la Figura 5.18 se muestra el retardo desde el interfaz UNI (entrada del router del servicio de operadora) y la plataforma de ciudad. Es importante destacar el impacto en el retardo producido por las colas en el router de salida.

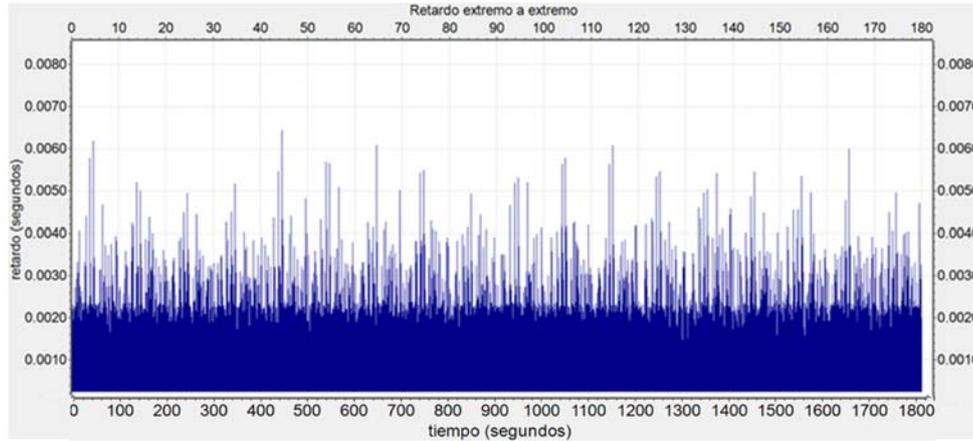


Figura 5.18 Retardo interfaz UNI Metro Ethernet - Plataforma de Ciudad

Con esta configuración no existe pérdida de paquetes como se observa en la Figura 5.19



Figura 5.19 Paquetes enviados / recibidos

Respecto al Jitter, el resultado de la Figura 5.20 muestra el impacto de los mecanismos de control de flujo y colas en el interfaz UNI del acceso Metro Ethernet. En este caso el Jitter de la red MPLS es despreciable.

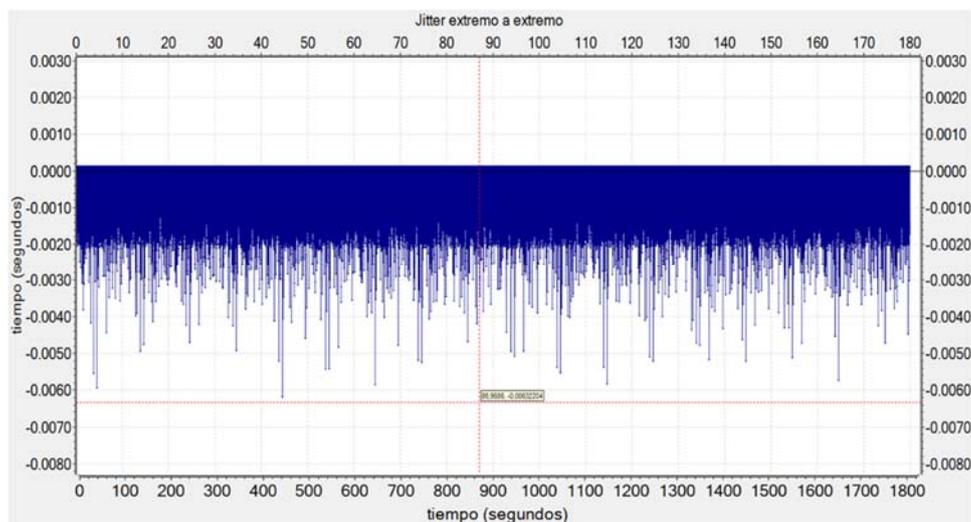


Figura 5.20 Jitter extremo a extremo

Como conclusiones de este proyecto, se ha podido evaluar una solución IoT a dos niveles, una red WSN que permite la comunicación entre luminarias más una pasarela que consolida este tráfico y se conecta con la plataforma de ciudad.

En esta simulación se ha escogido un período de muestreo de diez segundos con el fin de poder identificar con rapidez el posible robo de cobre. Los sistemas comerciales suelen establecer el muestreo cada 30 segundos o superior para disminuir el consumo de datos en redes con limitaciones de volumen de tráfico, como ocurre en las VPN M2M

5.2.4 Proyecto Aparcamiento Regulado para Personas con Movilidad Reducida, Servicios de Carga y Descarga y Paradas de Taxis.

València cuenta actualmente con unas 1.600 plazas para personas de movilidad reducida (PMR), alrededor de 1.500 vados de carga y descarga y unas 225 paradas de taxi. El número de posibles usuarios PMR se estima en 7.000 y 3.000 los taxis en el área metropolitana, no se dispone de datos de número de vehículos de carga/descarga.

Esta actuación plantea la implantación de un sistema de gestión inteligente de plazas de aparcamiento de la ciudad, con el que se puedan conseguir los siguientes objetivos:

- Controlar el uso idóneo de las plazas de aparcamiento para personas con movilidad reducida (PMR), verificando si el usuario está registrado o no, enviando la información a la plataforma del servicio, para ser procesada y ofrecida a los ciudadanos.

- Controlar el uso de las plazas de aparcamiento de Carga y Descarga (CyD) detectando la ocupación de las plazas y el tiempo de ocupación de estas, con el fin de determinar posibles incumplimientos de la normativa.
- Facilitar al ciudadano información en tiempo real de la disponibilidad de taxis en paradas de taxi (PT) de alto impacto para el turismo o negocios en la ciudad.

El número de plazas a sensorizar recogido en la convocatoria se muestra a continuación:

- Plazas para personas con movilidad reducida: 425 plazas
- Plazas de Carga y Descarga: 125 plazas
- Paradas de Taxi: 126 plazas

Esta solución permitirá conocer en tiempo real el estado de cada plaza, el tiempo de permanencia y si ésta está ocupada por un usuario habilitado para ello o no (solamente en el caso de plazas PMR).

En la Figura 5.21 se muestra un plano con la ubicación de las plazas de aparcamiento a sensorizar por barrios, donde se puede observar que las plazas (puntos apenas visibles) están distribuidas por toda la ciudad.



Figura 5.21 Plano con ubicación de plazas PMR, CyD y paradas de taxi

Las funcionalidades de la solución permitirán, entre otras:

- Detectar la presencia física de un vehículo en cada plaza.
- Los tiempos de llegada y salida de los vehículos.

- Comprobación si el vehículo está registrado (sólo para plazas PMR).
- Período de envío de datos programable desde 1 segundo a 24 horas.

5.2.4.1 Arquitectura de la solución

En la Figura 5.22 se muestra una hipotética arquitectura de comunicaciones propuesta en esta tesis para este proyecto, aunque puede no coincidir con la que finalmente implemente el ayuntamiento.

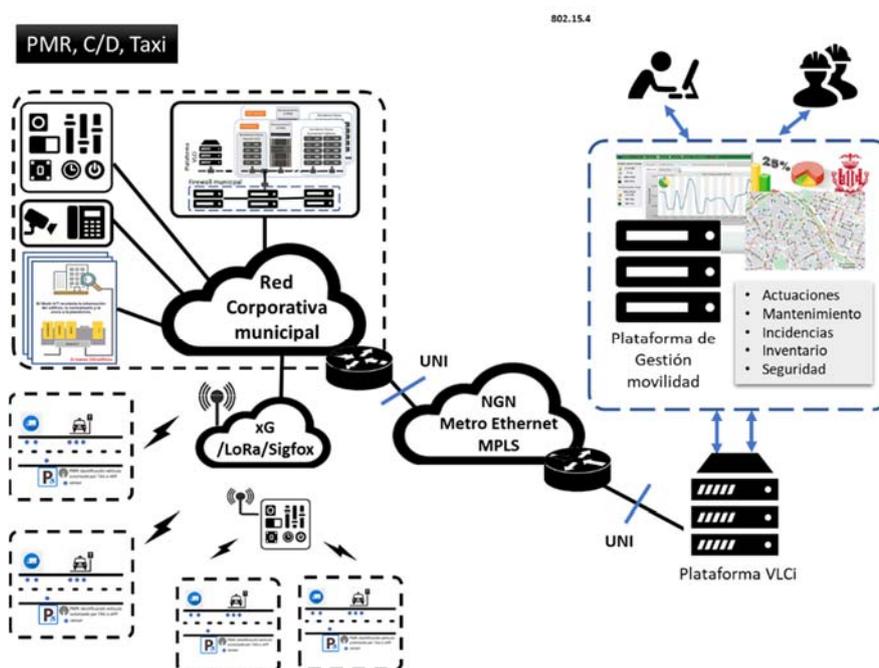


Figura 5.22 Arquitectura solución PMR, CyD y Taxi

La instalación de sensores de aparcamiento requiere en muchos casos la utilización de dispositivos con una elevada vida útil (típicamente 8 a 10 años), resistencia contra la intemperie y agresiones mecánicas dado que se instalan directamente sobre la calzada. Con estos condicionantes, las soluciones de comunicación han de gestionar eficientemente la energía y ofrecer cobertura adecuada por la atenuación producida al posicionarse el vehículo sobre el dispositivo.

Básicamente existen dos arquitecturas mayoritarias en el mercado para capturar la información de los sensores. La primera consta de sensores de parking que se conectan a un “Concentrador” que es el encargado de gestionar los sensores que de él dependen, y conectarse con la Plataforma de la Ciudad.

La conexión entre el Concentrador y los sensores de parking se realiza a través de una arquitectura WSN, típicamente basada en RF (433/868 MHz).

La conexión del concentrador con la Plataforma de Ciudad permite la utilización de accesos VPN-IP M2M contratados a un operador de red celular con un APN privado para garantizar la seguridad de la solución. La tecnología de acceso podrá ser 3G, 4G, Lora o SigFox.

La segunda solución consiste en la conexión directa del sensor con la VPN del operador utilizando Lora o SigFox. En un futuro está previsto que el mercado ofrezca además accesos NB-IoT

La conectividad entre la red del operador a la red corporativa municipal se realizará a través de un acceso del servicio VPN de operador como en descripciones previas. Por ejemplo, si se utilizase el servicio VPN-IP de Telefónica, el acceso estará basado en Metro-ethernet o FTTH. Este tráfico se encaminará hacia la plataforma de ciudad a través del acceso Metro Ethernet correspondiente.

De manera arbitraria se ha escogido un sensor de parking magnético del fabricante Libelium que no necesita nodo concentrador y cuyas características se describen en la Tabla 5.5.

Tabla 5.5 Características de conectividad del dispositivo de Libelium

Comunicaciones	Características
LoRaWAN	LoRaWAN radio for Europe: 863.0 to 870.0 MHz
Sigfox	Sigfox radio for Europe: 868.0 to 869.7 MHz
Modo día	Una medida cada minuto (configurable). Se activa el interfaz radio si se detecta cambio.
Trama keep - alive	Trama básica con indicación del estatus del sensor y batería.
Tamaño de la trama	11 bytes

5.2.4.2 Simulación del escenario con SimulCity

Con el fin de evaluar el tráfico generado por los sensores hacia la Plataforma de Ciudad, se calculará el ancho de banda, pérdida de paquetes, retardo y Jitter desde el interfaz de la red corporativa (UNI) con la Plataforma que reside en un Data Center. Sin pérdida de generalidad, SimulCity podría utilizarse para la evaluación del acceso VPN M2M del operador a la red corporativa municipal.

Dado que en la actualidad se están evaluando otros proyectos de sensorización, se ha optado por simular la red a dimensionado final, es decir, unas 1.600 plazas para personas de movilidad reducida (PMR), alrededor de 1.500 vados de carga y descarga y unas 225 paradas de taxi, total de 3.325 plazas.

- Cada sensor enviará información solamente en el momento en el que detecta un cambio de estado (vehículo entrando o saliendo en la plaza).

- Se definirán tres tipos de fuentes de tráfico para simular el comportamiento de usuarios PMR, CyD y taxi.
- No existen otras fuentes de tráfico en el interfaz.
- El caudal seleccionado es “Plata” con un CIR = 1Mb/s.

La configuración de las fuentes en SimulCity se realizará como en apartados anteriores, pero generando tres tipos diferentes de fuentes de datos que compartirán el interfaz. En este caso los apartados de configuración 1 a 3 serán comunes y la configuración de las fuentes de datos y su comportamiento serán propios de cada una:

1. Identificación del Servidor Destino i [0-2].
 - a. `**FuenteDatos.H[i].udpApp[*].destAddresses = "Cloud.ServerDatos.S[i]"`
2. Distribución de inicio de activación de la fuente.
 - a. `**FuenteDatos.H[i].udpApp[*].startTime = uniform(0s,1s)`
3. Calidad de servicio asignada
 - a. `**.*Datos.*[i].udpApp[*].*Port = 1003 #Tráfico Oro (1000 = multimedia, 1001 = oro, 1002 = crítico, 1003 = plata)`
4. Tamaño de la fuente de datos.
 - a. `**FuenteDatos.H[0].udpApp[*].data_size = xmldoc("DataSize.xml", "//data-size[@id='PMR']")`
 - b. `**FuenteDatos.H[1].udpApp[*].data_size = xmldoc("DataSize.xml", "//data-size[@id='CyD']")`
 - c. `**FuenteDatos.H[2].udpApp[*].data_size = xmldoc("DataSize.xml", "//data-size[@id='TAXI']")`
5. Perfil de comportamiento temporal.
 - a. `**FuenteDatos.H[0].udpApp[*].config = xmldoc("ConfigTraffic.xml", "//data-profile[@id='PMR']")`
 - b. `**FuenteDatos.H[1].udpApp[*].config = xmldoc("ConfigTraffic.xml", "//data-profile[@id='CyD']")`
 - c. `**FuenteDatos.H[2].udpApp[*].config = xmldoc("ConfigTraffic.xml", "//data-profile[@id='TAXI']")`

El tamaño de las fuentes se ha definido para las tres como una trama de longitud aleatoria que sigue una distribución exponencial acotada entre 120 y 210 bytes. Este valor se ha calculado en base a la suma de la información de contexto más las cabeceras de la pila de protocolos N2, N3, N4 y MQTT. Se ha utilizado una distribución exponencial para disminuir la probabilidad de generar un número elevado de paquetes grandes.

El comportamiento de cada fuente se define a través del perfil XML (data-profile) con valores arbitrarios.

- Para PMR se ha modelado el tiempo entre envío de paquetes como una distribución exponencial acotada, entre 6 minutos y 3 horas con media 1 hora.
- Para CyD se ha modelado el tiempo entre envío de paquetes como una distribución exponencial acotada, entre 5 y 50 minutos con media 20.
- Para TAXI se ha modelado el tiempo entre envío de paquetes como una distribución exponencial acotada, entre 5 y 30 minutos con media 10.

La Figura 5.23 muestra la configuración utilizada y en la Figura 5.24 la configuración de caudales en el interfaz.

```

<data-profile id="PMR">
  <numPackets type="constant" value="1"/>
  <SendInterval type="constant" value="1" />
  <numBursts type="constant" value="1" />
  <BurstInterval type='exponential' mean='3600' min='360' max='10800' />
</data-profile>
<data-profile id="CyD">
  <numPackets type="constant" value="1"/>
  <SendInterval type="constant" value="1" />
  <numBursts type="constant" value="1" />
  <BurstInterval type='exponential' mean='1200' min='300' max='3000' />
</data-profile>
<data-profile id="TAXI">
  <numPackets type="constant" value="1"/>
  <SendInterval type="constant" value="1" />
  <numBursts type="constant" value="1" />
  <BurstInterval type='exponential' mean='600' min='300' max='1800' />
</data-profile>

<data-size id="PMR">
  <textResourceSize type='exponential' mean='150' min='100' max='210' />
  <imageResourceSize type='exponential' mean='0' min='0' max='0' />
  <numResources type="constant" value="1" />
  <numImages type='uniform' beginning='0' end='0' />
</data-size>
<data-size id="CyD">
  <textResourceSize type='exponential' mean='150' min='100' max='210' />
  <imageResourceSize type='exponential' mean='0' min='0' max='0' />
  <numResources type="constant" value="1" />
  <numImages type='uniform' beginning='0' end='0' />
</data-size>
<data-size id="TAXI">
  <textResourceSize type='exponential' mean='150' min='100' max='210' />
  <imageResourceSize type='exponential' mean='0' min='0' max='0' />
  <numResources type="constant" value="1" />
  <numImages type='uniform' beginning='0' end='0' />
</data-size>

```

Figura 5.23 Configuración de perfiles utilizados en SimulCity para PMP, CyD, TAXI

```

**.Site[0].RouterHosts.**.ingressTC.meterPlata.cir = "1Mbps"
**.Site[0].RouterHosts.**.ingressTC.meterPlata.cbs = "0.125MiB"
**.Site[0].RouterHosts.**.ingressTC.meterPlata.pir = "2Mbps"
**.Site[0].RouterHosts.**.ingressTC.meterPlata.pbs = "0.25MiB"

```

Figura 5.24 Configuración de caudal en el interfaz Metro Ethernet

Una vez ejecutada la simulación para un periodo de diez horas, el ancho de banda consumido por las fuentes se muestra en la Figura 5.25.

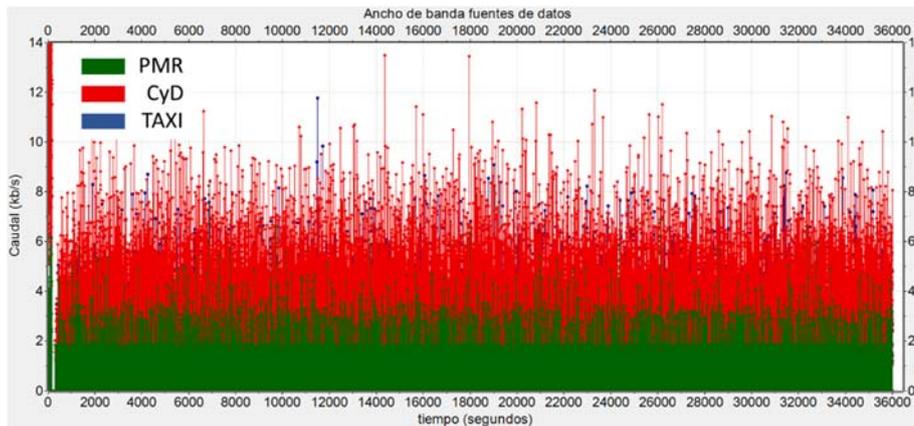


Figura 5.25 Ancho de banda PMP, CyD, TAXI

La Figura 5.26 representa el ancho de banda total donde se puede observar un caudal prácticamente despreciable como era de esperar.

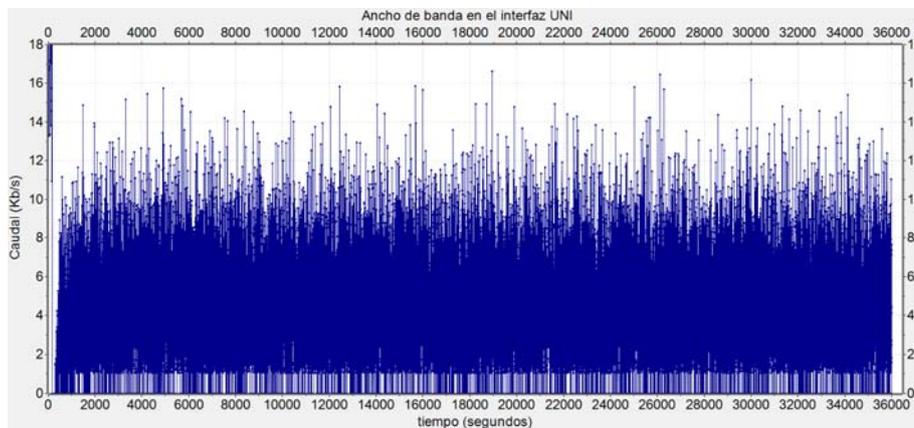


Figura 5.26 Ancho de banda interfaz UNI PMP, CyD, TAXI

En la Figura 5.27 se muestra el retardo desde el interfaz UNI (entrada del router del servicio de operadora) hasta la plataforma de ciudad. En este caso por el escaso caudal requerido, el retardo es despreciable.

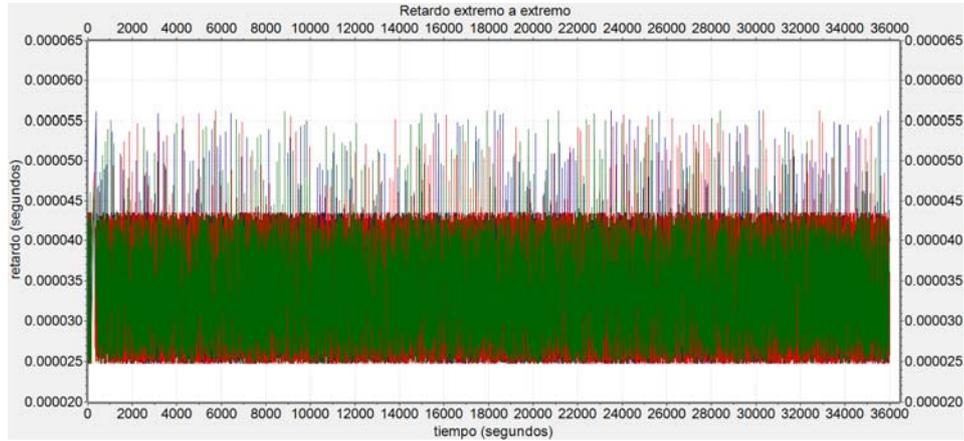


Figura 5.27 Retardo interfaz UNI Metro Ethernet - Plataforma de Ciudad

Con esta configuración no existe pérdida de paquetes como se observa en la Figura 5.28

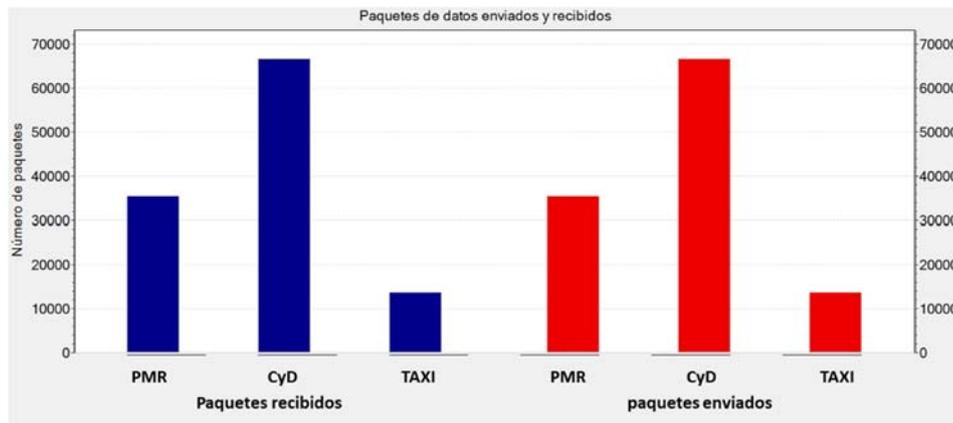


Figura 5.28 Paquetes enviados / recibidos

Respecto al Jitter de las fuentes de datos mostrado en la Figura 5.29, como se puede comprobar, también es despreciable.

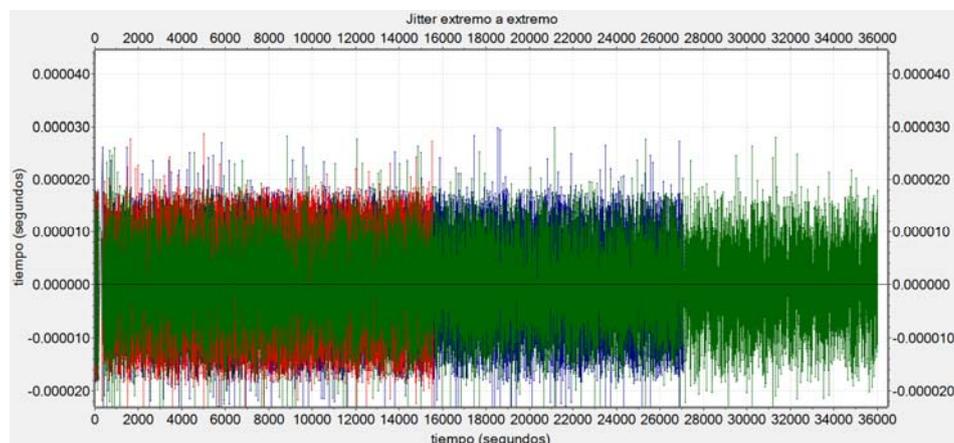


Figura 5.29 Jitter extremo a extremo

La conclusión de la simulación de este proyecto es que se ha podido comprobar el escaso impacto que este tipo de actuaciones imponen sobre la red de comunicaciones municipal siempre que se realice una adecuada configuración de los dispositivos involucrados. Aunque los parámetros de configuración se han escogido de manera arbitraria, la flexibilidad de la herramienta permite evaluar el impacto de posibles modificaciones, por ejemplo, disminuir drásticamente las lecturas para maximizar la vida de las baterías en plazas de escasa rotación como pueden ser las plazas PMR en barrios periféricos.

5.3 Actualización servicio de telefonía corporativa municipal

Como se apuntó en la introducción, y con el exclusivo propósito de mostrar las funcionalidades de la herramienta SimulCity, se realiza el ejercicio de migración de la actual infraestructura de comunicaciones de voz basada en centralitas TDM del fabricante Philips, cerca del final de su vida útil, a una nueva arquitectura de telefonía IP (ToIP) con servidores centralizados.

En principio, esta nueva solución permitirá la migración de aproximadamente 4.500 usuarios a este nuevo servicio de Telefonía IP, así como la migración del teléfono de atención ciudadana “010” que cuenta con 10 agentes.

De manera complementaria se implantará una solución de comunicaciones unificadas que incluye servicio de videoconferencia IP en calidad HD (1280x720) con una capacidad de hasta 20 usuarios simultáneos.

5.3.1.1 Arquitectura de la solución

En la Figura 5.30 se describe la arquitectura final de la solución diseñada en esta tesis donde los edificios municipales están conectados a la red corporativa municipal, bien

por enlaces directos de fibra o bien mediante una VPN-IP del operador en el que se definirá una VLAN expresa para ToIP con caudal de tipo “multimedia”.

La conexión con red pública, (accesos primarios virtuales) como se observa en la figura, se realizará desde el servidor de telefonía a la red NGN del operador, lo que permitirá eliminar los accesos primarios RDSI actualmente instalados en dependencias del ayuntamiento con el ahorro de costes correspondiente.

Para la telefonía IP se ha seleccionado un códec G.711 que es más utilizado en los servidores de ToIP con mayor cuota de mercado [122], fundamentalmente por razones de coste.

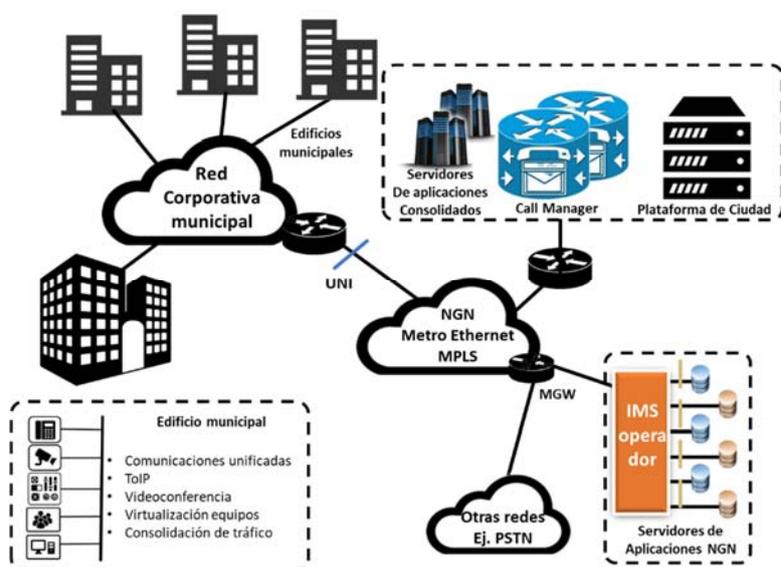


Figura 5.30 Arquitectura municipal consolidada

Respecto a la selección de la intensidad de tráfico para los servicios de voz se ha utilizado la información de la Tabla 5.6 de la Recomendación ITU-T Q. 543 [72]. Concretamente se ha seleccionado la Clase de flujo de tráfico I para la telefonía corporativa.

Tabla 5.6 Características del tráfico de voz

Clases de flujo de tráfico	Tipo de distribución	Intensidad de tráfico (Erlang)	Número medio de BHCA
1	Poisson	0.03	1.2
2	Poisson	0.06	2.4
3	Poisson	0.10	4
4	Poisson	0.17	6.8

Fuente: ITU-T Q.543

El comportamiento de las fuentes de voz de telefonía corporativa (perfil “low”) se ha modelado como un período de actividad (entre 1 y 3 minutos según una variable aleatoria de distribución exponencial) más un período de inactividad que también se ha modelado siguiendo una distribución exponencial entre 0 y 120 minutos.

Para el comportamiento de cada fuente del servicio 010 (usuarios hiperactivos - perfil “high”) se ha realizado un dimensionado del servicio de acuerdo con valores comúnmente utilizados en servicios de Call Center [123].

- Llamadas por hora = 120
- Tiempo medio de llamada = 180 segundos.
- Tiempo objetivo de respuesta = 10 segundos.
- Número de agentes = 10

Utilizando un modelo de espera Erlang-C [10] para calcular la intensidad de tráfico con las variables definidas en el párrafo anterior tendremos:

- Intensidad de tráfico (en Erlangs) = 6
- Probabilidad de esperas = 10,13%
- Probabilidad de que una llamada sea respondida antes de 10 segundos = 91.89%

Por lo tanto, podríamos modelar una fuente “high” como la suma de un período de actividad (entre 1 y 6 minutos según una variable aleatoria de distribución exponencial) y un período de inactividad también modelado siguiendo una distribución exponencial entre 0 y 10 minutos.

Para la definición de las fuentes del servicio de videoconferencia IP en calidad HD (1280x720) para 20 usuarios, se ha considerado la información técnica de un servicio comercial [124]. Se ha seleccionado una señal de vídeo de calidad “HD” con un factor de movimiento 1 (equivalente a imagen predecible) y 25 tramas por segundo. En este caso la tasa binaria resultante típica de un códec H.264 fluctúa entre 1.2 y 2.5 Mb/s [95].

El comportamiento de cada fuente se ha modelado de manera arbitraria como la suma de un período de actividad (entre 10 y 60 minutos según una variable aleatoria de distribución exponencial) más un período de inactividad que también se ha modelado siguiendo una distribución exponencial entre 0 y 240 minutos.

Debido a la naturaleza del tráfico de voz, se le ha asignado una clase de servicio “multi-media”, mientras que al vídeo se le ha asignado la inmediatamente inferior “oro”.

Para el dimensionado del caudal se ha procedido a simular cada uno de los tráficos por separado y a comprobar la tasa media en un período de 6 horas para disponer de una primera aproximación de régimen permanente. De estas simulaciones previas se ha realizado un primer cálculo del CIR de cada uno de los servicios (ver tablas de configuración en el siguiente apartado).

5.3.1.2 Simulación del escenario con SimulCity

En este caso, es necesario configurar las tres fuentes de tráfico con los perfiles descritos en párrafos anteriores donde:

- FuenteVoz.H[0].udpApp[*] representa las fuentes de telefonía corporativa.
- FuenteVoz.H[1].udpApp[*] representa las fuentes del servicio “010”.
- FuenteVideo.H[0].udpApp[*] representa las fuentes de videoconferencia.

También, y con el fin de mostrar el impacto de las fuentes en el ancho de banda se ha definido un régimen transitorio de una hora para la generación de tráfico de telefonía corporativa, es decir las 4.500 fuentes de este tipo se activarán de manera uniforme a lo largo de este tiempo. Para las fuentes del servicio “010”, el mecanismo será el mismo pero la ventana de activación se ha reducido a quince minutos. Por último, para las fuentes de vídeo, la ventana se ha establecido también en 15 minutos. Estos parámetros son configurables en el simulador.

La Figura 5.31 muestra la configuración final de las fuentes, la Figura 5.32 la configuración de los perfiles de tráfico y la Figura 5.33 la configuración final de los caudales Metro Ethernet.

```
**FuenteVoz.H[0].udpApp[*].destAddresses = "Cloud.ServerVoz.S[0]"
**FuenteVoz.H[0].udpApp[*].startTime = uniform(0s,3600s)
**.*Voz.*[0].udpApp[*].*Port = 1000 #Tráfico Multimedia
**FuenteVoz.H[0].udpApp[*].codec = xmldoc("VoiceCodec.xml", "//voice-codec[@id='g.711']")
**FuenteVoz.H[0].udpApp[*].config = xmldoc("ConfigTraffic.xml", "//voice-profile[@id='low']")
**FuenteVoz.H[1].udpApp[*].destAddresses = "Cloud.ServerVoz.S[1]"
**FuenteVoz.H[1].udpApp[*].chooseDestAddrMode = "once"
**FuenteVoz.H[1].udpApp[*].startTime = uniform(0s,900s)
**.*Voz.*[1].udpApp[*].*Port = 1000 #Tráfico Multimedia
**FuenteVoz.H[1].udpApp[*].codec = xmldoc("VoiceCodec.xml", "//voice-codec[@id='g.711']")
**FuenteVoz.H[1].udpApp[*].config = xmldoc("ConfigTraffic.xml", "//voice-profile[@id='high']")
**FuenteVideo.H[0].udpApp[*].destAddresses = "Cloud.ServerVideo.S[0]"
**FuenteVideo.H[0].udpApp[*].chooseDestAddrMode = "once"
**FuenteVideo.H[0].udpApp[*].startTime = uniform(0s,900s)
**.*Video.*[0].udpApp[*].*Port = 1001 #Tráfico Oro
**FuenteVideo.H[0].udpApp[*].quality = xmldoc("VideoQuality.xml", "//video-quality[@id='HD']")
**FuenteVideo.H[0].udpApp[*].config = xmldoc("ConfigTraffic.xml", "//video-profile[@id='webex']")
```

Figura 5.31 Configuración de las fuentes de tráfico de voz y multimedia

```
<voice-profile id="low">
  <Duration type='exponential' mean='2' min='1' max='3' />
  <TimeInactive type='exponential' mean='60' min='0' max='120' />
</voice-profile>
<voice-profile id="high">
  <Duration type='exponential' mean='3' min='1' max='6' />
  <TimeInactive type='exponential' mean='2' min='0' max='10' />
</voice-profile>
<video-profile id="webex">
  <Duration type='exponential' mean='30' min='10' max='60' />
  <TimeInactive type='exponential' mean='120' min='0' max='240' />
  <Movement type="constant" value="2" />
  <fps type="constant" value="25" />
</video-profile>
<video-quality id="HD">
  <ResX type="constant" value="1280" />
  <ResY type="constant" value="720" />
</video-quality>
```

Figura 5.32 configuración de los perfiles de tráfico de voz y multimedia

```
** Site[0].Datarate = 1 # 0 = 10Mbps, 1 = 100Mbps, 2 = 1Gbps
** Site[0].RouterHosts.**.ingressTC.meterVoz.cir = "8 Mbps"
** Site[0].RouterHosts.**.ingressTC.meterVoz.cbs = 1MiB
** Site[0].RouterHosts.**.ingressTC.meterVoz.pir = "16Mbps"
** Site[0].RouterHosts.**.ingressTC.meterVoz.pbs = 2MiB
** Site[0].RouterHosts.**.ingressTC.meterOro.cir = "40Mbps"
** Site[0].RouterHosts.**.ingressTC.meterOro.cbs = 6.25MiB
** Site[0].RouterHosts.**.ingressTC.meterOro.pir = "64Mbps"
** Site[0].RouterHosts.**.ingressTC.meterOro.pbs = 8MiB
```

Figura 5.33 Configuración de caudal en el interfaz Metro Ethernet

Una vez ejecutada la simulación para un periodo de una hora, SimulCity proporciona un fichero “ANF” con estadísticas detalladas sobre el comportamiento de las fuentes, el comportamiento de las diferentes colas, el volumen total de tráfico cursado y descartado por fuente, tráfico medio, retardo, Jitter. Etc. En total más 52GB de datos con 90.000 elementos, vectores, escalares e histogramas. De manera muy resumida, se describirán a continuación los parámetros más relevantes.

El ancho de banda consumido por las fuentes de voz se muestra en la Figura 5.34 donde se puede observar que en régimen permanente el caudal no supera los 9 Mb/s. SimulCity proporciona además de los valores medios del tráfico y su desviación típica por fuente única, los instantes de tiempo y tamaños de paquete para su posterior tratamiento estadístico con herramientas especializadas.

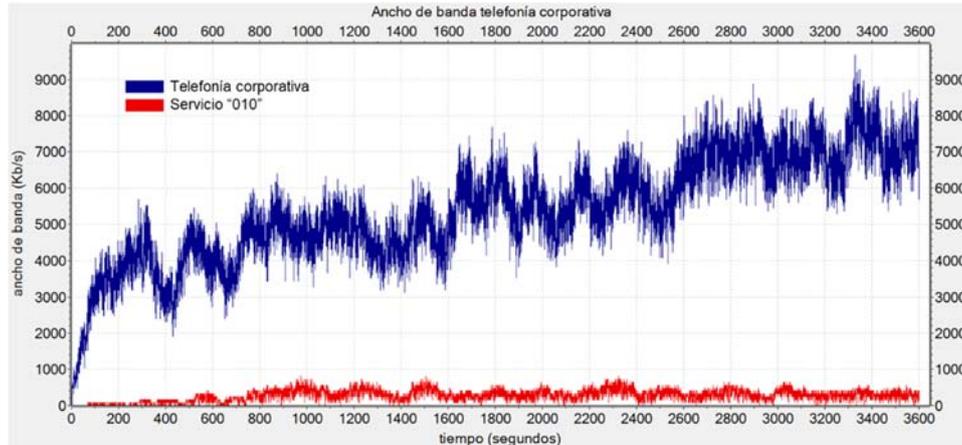


Figura 5.34 Ancho de banda fuentes de voz

El Figura 5.35. se puede observar el comportamiento de las fuentes de vídeo según el perfil de tráfico definido, y que no supera los 45Mb/s. De manera intencionada, se ha introducido un régimen transitorio para poder mostrar las variaciones del ancho de banda por la activación / desactivación de las sesiones de videoconferencia.

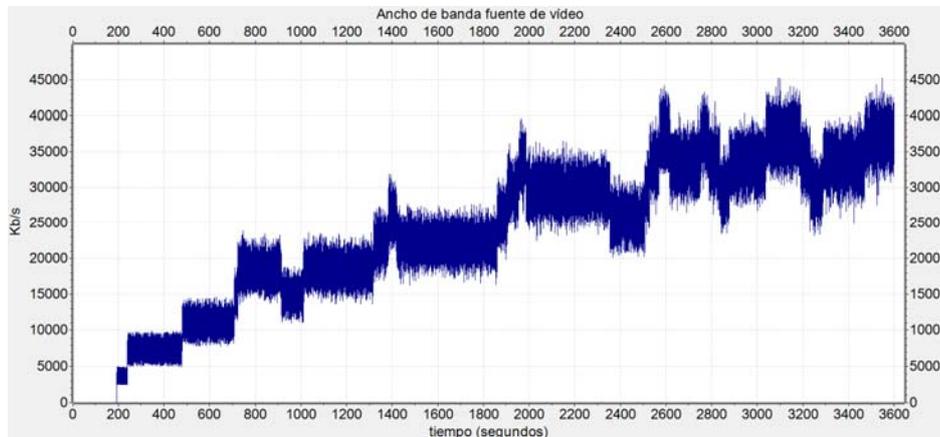


Figura 5.35 Ancho de banda fuente de vídeo

La Figura 5.36 muestra el retardo y el Jitter de las fuentes de voz hasta el servidor de telefonía. Puede observarse el efecto de los mecanismos de control y gestión de colas que incrementan estas variables según aumenta el tráfico. La pérdida de paquetes, el retardo y el Jitter están muy por debajo de lo exigido en las recomendaciones internacionales [79, 80]. La Tabla 5.7 muestra los valores máximos admitidos.

Tabla 5.7 Valores máximos de QoS en fuentes de VoIP

Pérdida de Paquetes (%)	Jitter (ms)	Retardo (ms)
1	10	150

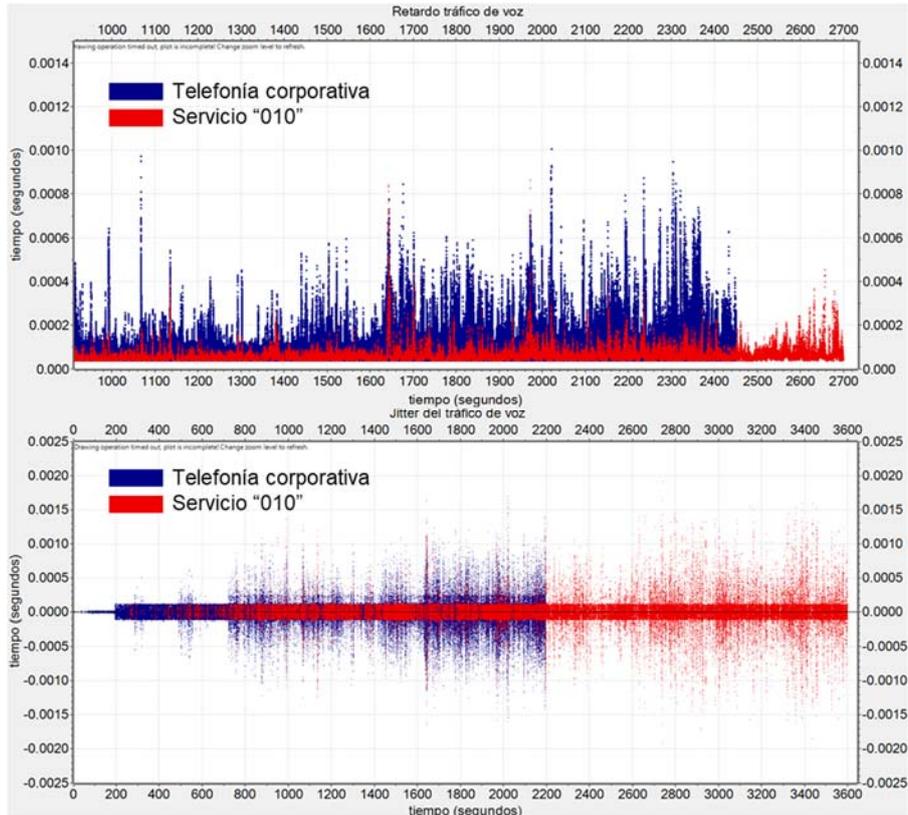


Figura 5.36 retardo y Jitter de las fuentes de voz

La Figura 5.37 muestra el retardo y el Jitter de las fuentes de vídeo hasta el servidor de videoconferencia. Puede observarse el efecto de los mecanismos de control y gestión de colas que incrementan estos parámetros según aumenta el tráfico. La pérdida de paquetes, el retardo y el Jitter están igualmente muy por debajo de lo exigido en las recomendaciones internacionales. La Tabla 5.8 muestra los valores máximos permitidos.

Tabla 5.8 Valores máximos de QoS en fuentes de vídeo

Pérdida de Paquetes (%)	Jitter (ms)	Retardo (ms)
1	30	150

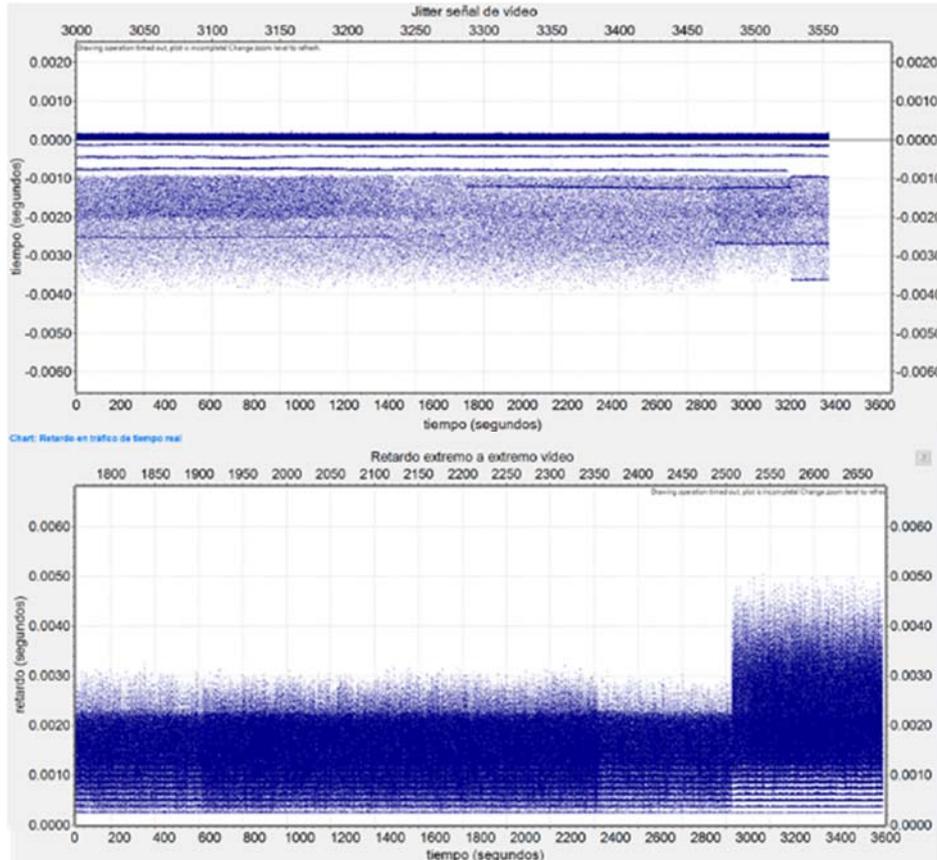


Figura 5.37 Retardo y Jitter de las fuentes de vídeo

Dado que estas gráficas son complejas de evaluar, SimulCity, como ya se mostró en el proyecto de residuos sólidos, puede mostrar la información anterior en forma de histograma para facilitar una mejor interpretación de los datos. Concretamente, en la Figura 5.38 se puede observar un histograma que relaciona número de paquetes y retardo de vídeo en la red.

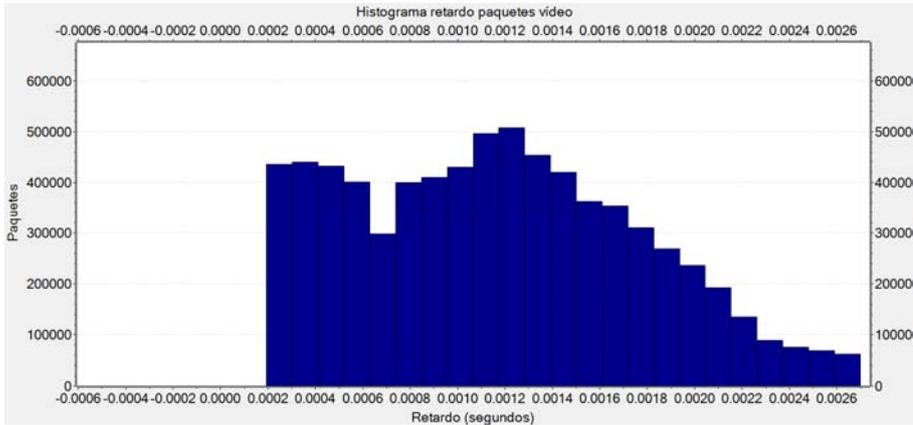


Figura 5.38 Histograma del retardo de paquetes de video en la red

La Figura 5.39 muestra el ancho de banda total. Puede observarse el efecto de los mecanismos de control y gestión de colas que aumentan estos valores según aumenta el tráfico. La pérdida de paquetes, el retardo y el Jitter están muy por debajo de lo exigido en las recomendaciones internacionales [79,80].

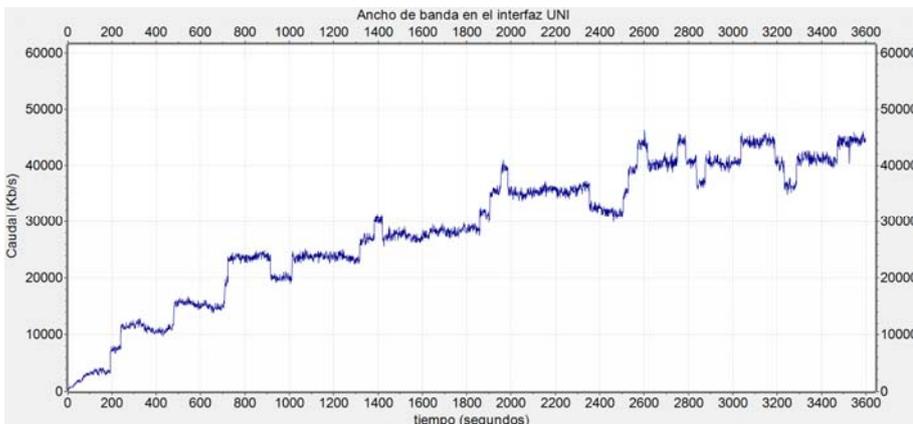


Figura 5.39 Ancho de banda interfaz UNI

Como conclusiones de este proyecto, se ha podido comprobar, a diferencia de la implantación de proyectos con uso masivo de dispositivos IOT del gran impacto que este tipo de actuaciones imponen sobre la red de comunicaciones municipal.

Es aquí donde SimulCity puede contribuir al dimensionado adecuado de los caudales necesarios para mantener la calidad de servicio requerida y disminuir los costes, ya que éstos están directamente relacionados con el tipo y volumen de caudal contratado.

Aunque los parámetros de configuración se han escogido de manera arbitraria, la flexibilidad de la herramienta permite evaluar el impacto de posibles modificaciones, por ejemplo, el incremento del número de fuentes de vídeo por la extensión del servicio de video conferencia para tareas de tele formación municipal.

5.4 Simulación del escenario global

En este apartado se mostrarán los resultados más significativos del escenario global que incluye la contribución de la totalidad de los proyectos descritos en este capítulo. Por limitaciones en la capacidad de almacenamiento en el sistema donde está instalada la herramienta, el tiempo de simulación se ha reducido a 15 minutos.

En esta simulación además se ha reducido drásticamente la ventana de activación de las fuentes a 10 segundos, excepto telefonía corporativa, para mostrar un hipotético de máximo volumen de contribución de las fuentes.

En la Tabla 5.9 se muestra el número de fuentes simultáneas que se simulan en este apartado, en total 8.836 fuentes de naturaleza heterogénea.

Tabla 5.9 Resumen de fuentes por proyecto

Proyecto	Fuentes	Clase Servicio
Fuentes de telefonía corporativa	4500	MULTIMEDIA
Fuentes de Call Center	10	MULTIMEDIA
Videoconferencia HD	20	ORO
Sensores de residuos no perecederos	237	PLATA
Centros de mando de luminarias	744	PLATA
Aparcamiento PMR	1600	PLATA
Aparcamiento CyD	1500	PLATA
Paradas de Taxi	225	PLATA

Para la configuración de los caudales por clase de servicio se han utilizado los valores de la Tabla 5.10 que se han obtenido de los caudales medios de las fuentes simuladas individualmente redondeadas al Mb superior.

Tabla 5.10 Caudales (CIR y EIR)

Clase Servicio	CIR	EIR	Caudal medio
MULTIMEDIA	7 Mb/s.	14 Mb/s.	6,491 Mb/s
ORO	40 Mb/s.	60 Mb/s.	36,416 Mb/s
PLATA	6 Mb/s.	12 Mb/s.	5,466 Mb/s

La Figura 5.40 refleja el ancho de banda consumido por las 4.500 las fuentes de voz. Se puede observar que el flujo de llamadas activas se corresponde aproximadamente a las 135 esperadas (135 fuentes simultáneas x 88Kb/s Códec G.709) con los criterios de diseño que se definieron en el apartado 5.3.1.1.

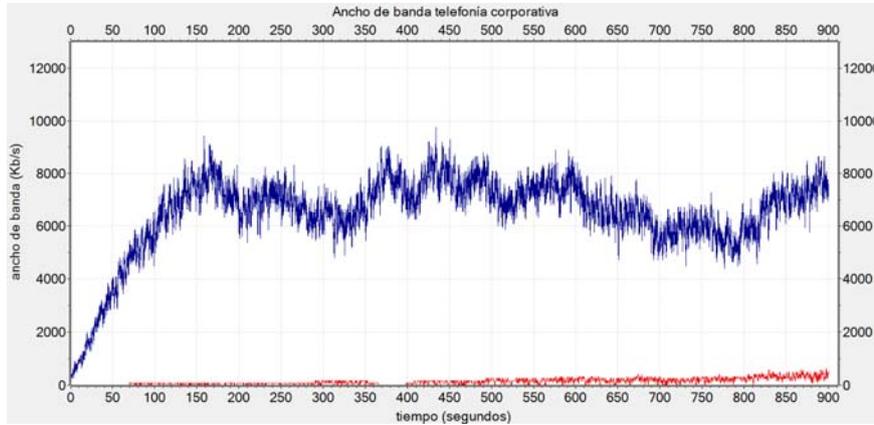


Figura 5.40 Ancho de banda fuentes de voz

La Figura 5.41 refleja el ancho de banda consumido por las fuentes de vídeo. Se puede observar que el flujo de videoconferencias activas se corresponde con las 20 esperadas (20 fuentes simultáneas x 1,9 Mb/s, calidad = HD, movimiento = 1, fps = 25) con los criterios de diseño que se definieron en el apartado 5.3.1.1.

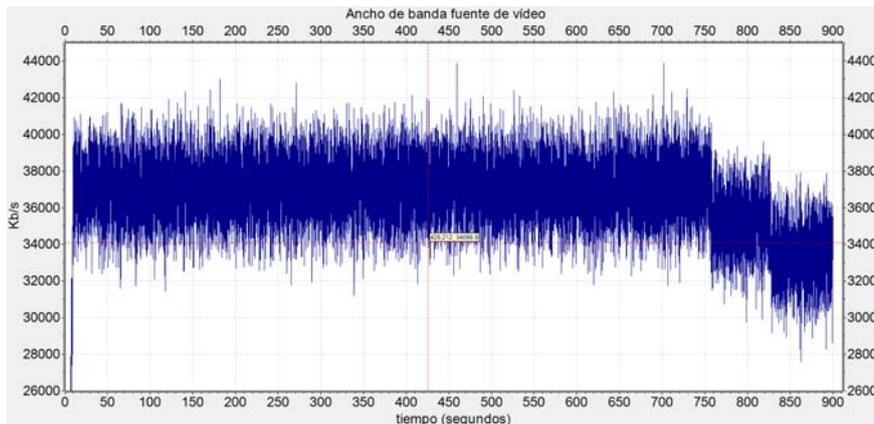


Figura 5.41 Ancho de banda fuentes de vídeo

La Figura 5.42 refleja el ancho de banda consumido por los proyectos de residuos sólidos, luminarias, PMR, Carga y Descarga y Taxi. Solamente es relevante el impacto del proyecto de control de luminarias por el elevado número de elementos a gestionar.

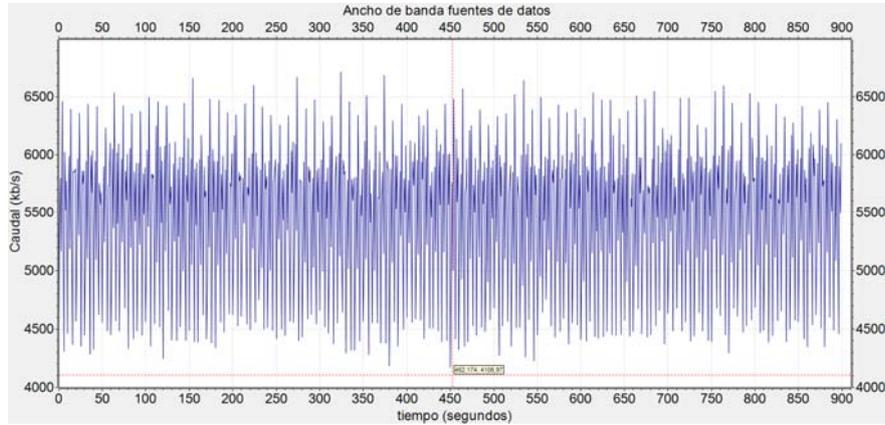


Figura 5.42 Ancho de banda agregado de las fuentes de datos IoT

La Figura 5.43 refleja el ancho de banda entregado a la red en el interfaz UNI. Los caudales, como se comentó en párrafos anteriores, se han dimensionado para evitar pérdidas. Cabe destacar que, como resultado de este dimensionado, el caudal máximo se sitúa sobre los 52 Mb/s.

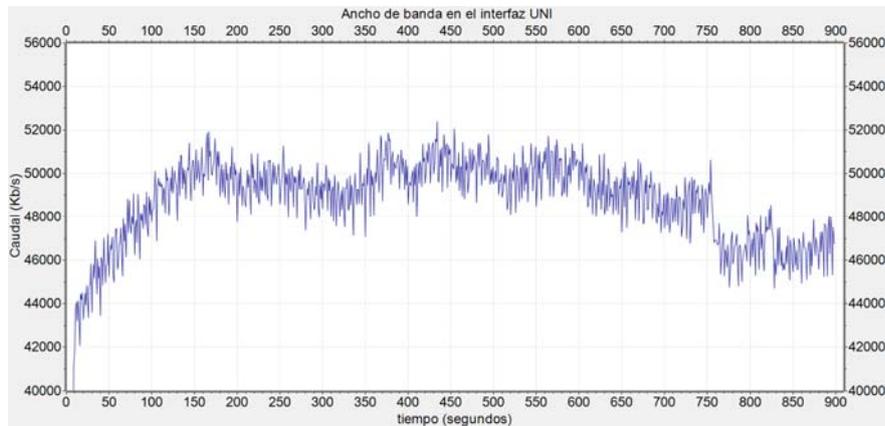


Figura 5.43 Ancho de banda interfaz UNI

La Figura 5.44 muestra el retardo del tráfico de voz en forma de histograma. El retardo prácticamente es constante (obsérvese que el número de paquetes se ha dibujado en escala logarítmica) y es prácticamente constante en torno a los 40 microsegundos, muy alejado de valores que pudieran poner en riesgo la calidad de servicio ofrecida.

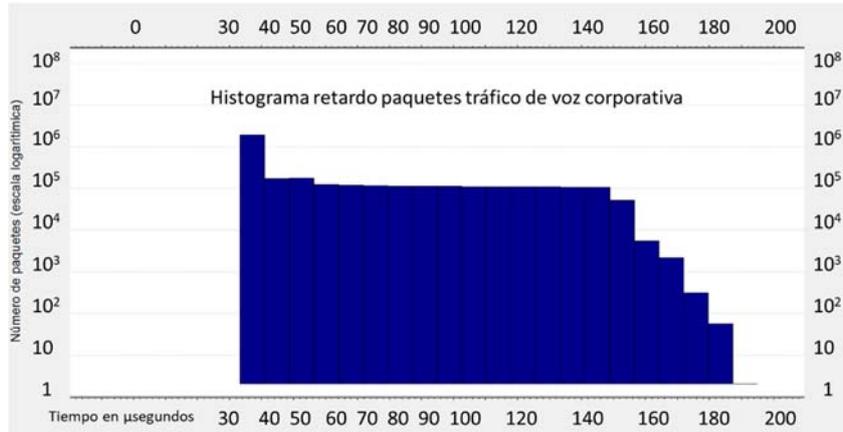


Figura 5.44 Retardo tráfico de voz

La Figura 5.45 muestra el histograma del retardo del tráfico de vídeo. El retardo ahora está más distribuido (obsérvese que ahora el número de paquetes se ha dibujado en escala lineal) y está distribuido entre los 200 microsegundos y 1,7 milisegundos, muy alejado igualmente de valores que pudieran poner en riesgo la calidad de servicio ofrecida. Esta distribución puede explicarse por el impacto en las colas de la activación de fuentes de vídeo de manera simultánea.

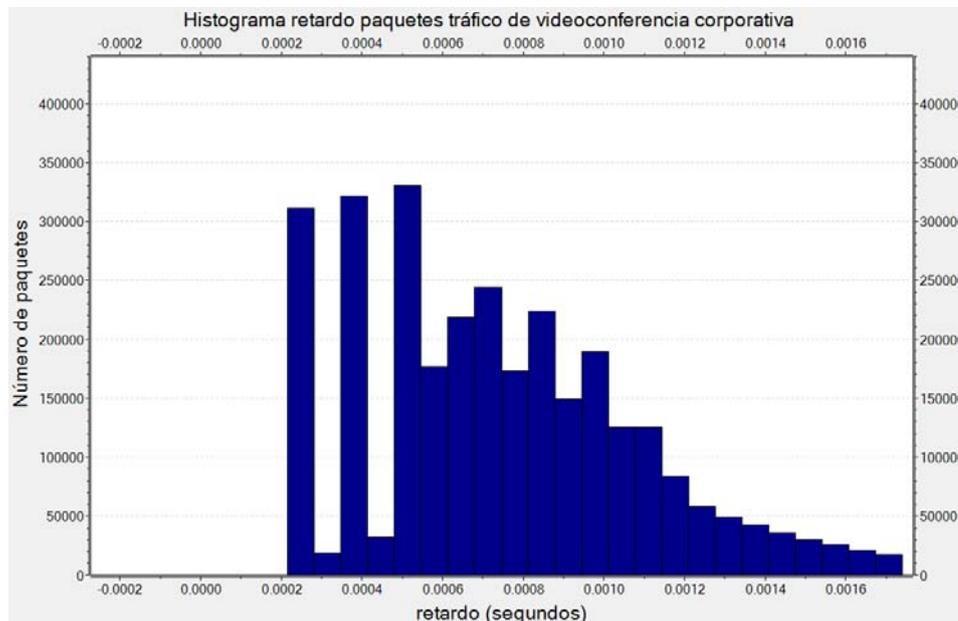


Figura 5.45 Retardo tráfico de vídeo

Como conclusión de este apartado, se ha podido realizar un dimensionamiento adecuado de cada uno de los tipos de caudales que ofrece el operador: multimedia, oro y plata y se ha evaluado el comportamiento global de la solución. Se ha comprobado que los valores ofrecidos por SimulCity para un entorno de hora cargada están dentro de lo esperado y que el perfil de tráfico está claramente dominado por los servicios de telefonía corporativa y videoconferencia.

Los servicios de IoT representan una parte residual del tráfico excepto cuando el número de dispositivos a gestionar es muy elevado como, por ejemplo, el proyecto de iluminación.

Otro elemento que conviene destacar es el consumo de recursos de computación. El aumento del número de fuentes a simular incide drásticamente en el tiempo de simulación. Por ejemplo, por cada segundo de simulación de este último escenario se han necesitado aproximadamente 18 segundos de cómputo, (utilizando un procesador Intel I7 4790K de 4 núcleos físicos y 4GHz. y 16GB de RAM). En cuanto al almacenamiento, SimulCity ofrece una gran cantidad de información sobre la simulación realizada, pero esto puede ser un problema en escenarios como el dibujado, donde 15 minutos de simulación requieren almacenar más de 50 GB de datos.

5.5 Conclusiones

En este capítulo se han descrito diferentes casos de uso de la herramienta SimulCity dentro de los proyectos actualmente en marcha del Ayuntamiento de València y su utilidad para dimensionar adecuadamente los caudales del interfaz Metro Ethernet que conecta la red corporativa municipal con la Plataforma de Ciudad que reside en un Data-Center de Telefónica.

En primer lugar, se han evaluado diferentes proyectos en el ámbito del IoT relacionados con la Iniciativa “Impulso VLCi”, proyecto de recogida de residuos sólidos no perecederos, alumbrado y sensores de aparcamiento. En todos los casos se ha definido una arquitectura de comunicaciones, un patrón de tráfico para las fuentes se ha realizado una simulación del sistema y obtenido resultados.

Solamente la gestión de alumbrado ha producido un volumen de datos reseñable debido fundamentalmente al elevado número de dispositivos a gestionar y a la necesidad de monitorización continua para evitar el robo de cobre. Como era de esperar, este tipo de proyectos basados en dispositivos con tan escasos requerimientos temporales, no tienen impacto significativo en la arquitectura de comunicaciones municipal.

Sin embargo, la migración de la red de comunicaciones municipal de voz basada en TDM a una telefonía IP corporativa junto con la implantación de una red de videoconferencia IP ha permitido explotar al máximo las capacidades de la herramienta.

SimulCity ha simulado con éxito el comportamiento global de fuentes de tráfico de gran número de dispositivos: 110.600 dispositivos IoT, 4.500 fuentes de voz de telefonía

corporativa, 10 agentes del servicio de atención al ciudadano “010”, y un servicio de videoconferencia corporativo con una capacidad de hasta 20 usuarios.

Los resultados obtenidos han permitido conseguir un dimensionado de caudales, para cada una de las clases de servicio contratables, ajustado a las necesidades reales de la corporación.

Capítulo 6

Conclusiones y líneas de trabajo futuras

6.1 Conclusiones

En esta tesis de investigación aplicada se ha diseñado una herramienta, denominada “SimulCity”, que permite evaluar el impacto de la introducción de nuevos servicios relacionados con el concepto de “Ciudad Inteligente” en la red de comunicaciones municipal.

Las conclusiones se han estructurado en tres grandes bloques relacionados con

- El marco de referencia de las Ciudades Inteligentes.
- La arquitectura de red y el análisis y caracterización del tráfico.
- La herramienta de simulación “SimulCity”.

6.1.1 Marco de referencia

Para conseguir los objetivos propuestos en esta tesis es necesario conocer el marco de referencia de las Smart Cities e identificar los atributos y requisitos que permitirían a una ciudad denominarse “inteligente”. En base a estos atributos y requisitos, este conocimiento permitiría diseñar una arquitectura TIC municipal para dotar de inteligencia a las infraestructuras físicas, y su capacidad de relación con el entorno, personas o sistemas.

También es necesario conocer la arquitectura del modelo de capas TIC que se relacionan entre sí a través de interfaces normalizados. La capa de captación es la encargada de obtener información del entorno y actuar sobre las infraestructuras físicas o las personas; contiene sensores, actuadores y terminales. La siguiente capa, es la capa de red (comunicaciones) encargada de consolidar la información de la capa de captación y transmitirla

a la capa de servicios, soporte y contenidos. Existen además otras capas complementarias (medios, datos y soporte) que permiten realizar determinadas funciones auxiliares.

Por último, el conocimiento de la arquitectura y servicios proporcionados por una Plataforma de Ciudad, elemento crítico de la arquitectura, permitirá complementar el modelo global, consolidar la información del mundo físico, establecer lógicas de negocio y proveer de mecanismos de análisis para la toma de decisiones

6.1.2 Arquitectura de red, análisis y caracterización de tráfico

Una vez establecido el marco conceptual de Ciudad Inteligente se pone de manifiesto la complejidad subyacente de una red donde pueden convivir múltiples tecnologías tanto de captación, acceso y transporte y la necesidad de diseñar un modelo de red convergente (network model) que se asemeje lo más posible a la realidad de una red municipal, asumiendo que la Plataforma de Ciudad se encuentra alojada en un Centro de Datos externo.

Con esta red modelo ya definida, el siguiente paso ha sido identificar los diferentes tipos de tráfico existentes o que puedan existir y modelar su comportamiento. Una primera conclusión, dado el número y variedad de fuentes heterogéneas, indica la conveniencia de estudiarlas agrupadas por tipología.

6.1.2.1 Caracterización de las fuentes de voz

En las fuentes de voz (VoIP), la literatura existente y las referencias bibliográficas estudiadas, muestran diferentes aproximaciones para modelar analíticamente este tipo de fuentes. Casi todas ellas capturan un tráfico de muestra real y ajustan una función analítica que se adapte al tráfico capturado. Por otra parte, los códecs avanzados utilizan propiedades características de la voz, como la supresión de silencios y generan, por tanto, patrones y tasas de bit muy correladas con el comportamiento del hablante (idioma y conversación). Estos enfoques basados en modelos ON-OFF adolecen de generalidad. Una segunda conclusión es que para modelar una fuente de voz es necesario modelar el funcionamiento del códec concreto y posteriormente los períodos de actividad / inactividad con funciones de distribución flexibles.

6.1.2.2 Caracterización de las fuentes de vídeo

En las fuentes de vídeo, a diferencia del tráfico de voz, son flujos continuos, y los análisis de comportamiento indican, en función de los códecs empleados (de tasa de bit variable), distribuciones tipo log-normal, pero como ocurre en el tráfico de VoIP, el ancho de banda necesario para una fuente de vídeo presenta una fuerte correlación con la naturaleza de la fuente individual, es decir, de la cantidad de movimiento de la imagen, además de su resolución y del códec utilizado. Una tercera conclusión apoya la conveniencia de introducir en los modelos de fuentes de vídeo un factor adicional que permita modelar con mayor exactitud diferentes servicios de vídeo en el entorno municipal como pueden ser servicios de teleformación o videoconferencia.

6.1.2.3 *Caracterización de las fuentes MTC*

En las fuentes MTC, dos características se pueden extraer de estas fuentes, comportamientos muy dispares y cierto grado de “sincronización” entre fuentes homogéneas, aspecto que hace aún más complejo el posible modelado analítico. Ante esta dificultad, se han desarrollado modelos de tráfico “agregado” en el que las fuentes se analizan como un único flujo. 3GPP ha definido varios modelos, pero éstos no son suficientemente válidos para un número reducido de fuentes y patrones de tráfico complejo. Una cuarta conclusión de este apartado, es que el análisis analítico del tráfico en redes convergentes implica gran complejidad y que por tanto los modelos de simulación muestran un adecuado balance entre complejidad, flexibilidad y resultados.

6.1.3 *SimulCity*

Como contribución más relevante, se ha desarrollado una aplicación basada en el entorno de simulación OMNET++ que permite el modelado y simulación de entornos HTC+ MTC en una Smart City evaluando diferentes parámetros de calidad de servicio extremo-extremo, ancho de banda, pérdida de paquetes, retardo y jitter de un conjunto arbitrario de fuentes heterogéneas.

6.1.3.1 *Modelado de las fuentes de tráfico*

La herramienta permite definir un número arbitrario de grupos de fuentes homogéneas donde cada fuente de tráfico puede definirse con un modelo de tráfico (perfil) característico y un comportamiento (actividad) que generalmente es una variable aleatoria con una distribución definida. Para ello se han definido tres tipologías: voz, vídeo y MTC, cada una con parámetros específicos.

6.1.3.2 *Simulación y evaluación de los mecanismos de Calidad de Servicio (QoS)*

Con el fin de obtener una simulación lo más similar posible a un entorno real, se ha utilizado como referencia un servicio IP basado en la tecnología Metro Ethernet ofrecido por un operador español. En este servicio se definen tres categorías de tráfico: Multimedia, Oro y Plata, siendo Multimedia del tráfico con la máxima prioridad y Plata el de mínima.

Se ha definido un módulo que simula las funciones básicas de un router de operadora donde se pueden definir diferentes parámetros contratables al operador con el fin de evaluar el rendimiento y la calidad de servicio entre la red municipal y la plataforma de ciudad.

La simulación del Core Network, retardo y pérdida de paquetes se ha realizado aplicando a cada paquete una variable aleatoria exponencial negativa acotada con los valores de nivel de servicio (SLA) máximos y mínimos definidos por el operador en el contrato del servicio.

6.1.3.3 *Evaluación del tráfico y calidad de servicio extremo a extremo. Dimensionado de caudales, servicios y aplicaciones verticales*

En el dominio de las Aplicaciones confluye el tráfico generado por las diferentes redes de acceso de la Smart-City. En este punto, se ha analizado el comportamiento y prestaciones relacionadas con la gestión de la calidad de servicio extremo a extremo por cada clase de servicio y/o servicio subyacente: retardo total, jitter y pérdida de paquetes.

6.2 **Futuras líneas de trabajo**

Aunque se ha contextualizado el entorno tecnológico de una Ciudad Inteligente, la introducción de nuevas tecnologías como el NBloT, y las redes 5G plantean nuevos retos en la definición de la arquitectura de comunicaciones de la ciudad. Destaca especialmente el impacto del despliegue en el entorno urbano de las redes 5G sobre la infraestructura física existente y cómo se verán afectados los servicios de conectividad tanto para ciudadanos como para la propia red corporativa municipal.

El autor propone dos futuras líneas de trabajo. La primera relacionada con aspectos relacionados con el modelado y caracterización de fuentes de tráfico, y una segunda relacionada con mejoras y nuevos desarrollos de la herramienta SimulCity.

6.2.1 *Caracterización de nuevas fuentes de tráfico y control de la calidad de servicio*

Los retos para garantizar la calidad de servicio de “múltiples clases de servicio” que existirán en las redes 5G exigirán no solamente un análisis mucho más minucioso del comportamiento de las nuevas fuentes de datos [125], sino que también el desarrollo de nuevos mecanismos que aseguren la calidad del tráfico en un entorno donde convivirán servicios multimedia con miles de dispositivos IoT. Una primera línea de trabajo consistiría en profundizar en los resultados de esta tesis para su aplicación en redes móviles 5G. Una segunda línea incidiría en la adaptación de los mecanismos de control de congestión y priorización de tráfico, y por último una tercera línea incidiría en el modelado de fuentes agregadas en ese entorno, así como el grado de autosimilitud que ese tipo de tráfico en entornos heterogéneos.

6.2.2 *Evolución de la herramienta SimulCity*

SimulCity ha demostrado un funcionamiento adecuado con los objetivos perseguidos y ha sido capaz de caracterizar el tráfico agregado de miles de fuentes en el interfaz de la red corporativa municipal con la plataforma de ciudad. Sin embargo, de la experiencia de uso de SimulCity se han identificado diversas funcionalidades que son susceptibles de mejora, así como la necesidad de ampliar sus capacidades. Dentro de las futuras versiones se incluirían mejoras en prestaciones, modelado de fuentes y usabilidad.

La primera línea de trabajo persigue mejorar las prestaciones. SimulCity genera un volumen de información que crece potencialmente con el número de fuentes a simular y

que en gran parte de los casos no es estrictamente necesaria. La consecuencia es el consumo de una capacidad de proceso significativa que resta velocidad a la simulación. Una posible línea de mejora sería configurar previamente a la simulación los vectores y escalares a evaluar.

La segunda línea de trabajo estaría relacionada con el modelado de fuentes. La flexibilidad de la herramienta permite simular una gran variedad de fuentes, pero sería conveniente que SimuCity pudiera procesar fuentes modeladas por la extracción de tráfico real de las fuentes. Esta facilidad permitiría estudiar aspectos como el grado de autosimilitud del tráfico

La tercera línea de trabajo estaría relacionada con la usabilidad. Como se ha podido comprobar en diferentes capítulos donde se han mostrado resultados de SimuCity, la introducción de los parámetros de configuración se realiza actualmente mediante ficheros XML que es una tarea tediosa y sujeta a errores. En este sentido, se desarrollará un interfaz gráfico que permita la configuración de las fuentes de tráfico y parámetros de dimensionado de la red y calidad de servicio bajo un entorno gráfico. Además, se realizará la comprobación automática de la coherencia de los valores introducidos en la herramienta, evitando de esta manera errores en la simulación.

Bibliografía

- [1] Naciones Unidas, Agenda 2030: Objetivos de Desarrollo Sostenible A/RES/70/1. Disponible en: <http://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/> (2015).
- [2] United Nations Conference on Housing and Sustainable Urban Development (Habitat III), final document, A/conf.26/4, Quito (Ecuador), (2016).
- [3] United Nations Conference on Housing and Sustainable Urban Development (Habitat III), Policy paper 9: Urban services and technology, A/conf.226/PC.3/22, Quito (Ecuador), (2016).
- [4] V. Gutiérrez, D. Amaxilatis, G. Mylonas and L. Muñoz, "Empowering Citizens Toward the Co-Creation of Sustainable Cities," in *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 668-676, April 2018. doi: 10.1109/JIOT.2017.2743783
- [5] L. Sánchez et al., "SmartSantander: IoT experimentation over a smart city testbed", *Computer Networks*, vol. 61, pp. 217–238, 2014. doi: 10.1016/j.bjp.2013.12.020.
- [6] CO59/15-AE, Ayudas para el desarrollo del programa de ciudades inteligentes de la Agenda Digital para España <http://boe.es/boe/dias/2015/08/01/pdfs/BOE-A-2015-8703.pdf> (2015).
- [7] ITU-T Focus Group on St Sustainable Cities. An overview of sustainable cities and the role of information and communication technologies TR1/WG1 (10/2014)
- [8] ITU-T Recommendation Y. 2001- Next Generation Networks General Overview.
- [9] ITU-T Recommendation Y. 2011- Next Generation Networks – Frameworks and functional architecture models. General principles and general reference model for Next Generation Networks.

- [10] 3GPP TR 23.888, System improvements for Machine-Type Communications (MTC) (Release 11) (2012-09)
- [11] ITU-T Y.2221 Next Generation Networks – Service aspects: Service capabilities and service architecture Requirements for support of ubiquitous sensor network (USN) applications and services in the NGN environment (01/2010)
- [12] ITU-T Focus Group on St Sustainable Cities, Cybersecurity, data protection and cyber resilience in sustainable cities (03/2015)
- [13] ITU-T Recommendation Y. 1541, Quality of service and network performance, Network performance objectives for IP-based services (12/2011)
- [14] ITU-T Recommendation Y. 2617, Quality of service guaranteed mechanisms and performance model for public packet telecommunication data networks (06/2016)
- [15] Service requirements for machine-type communications 3GPP TS 22.368 V12.4.0 (2014-10)
- [16] ITU-T Q.3925 Testing for next generation networks, Traffic flow types for testing quality of service parameters on model networks (03/2012)
- [17] Thomas Telkamp, Best Practices for Determining the Traffic Matrix in IP Networks V 3.0, NANOG 39, February 2007, Toronto, <<http://www.cariden.com/technologies/papers.html>
- [18] R. Goleva, D. Atamian, S. Mirtchev and D. Dimitrova, ‘3G Network Traffic Sources Measurement and Analysis’, Transactions on Emerging Telecommunications Technologies, 25, (2014), 8, pp. 798-814. DOI: 10.1002/ett.2703
- [19] A.D. Shaikh, K.J Blow, M.A. Eberhard and S.A. Fowler ‘Language independent on-off voice over IP source model with lognormal transitions’, IET Communications, 7 (2013), DOI: 10.1049/iet-com.2012.0686, 1449 - 1455
- [20] Evans M. Filfilsh C.: Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice, Morgan Kaufmann 2007

- [21] ITU-T Recommendation Y.2112: A QoS control architecture for Ethernet-based IP access networks (2007)
- [22] ITU-T Recommendation Y.2113: Ethernet QoS control for next generation networks (2009)
- [23] Telefonica S.A., ‘Macrolan Service’ (<<http://www.movistar.es/grandes-empresas/soluciones>>) (2014)
- [24] Cisco Systems Inc., ‘Cisco Catalyst 3650 Configuration Guide’ (<<http://www.cisco.com/c/en/us/support/switches/catalyst-3650-series-switches/products-installation-and-configuration-guides-list.html>>) (2014)
- [25] European Parliament-Policy Department Economic and scientific policy (2014): Mapping Smart Cities in the EU
- [26] British Standards Institution (BSI) PAS 181:2014 Smart city framework, 2014.
- [27] British Standards Institution (BSI) PAS 182:2014 Smart city concept model, Guide to establishing a model for data interoperability, 2014.
- [28] ITU-T Y.frame-scc (under study), Framework and high-level requirements of smart cities and communities, 10 November 2015, available at: <<http://handle.itu.int/11.1002/roadmap/2.1.13672>>, downloaded: 28/09/2017.
- [29] ITU-T Y Suppl. 33 (01/2016), ITU-T Y.4000 series – Smart sustainable cities - Master plan, 26 January 2016, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12760>>, downloaded: 28/09/2017.
- [30] ITU-T Y.SC-Overview (under study), An overview of smart cities and communities and the role of information and communication technologies, 10 November 2015, available at: <<http://handle.itu.int/11.1002/roadmap/2.1.13668>>, downloaded: 28/09/2017
- [31] ITU-T Y.infra (under study), Overview of city infrastructure, 10 November 2015, available at: <<http://handle.itu.int/11.1002/roadmap/2.1.13673>>, downloaded: 28/09/2017

- [32] ITU-T Y Suppl. 27 (01/2016), ITU-T Y.4400 series – Smart sustainable cities - Setting the framework for an ICT architecture, 26 January 2016, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12753>>, downloaded: 28/09/2017
- [33] ITU-T Y Suppl. 29 (01/2016), ITU-T Y.4250 series – Smart Sustainable Cities - Multi-service infrastructure in new-development areas, 26 January 2016, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12755>>, downloaded: 28/09/2017
- [34] ITU-T Y Suppl. 30 (01/2016), ITU-T Y.4250 series – Smart sustainable cities - Overview of smart sustainable cities infrastructure, 26 January 2016, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12756>>, downloaded: 28/09/2017
- [35] ITU-T Y.isw-ssc (under study), The Integrated Sensor Web Resource Metadata for Smart Sustainable Cities, 10 November 2015, available at: <<http://handle.itu.int/11.1002/roadmap/2.1.13675>>, downloaded: 28/09/2017
- [36] ITU-T Y.ism-ssc (under study), A Technical Framework of Integrated Sensing & Management for Smart Sustainable Cities, 10 November 2015, available at: <<http://handle.itu.int/11.1002/roadmap/2.1.13674>>, downloaded: 28/09/2017
- [37] ITU-T Y.4805 (under study), Identifier service requirements for the interoperability of Smart City applications, 10 November 2015, available at: <<http://handle.itu.int/11.1002/roadmap/2.1.13671>>, downloaded: 28/09/2017
- [38] ITU-T Y Suppl. 31 (01/2016), ITU-T Y.4550 series – Smart sustainable cities - Intelligent sustainable buildings, 26 January 2016, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12757>>, downloaded: 28/09/2017
- [39] ITU-T Y Suppl. 28 (01/2016), ITU-T Y.4550 series – Smart sustainable cities - Integrated management, 26 January 2016, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12754>>, downloaded: 28/09/2017
- [40] ITU-T Y.SC-OpenData (under study), Framework of Open Data in Smart Cities, 10 November 2015, available at: <<http://handle.itu.int/11.1002/roadmap/2.1.13670>>, downloaded: 28/09/2017

- [41] ITU-T Y.SC-Residential (under study), Requirements of Smart Residential Communities, 10 November 2015, available at: <<http://handle.itu.int/11.1002/roadmap/2.1.13669>>, downloaded: 28/09/2017
- [42] ITU-T Y.4900/L.1600 (06/2016), Overview of key performance indicators in smart sustainable cities, 5 July 2017, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12627>>, downloaded: 28/09/2017
- [43] ITU-T Y.4902/L.1602 (06/2016), Key performance indicators related to the sustainability impacts of information and communication technology in smart sustainable cities, 5 July 2017, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12662>>, downloaded: 28/09/2017
- [44] ITU-T Y.4901/L.1601 (06/2016), Key performance indicators related to the use of information and communication technology in smart sustainable cities, 5 July 2017, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12661>>, downloaded: 28/09/2017.
- [45] ITU-T Y.4903/L.1603 (10/2016), Key performance indicators for smart sustainable cities to assess the achievement of sustainable development goals, july 2017, available at: <<http://handle.itu.int/11.1002/roadmap/2.1000.12884>>, downloaded: 28/09/2017
- [46] City Protocol Agreement: CPA-I_001_Anatomy City Anatomy: A Framework to support City Governance, Evaluation and Transformation (02/2015).
- [47] Red.es, Estudio y Guía Metodológica sobre Ciudades Inteligentes (05/2015).
- [48] ISO/TC 268: Sustainable development in communities.UNE-ISO 37120:2015 Desarrollo sostenible en las ciudades. Indicadores para los servicios urbanos y calidad de vida.
- [49] AENOR AEN/CTN 178 PNE 178201:2016 “Ciudades inteligentes. Definición, atributos y requisitos”, Asociación Española de Normalización (AENOR)
- [50] Designing the Internet of Things Adrian MCEwen, Hakim Cassimaly, Willey, ISBN 978-1-118-43062-0 (paperback); ISBN 978-1-118-43063-7

- [51] ITU. Recommendation ITU-T Y.2060 (06/2012), Next Generation Networks – Frameworks and functional architecture models. Overview of the Internet of things, <<http://handle.itu.int/11.1002/1000/11559>>, 2012.
- [52] 3GPP TS 36.306 V14.4.0 (2017-09) Technical Specification Group Radio Access Network, Evolved Universal Terrestrial Radio Access (E-UTRA), User Equipment (UE) radio access capabilities (Release 14)
- [53] AENOR AEN/CTN 178 PNE 178104 v2 (borrador) “Ciudades Inteligentes. Requisitos de interoperabilidad para una Plataforma de Ciudad Inteligente”, Asociación Española de Normalización (AENOR)
- [54] FIWARE Community. <<https://www.fiware.org/>> [Consulta: 24 de junio de 2018]
- [55] FIWARE Community. *Fiware-iot-stack*. <<http://fiware-iot-stack.readthedocs.io/en/release-v4.1/>> [Consulta: 24 de junio de 2018]
- [56] FIWARE Community. *Fiware-iot-stack data_api*. <http://fiware-iot-stack.readthedocs.io/en/release-v4.1/data_api/index.html> [Consulta: 24 de junio de 2018]
- [57] FIWARE Community. *Fiware-orion apiv1*. <https://fiware-orion.readthedocs.io/en/master/user/walkthrough_apiv1/index.html> [Consulta: 24 de junio de 2018]
- [58] APACHE Community. *Hadoop*. <<http://hadoop.apache.org/>> [Consulta: 24 de junio de 2018]
- [59] MONGODB. *Mongodb*. <<https://www.mongodb.com/es>> [Consulta: 24 de junio de 2018]
- [60] APACHE Community. *Flume* <<http://flume.apache.org>> [Consulta: 24 de junio de 2018]
- [61] FIWARE Community. *Fiware-cygnus*. <<https://catalogue.fiware.org/enablers/cygnus>> [Consulta: 24 de junio de 2018]
- [62] FIWARE Community. *Fiware-ckan*. <<http://docs.ckan.org/en/ckan-2.3.4/>> [Consulta: 24 de junio de 2018]

- [63] FIWARE Community. *Fiware-sth-comet*
<<https://catalogue.fiware.org/enablers/sth-comet>> [Consulta: 24 de junio de 2018]
- [64] Microstrategy. *Microstrategy BI*. <<https://www.microstrategy.com/es>>
[Consulta: 24 de junio de 2018]
- [65] Talend Community. *Telefonica iop Talend connectors*
<<https://github.com/telefonicaid/iotp-talend-connectors>> [Consulta: 24 de junio de 2018]
- [66] Talend Community. *Telefonica iop Talend data-integration*
<<http://www.talend.com/products/data-integration>> [Consulta: 24 de junio de 2018]
- [67] JBOSS. *Jboss fuse overview*. <<http://www.jboss.org/products/fuse/overview/>>
[Consulta: 24 de junio de 2018]
- [68] FIWARE Community. *Fiware CEP*.
<<https://catalogue.fiware.org/enablers/complex-event-processing-cep-proactive-technology-online>> [Consulta: 24 de junio de 2018]
- [69] FIWARE Community. *Fiware identity management keyrock*
<<https://catalogue.fiware.org/enablers/identity-management-keyrock>>
[Consulta: 24 de junio de 2018]
- [70] FIWARE Community. *Fiware pep-proxy-wilma CEP*. <<https://catalogue.fiware.org/enablers/pep-proxy-wilma>> [Consulta: 24 de junio de 2018]
- [71] ITU-T Recommendation G. 1029 Multimedia Quality of Service and performance – Generic and user-related aspects - Voice service diagnosis framework (02/2014)
- [72] ITU-T Recommendation Q. 543 Objetivos de diseño para la calidad de funcionamiento de las centrales digitales (03/1993)
- [73] ITU-T Recommendation Q. 3900 Methods of testing and model network architecture for NGN technical means testing as applied to public telecommunication networks (09/2006)

- [74] Huang J, Qian F, Mao ZM, Sen S, Spatscheck O. Screen-off traffic characterization and optimization in 3G/4G networks, In ACM conf. on Internet measurement (IMC '12), Boston, MA, USA, 2012; 357-364, DOI:10.1145/2398776.2398813.
- [75] A. Varet and N. Larrieu, "How to generate realistic network traffic," in Proc. IEEE 38th Annual International Computers, Software & Applications Conference (COMPSAC), 2014.
- [76] A. P. Vela, A. Vía, F. Morales, M. Ruiz and L. Velasco, "Traffic generation for telecom cloud-based simulation," 2016 18th International Conference on Transparent Optical Networks (ICTON), Trento, 2016, pp. 1-4.doi: 10.1109/ICTON.2016.7550544.
- [77] ITU-T Recommendation Q. 3925 Traffic flow types for testing quality of service on model networks (03/2012)
- [78] ITU-T Recommendation Y.1540 Internet protocol aspects – Quality of service and network performance Internet protocol data communication service – IP packet transfer and availability performance parameters. (07/2016).
- [79] ITU-T Recommendation Y.1541 Internet protocol aspects – Quality of service and network performance Internet protocol data communication service – Network performance objectives for IP-based services. (12/2011)
- [80] Iversen, V.B. (2010), Teletraffic Engineering and Network Planning, Technical University of Denmark.
<http://www.osti.gov/eprints/topicpages/documents/record/982/1473132.html>.
- [81] Sheluhin, O. I., Smolskiy, S. M. and Osin, A. V. (2007). Self-Similar Processes in Telecommunications (eds O. I. Sheluhin, S. M. Smolskiy and A. V. Osin). doi:10.1002/9780470062098.ch2.
- [82] Willinger, W., Taqqu, M., Sherman, R., and Wilson, D. (1997), Self-similarity through high-variability. IEEE/ACM Transaction on Networking, vol. 5, No. 1, 1997, pp. 71-86.
- [83] Mirtchev S, Goleva R. Discrete time single server queueing model with a multimodal packet size distribution. In Modeling and Control of Information Processes, Atanasova T (ed.). Sofia: Bulgaria, 2009; 83–101.

- [84] Birke, R., Mellia, M., Petracca, M., and Rossi, D. (2010), Experience of VoIP Traffic Monitoring in a Commercial ISP, *International Journal of Network Management*, vol. 20, Issue 5, pp. 339-359.
- [85] ITU-T Recommendation P.59 Telephone Transmission Quality Objective measuring apparatus – Artificial Conversational Speech (03/1993)
- [86] A.E.García, K.D. Hackbarth, A.Brand, R.Lehnert: Analytical Model for Voice over IP traffic characterization (2002)
- [87] ITU-T Recommendation H.261 Line transmission of non-telephone signals, video codec for audiovisual services at p x 64 kbits, (03/93).
- [88] Overview of the H.264/AVC video coding standard Por: Wiegand, T; Sullivan, GJ; Bjontegaard, G; et ál..IEEE Transactions on circuits and systems for video technology, volumen: 13 Número: 7 Páginas: 560-576 (07/2003)
- [89] ITU-T Recommendation H.264 Infrastructure of audiovisual services – Coding of moving video, Advanced video coding for generic audiovisual services, (04/17).
- [90] Overview of the High Efficiency Video Coding (HEVC) Standard, Sullivan, Gary J.; Ohm, Jens-Rainer; Han, Woo-Jin; et ál, IEEE Transactions on circuits and systems for video technology, Volumen: 22 Número: 12 Páginas: 1649-1668, (12/2012).
- [91] ITU-T Recommendation H.265 Infrastructure of audiovisual services – Coding of moving video, High efficiency video coding, (02/18).
- [92] Krunz M. Makowski M. “Modeling Video Traffic Using $M/G/\infty$ Input Proccesses: A compromise between Markovian and LRD Models”, IEEE Journal of Selected Areas in Communications, pp.733-748, (1998).
- [93] Kush A.: H.264 for the rest of us, available at: <<http://www.images.adobe.com> (2009).
- [94] Y. Morioka. LTE for Mobile Consumer Devices. In *ETSI Workshop on Machine to Machine Standardization*, 2011.
- [95] 3GPP. Service requirements for machine-type communications 3GPP TS 22.368 V14.0.1 (2017-08)

- [96] M. Laner, P. Svoboda, N. Nikaein and M. Rupp, "Traffic Models for Machine Type Communications," ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems, Ilmenau, Germany, 2013, pp. 1-5.
- [97] M. Z. Shafiq et al. A First Look at Cellular Machine-to-Machine Traffic – Large Scale Measurements and Characterization. In *SIGMETRICS/Performance'12, London, UK*, 2012.
- [98] G. Casale, E. Z. Zhang, and E. Smirni. Trace data characterization and fitting for Markov modeling. *Elsevier Perform. Eval.*,67(2):61–79, 2010.
- [99] M. Laner, P. Svoboda, and M. Rupp. Modeling Randomness in Network Traffic. In *SIGMETRICS'12, London, UK*, 2012.
- [100] 3GPP. GERAN improvements for Machine-Type Communications (MTC) (Release 12), TR 43.868, 2014.
- [101] 3GPP. Study on RAN Improvements for Machine-type communications. Technical report, TR 37.868, 2012.
- [102] Gomes J. V. P., Inacio P. R. M., Lakic B., Freire M. M., da Silva H. J. A., and Monterio P. P., "Source traffic analysis", *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2010.
- [103] Hassan, Hassan & Garcia, Jean-Marie & Brun, O. (2018). Generic modeling of multimedia traffic sources.
- [104] C. Cao, Y. Zuo and F. Zhang, "Research on Comprehensive Performance Simulation of Communication IP Network Based on OPNET," 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xiamen, 2018, pp. 195-197. doi: 10.1109/ICITBS.2018.00057.
- [105] A. Y. Privalov and A. Tsarev, "Analysis and simulation of WAN traffic by self-similar traffic model with OMNET," 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), Nicosia, 2014, pp. 629-634. doi: 10.1109/IWCMC.2014.6906429.
- [106] L. Muscariello, M. Meillia, M. Meo, M. A. Marsan and R. L. Cigno, "An MMPP-based hierarchical model of Internet traffic," 2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577), 2004, pp. 2143-2147 Vol.4. doi: 10.1109/ICC.2004.1312897

- [107] Telefonica, Servicio Macrolan (<<http://www.movistar.es/grandes-empresas/soluciones>>) (2018)
- [108] ITU-T Recommendation E.800, Quality of telecommunication services: concepts, models, objectives and dependability planning – Terms and definitions related to the quality of telecommunication services, Definitions of terms related to quality of service, (09/2008).
- [109] Martínez, I., J. García, and E. Viruete. "Modelado de parámetros de tráfico y análisis cuantitativo de QoS para servicios de e-Salud en entornos rurales." *Technology* 16, no. 8 (1995): 52-55.
- [110] B. Cao, M. Shi and C. Li, "The solution of web front-end performance optimization," 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, 2017, pp. 1-5. doi: 10.1109/CISP-BMEI.2017.8302083
- [111] Cisco Systems Inc., Cisco Catalyst 3650 Configuration Guide (<<http://www.cisco.com/c/en/us/support/switches/catalyst-3650-series-switches/products-installation-and-configuration-guides-list.html>>) (2018)
- [112] Ethernet Service Attributes Phase 3, Metro Ethernet Forum, October 2013.
- [113] O. Aboul-Magd, S. Rabie. "A Differentiated Service Two-Rate, Three-Color Marker with Efficient Handling of in-Profile Traffic", RFC 4115, July 2005.
- [114] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang. "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998
- [115] OMNeT Project. [Online]. Available: <<http://OMNeTpp.org/>>
- [116] Eclipse framework, [Online]. Available: <<https://www.eclipse.org/>>
- [117] INET Project. [Online]. Available: <<http://inet.OMNeTpp.org/>>
- [118] E. S. Alias, M. Mukhtar and R. Jenal, "Embedding key user values for the adoption of unified communication and collaboration service," 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI), Langkawi, 2017, pp. 1-4. doi: 10.1109/ICEEI.2017.8312412

- [119] Estudio y guía metodológica sobre ciudades inteligentes. ONTSI (noviembre 2015) <<http://www.ontsi.red.es/ontsi/es/>>
- [120] FiWare IoT Agent Provision API Documentation, <<https://telefonicaiotagents.docs.apiary.io/#>>
- [121] IoT Agent for the Ultralight 2.0 protocol (with HTTP and MQTT transport) <<https://github.com/telefonicaid/iotagent-ul>>
- [122] Cisco CallManager Administration Guide (capítulo 7) <https://www.cisco.com/en/US/docs/voice_ip_comm/cucm/admin/4_0_1/cmcfgr/bccm.pdf>
- [123] Chromy, E.–Diezka, J.–Kovacik, M.–Kavacky, M.: Traffic Analysis in Contact Centers, Proceedings of the 11th International Conference KTTO (2011), 19-24.
- [124] Cisco WP_enterprise_video-bandwidth_es.pdf, <<https://communities.cisco.com/docs/DOC-29683>>
- [125] Constandinos X. Mavromoustakis, George Mastorakis, Jordi Mongay Batalla, Internet of Things (IoT) in 5G Mobile, ISBN 9783319309118, 9783319309132 (online) DOI:10.1007/978-3-319-30913-2