



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona



QUIC sobre IP Multicast

Treball Final de Grau realitzada a
l'Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
per

Enric Perpinyà Pitarch

En compliment parcial
dels requisits per al Grau en
Serveis i Sistemes de Telecomunicacions **ENGINYERIA**

Director: Jorge Mata Díaz
Barcelona, Data 2022



Continguts

Llistat de figures	3
Llistat de taules	3
1 Introducció	9
1.1 Transfons del project	10
1.2 Objectius	10
1.3 Requeriments i especificacions	11
1.4 Pla de feina	11
1.4.1 Estructura de la feina	11
1.4.2 Paquets de feina, tasques i fites	12
1.5 Diagrama temporal (Diagrama de Gantt)	16
2 Estat de l'art sobre les tecnologies utilitzades	17
2.1 Multicast	17
2.1.1 Multicast: IGMP	17
2.1.2 Multicast: la seguretat	18
2.2 Contexte de QUIC	18
2.3 QUIC	18
2.3.1 QUIC: establiment de la connexió	19
2.3.2 QUIC: Migració	19
2.3.3 QUIC: capçaleres més dinàmiques	20
2.3.4 QUIC: Multiplexat	20
2.3.5 QUIC: Control de fluxe a nivell de connexió i de paquets de dades	20
2.3.6 QUIC: Autenticació i encriptació de la capçalera i càrrega útil	21
2.4 QUIC Multicast	21
2.5 Software i llibreries importants	21
2.5.1 VNX: Virtual Network over Linux	21
2.5.2 Wireshark	22
2.5.3 NGTCP2: QUIC unicast	22
2.5.4 NGHQ: QUIC multicast	22
2.5.5 Mencions importants	22
3 Metodologia	23
3.1 Entorn de proves	23
3.2 Disseny del sistema	23
3.2.1 Servidor	24
3.2.2 Client	24
3.3 Metodologia de les proves	24
4 Experiments and results	25
5 Pressupost	26
6 Impacte ambiental	27
7 Conclusions	28
8 Future Work	28
Appendices	29

Llistat de figures

1	Trafic d'Internet	9
2	Diagrama de Gantt del projecte	16
3	Unicast vs Broadcast vs Multicast	17
4	Comparativa establiment connexió entre TCP+TLS i QUIC. Imatge de Cloudflare.	19

Llistat de taules

Abbreviations

API Application Programming Interface

BBC British Broadcast Company

CBR Constant BitRate

CDN Content Delivery Network

ETSETB Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

HLS HTTP Live-Streaming

HTTP Hypertext Transfer Protocol

IETF Internet Engineering Task Force

IGMP Internet Group Management Protocol

IP Internet Protocol

ISP Internet Service Provider

KVM Kernel-based Virtual Machine

RFC Request For Comments

TCGI Transport Control i Gestió d'Internet

TCP Transmission Control Protocol

TDT Televisión Digital Terrestre

WebRTC Web Real Time Communications

Abstracte

El present document descriu i compara l'implementació d'un servidor per retransmetre continguts en temps real (Televisió, radio, etc) fet amb la tecnologia QUIC sobre multicast i un altre fet amb la tecnologia QUIC unicast. L'objectiu és demostrar que l'ús de multicast és millor per distribuir aquest tipus de contingut en termes d'escalabilitat i de sostenibilitat.

Abstracto

El presente documento describe y compara la implementación de un servidor para retransmitir contenido en tiempo real (Televisión, radio, etc) hecho con la tecnología QUIC sobre multicast y otro hecho con la tecnología QUIC unicast. El objetivo es demostrar que es mejor utilizar multicast para distribuir este tipo de contenidos en términos de escalabilidad y de sostenibilidad.

Abstract

The present document describes and compares the implementation of a server dedicated for real-time broadcasting (TV, radio, etc) done with QUIC over Multicast technology and another done with QUIC unicast. The main goal is to prove that multicast is better for distributing this type of content in terms of scalability and sustainability.

Historial de revisió i aprovació

Revisió	Data	Propòsit
0	05/03/2022	Creació del document
1	dd/mm/yyyy	Revisió del document

LLISTA DE DISTRIBUCIÓ DEL DOCUMENT

Nom	correu
Enric Perpinyà Pitarch	enric.perpinya@estudiantat.upc.edu
Jorge Mata Díaz	jorge.mata@entel.upc.edu

Escrit per:		Revisat i aprovat per:	
Data	05/03/2022	Data	dd/mm/yyyy
Nom	Enric Perpinyà	Nom	Jorge Mata
Posició	Autor	Posició	Supervisor

1 Introducció

En els darrers anys, els serveis de contingut en temps real han experimentat un creixement quasi exponencial, sobretot arran de la popularitat de pàgines web com `youtube.com`, `twitch.tv` o sobretot amb l'ús de les videoconferències durant la pandèmia. No obstant això, les tècniques actuals per distribuir aquest tipus de contingut, malgrat haver millorat bastant amb el desenvolupament d'eines com WebRTC, encara representen un problema molt greu tant a la part dels servidors, principalment per les CDN, com també pel volum de tràfic que implica a les xarxes, principalment pels ISP. De fet, s'està tornant una qüestió tecnològica tant greu que molts serveis d'aquests han decidit apujar els seus preus per poder fer front a la demanda.¹

Al mateix temps que hi ha aquest augment de la demanda per serveis nous d'Internet, també hi ha altres que estan intentant canviar de plataforma de distribució tecnològica com la televisió i la ràdio; és a dir, les cadenes de televisió estan intentant transmetre el seu contingut utilitzant Internet en lloc d'antenes i cable. De fet, fer-ho a través d'Internet permet estalviar costos significatius de distribució al mateix temps que permet tenir una retroactivitat i perspectiva més acurada de la quantitat de gent que està veient l'emissió.

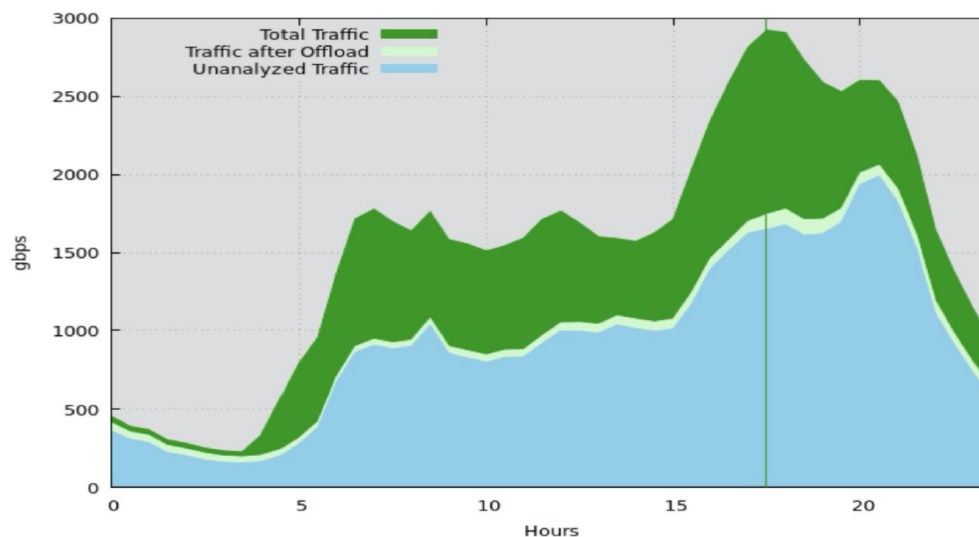


Figura 1: Volum de tràfic en un dia amb la sortida d'un nou videojoc. En blau el tràfic no analitzat, en verd fort el tràfic normal, en verd fluix el tràfic si la descarrega es fes amb multicast. Imatge d'Akamai.

L'any 2015, el tràfic de serveis de vídeo en temps real, com Netflix, YouTube o Twitch, va representar pràcticament el 70% del tràfic.² Normalment, aquest tipus de tràfic va per pics, ja que quan una nova temporada d'una sèrie famosa, quan un youtuber important puja un nou vídeo o s'està fent un esdeveniment en directe la quantitat de gent veient el contingut sol augmentar de manera sobtada sobretot als primers moments. Això sol implicar en general problemes per donar accés a tothom al servei depenent de quanta gent sigui, mentre que al mateix temps altres serveis també es vegin afectats (les CDN solen tenir molts serveis simultàniament distribuïts en microserveis).

¹Notícia a Xataka el 29-11-21: 13 operadoras piden subir precios a Netflix y Youtube.

²Notícia a Hipertextual el 07-12-15: El streaming ya representa el 70% del tráfico de la red

Per ficar dades i posar-nos en context del volum de tràfic, el màxim que s'ha arribat a donar simultàniament per part de la CDN Akamai ha estat un pico de 200 Tbps en març de 2021.³ Suposant un tràfic lineal, per simplificar, ja que normalment s'utilitzen codificacions CBR en directe en temps real, amb una resolució de 1080p el tràfic és de 5 Mbps/connexió i fins a 20 Mbps/connexió en cas de 4k. Llavors, en 1080p suposant aquell tràfic màxim suportat llavors es podrien arribar fins a 40 milions de clients en 1080p i fins a 10 milions en 4k. Si tota Espanya volgués mirar un contingut en directe, seria quasi impossible donar el servei correctament en 4k i molt just en 1080p.

Aquests problemes de subministrament solen ser a causa de que les connexions que es fan són punt a punt; dit d'una altra manera, són **unicast**. Això implica que el servei que pots donar en gran part està limitat al nombre de connexions simultànies que es pot suportar. Malgrat que l'ús d'unicast té grans avantatges com la facilitat en la seguretat, la simplificació en l'arquitectura del programa o que pràcticament tots els dispositius del món ho suporten de manera nativa, també comporta un gran inconvenient.

Si el contingut que has de distribuir als clients és exactament el mateix com en el cas de la televisió o la ràdio, unicast pot arribar a saturar un servei si el volum d'usuaris és molt alt. Amb això en ment, ja a finals dels anys 80 es va publicar una extensió de d'IP, [RFC 988] que solucionava aquest problema: **IP Multicast**. Malgrat no estar tan establerta com IP unicast, sí que la gran majoria de dispositius l'accepten.

Per altra banda, al llarg dels anys, a les connexions unicast normalment han fet ús d'un protocol anomenat TCP. Aquest protocol va ser especialment important en els anys 90 i a principis dels 2000 perquè Internet pogués funcionar i popularitzar-se, ja que les connexions eren lentes i poc fiables. Aquest protocol incorpora una sèrie de mecanismes per intentar pal·liar aquesta qüestió. No obstant això, amb la millora de les connexions i la seva fiabilitat hi ha altres protocols que poden ser més útils i millors en l'Internet que tenim actualment, incorporant funcionalitats que antigament no s'havien plantejat o que no semblaven rellevants. Un protocol que pareix que substituirà TCP és QUIC, ja que millora bastant certs aspectes de TCP com l'establiment de la connexió molt més ràpidament (menys RTTs) o el fet de poder restablir la sessió amb el servidor en canviar de xarxa.

1.1 Transfons del projecte

Aquest projecte es fa des de zero, encara agafa principalment dos projectes de codi lliure d'Internet i un borrador del IETF:

1. *Hypertext Transfer Protocol (HTTP) over multicast QUIC*, de Lucas Pardue, Richard Bradbury i Sam Hurst.
2. *NGHQ*: Llibreria escrita en C que implementa part de l'esborrany de *HTTP sobre multicast QUIC* fins la versió 7.
3. *NGTCP2*: Llibreria escrita en C que implementa el protocol QUIC segons l'estàndard escrit en el RFC 9000.

La idea original ha estat de l'autor, encara que l'enfocament i la metodologia han estat proposades pel professor, Jorge Mata.

Tot el codi utilitzat és de codi lliure desenvolupat per enginyers i que està pràcticament tot allotjat en plataformes online com Github. Tot el codi desenvolupat en aquest projecte està allotjat també al Github i també és de codi lliure.

1.2 Objectius

- Desenvolupar un servidor servidor QUIC unicast i un servidor QUIC multicast per distribuir contingut en temps real.
- Conèixer en profunditat les avantatges i limitacions de la tecnologia QUIC [rfc9000]
- Discutir la possibilitat de fer un perfil concret de QUIC per Multicast.

³Notícia a Addvalue el 24-03-22: Akamai hits huge new 200 Tbps web traffic peak

- Demostrar les limitacions d'unicast en la retransmissions a un gran número de clients i com Multicast pot ser una solució.

1.3 Requeriments i especificacions

Requeriments del projecte:

- Proveir un sistema de transmissió de contingut en directe via unicast i/o multicast.
- Possibilitat l'avaluació del tràfic generat de manera quantitativa i qualitativa.
- Implementar software gratuït i de codi lliure.
- Proveir una guia a l'usuari final per facilitar l'ús i instal·lació dels softwares tant del client com del servidor [Annexe I].

Especificacions del projecte:

- Avaluar el tràfic generat per multicast i unicast en escenaris amb diversos clients escoltant la retransmissió.
- Creació d'un entorn de proves virtual, on el servidor i els clients seràn màquines virtuals Linux.
- Avaluar el tràfic amb eines d'ús extensiu, proveint els filtres necessaris per decodificar els missatges transmesos.

1.4 Pla de feina

En aquesta secció de document es descriu en detall tots els blocs de treball en el qual s'ha dividit la feina i els temps (Diagrama de Gantt) en els que s'ha fet la tasca corresponent.

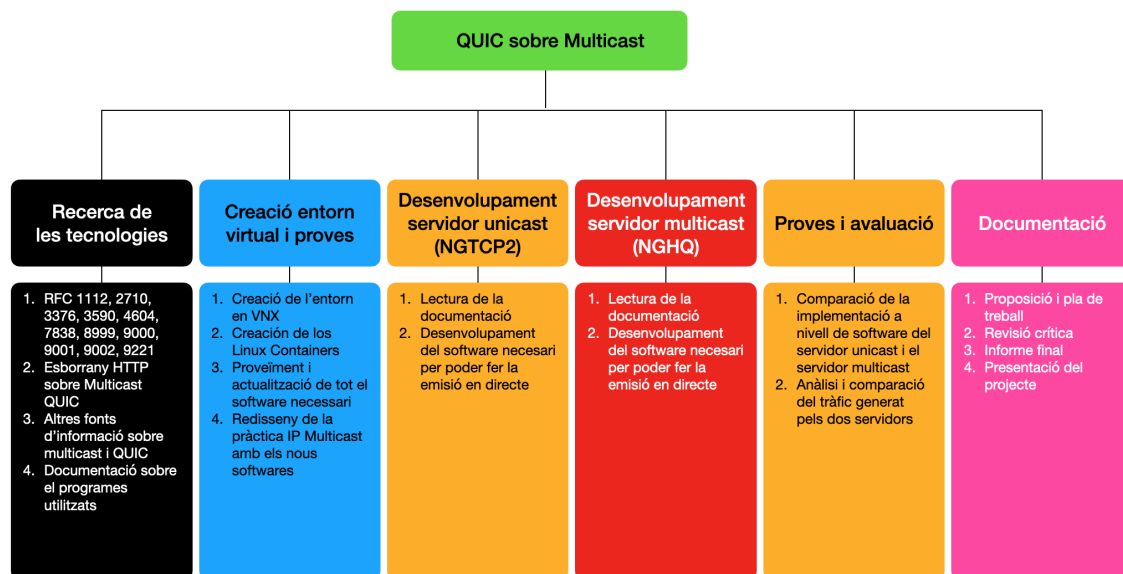
Al llarg del projecte, hi ha hagut diversos canvis en respecte al pla original. Els principals canvis han estat els següents:

- **Lectura de RFCs.** Donat la complexitat del projecte i la profunditat d'aquest s'ha vist que la quantitat de documentació al respecte que s'ha hagut de llegir ha estat molt major de la qual s'esperava, en gran part, ja que el nombre de tecnologies necessaries per poder entendre i desenvolupar una plataforma com la que es demana és molt major del qual es podria esperar a priori. També, a causa que s'utilitzen tecnologies noves com QUIC, hi ha hagut RFCs que han aparegut durant el transcurs com el 9221.
- **Certs programes no existien o no compilaven.** Per fer l'escenari de proves s'ha volgut fer servir un software nou per virtualitzar la xarxa intentant imitar la de la pràctica IP multicast de l'assignatura TCGI del grau d'Enginyeria de serveis i sistemes de telecomunicacions del ETSETB. Es va trobar que les màquines virtuals i el software utilitzat en aquelles màquines és antic i qualcun ja no està disponible. S'ha buscat alternatives al respecte.

El que més ha dificultat i endarrerit el desenvolupament ha estat la lectura de la documentació, ja que era molt més extensa del que s'esperava a priori a més que també entrava en molts detalls que semblaven contradictoris amb la idea, encara que sembla que no. S'ha de pensar que la proposta és una tecnologia que encara no s'ha desenvolupat del tot i encara està en procés, llavors això és més una prova del concepte que una demostració per ficar-ho a producció. Encara s'està lluny d'aquest punt com es veurà al llarg del treball.

1.4.1 Estructura de la feina

[ESBORRANY - FALTA L'IMATGE FINAL]



1.4.2 Paquets de feina, tasques i fites

Projecte: Recerca de les tecnologies	WP ref: WP1	
Element principal: Lectura de la documentació	Pàg. 1 de 6	
Breu descripció: Lectura del RFCs necessaris per tal d'entendre i proposar solucions adequades per crear un perfil de QUIC sobre multicast. Lectura de la documentació necessaria	Data d'inici estimada: 01-10-2021 Data de finalització estimada: 01-11-2021	
	Data d'inici real: 14-09-2021 Data de finalització real: 30-04-2022	
Tasques internes: T1. RFC 1112, 2710, 3376, 3590, 4604, 7838, 8999, 9000, 9001, 9002, 9221 T2. Esborrany HTTP sobre Multicast QUIC T3. Altres fonts d'informació sobre multicast i QUIC T4. Documentació sobre el programes utilitzats	Entregables: Tasca 1 i 2	Dates: incluides en el document

Projecte: Creació entorn virtual i proves	WP ref: WP2	
Element principal: Entorn virtual	Pàg. 2 de 6	
Breu descripció: Creació de l'entorn virtual amb VNX, proveïment de les màquines virtuals amb el software necessari i fer la pràctica d'IP multicast de TCGI per testejar que l'entorn funciona correctament	Data d'inici estimada: 16-09-2021 Data de finalització estimada: 10-10-2021	
	Data d'inici real: 16-09-2021 Data de finalització real: 30-12-2022	
Tasques internes: T1. Creació de l'entorn en VNX T2. Creació dels Linux Containers T3. Proveïment i actualització de tot el software necessari T4. Rediseny de la pràctica IP Multicast amb els nous softwares	Entregables: Presentació al professor de l'escenari montat	Dates: 14-01-2022

Projecte: Desenvolupament servidor unicast (NGTCP2)	WP ref: WP3	
Element principal: Desenvolupament de software	Pàg. 3 de 6	
Breu descripció: Creació d'un servidor unicast que utilitzi el protocol QUIC per poder utilitzar-lo per comparar amb un que utilitzi multicast. També s'ha de fer el client.	Data d'inici estimada: 01-11-2021 Data de finalització estimada: 01-12-2021	
	Data d'inici real: 15-02-2022 Data de finalització real: 05-05-2022	
Tasques internes: T1. Lectura de la documentació T2. Desenvolupament del software necessari per poder per poder fer la emissió en directe.	Entregables: Tasca 1 Tasca 2 Tasca 3	Dates: 15-05-2022

Projecte: Desenvolupament servidor multicast (NGHQ)	WP ref: WP4	
Element principal: Desenvolupament de software	Pàg. 4 de 6	
Breu descripció: Creació d'un servidor multicast que utilitzi el protocol QUIC per poder utilitzar-lo per comparar amb un que utilitzi unicast. També s'ha de fer el client.	Data d'inici estimada: 01-11-2021 Data de finalització estimada: 01-12-2021	
	Data d'inici real: 15-02-2022 Data de finalització real: 05-05-2022	
Tasques internes: T1. Lectura de la documentació T2. Desenvolupament del software necessari per poder per poder fer la emissió en directe.	Entregables: Tasca 1 Tasca 2 Tasca 3	Dates: 15-05-2022

Projecte: Proves i avaluació	WP ref: WP5	
Element principal: Simulació i proves de qualitat	Pàg. 5 de 6	
Breu descripció: Totes les simulacions i avaluacions necessaries per aconseguir proves concloents de en quin cas és millor utilitzar unicast i quin multicast.	Data d'inici estimada: 01-11-2021 Data de finalització estimada: 15-12-2021	
	Data d'inici real: 15-02-2022 Data de finalització real: 05-05-2022	
Tasques internes: T1. Comparació de l'implementació a nivell de software del servidor unicast i el servidor multicast T2. Anàlisi i comparació del tràfic generat pels dos servidors	Entregables: Tasca 1 Tasca 2 Tasca 3	Dates: 15-05-2022

Projecte: Documentació	WP ref: WP6	
Element principal: Documentació	Pàg. 6 de 6	
Breu descripció: Documentació que s'ha d'entregar. Un manual d'usuari per poder configurar ambdós servidors serà inclòs en el document final.	Data d'inici estimada: 15-09-2021 Data de finalització estimada: 15-01-2022	
	Data d'inici real: 20-01-2022 Data de finalització real: 15-05-2022	
Tasques internes: T1. Proposició i pla de treball T2. Revisió crítica T3. Informe final T4. Presentació del projecte	Entregables: Tasca 1 Tasca 2 Tasca 3	Dates: 08-10-21 30-11-21 15-05-22 28-05-22

Objectius

WP#	Tasca#	Títol curt	Objectius/Entregables	Data (setmana)
WP1	T1	RFC 1112, 2710, 3376, 4604, 7838, 8999, 9001, 9002, 9221	Estat de l'art	Inclós en el document
WP1	T1	Esborrany HTTP sobre Multicast QUIC	Estat de l'art	Inclós en el document
WP1	T1	Altres fonts d'informació sobre multicast i QUIC	Estat de l'art	Inclós en el document
WP1	T2	Documentació sobre el programes utilitzats	Guies	Inclós en el document
WP2	T2	Creació de l'entorn en VNX	Entorn virtual montat	Setmanes 37 a 52
WP2	T2	Creació de los Linux Containers	Màquines disponibles per l'entorn virtual	Setmanes 37 a 52
WP2	T2	Proveïment i actualització de tot el software necessari	Màquines provistes del software requerit	Setmanes 37 a 52
WP2	T2	Redisseny de la pràctica IP Multicast amb els nous softwares	Comentaris per actualitzar la pràctica d'IP Multicast amb software actual (annexe)	Setmanes 37 a 52
WP3	T1	Lectura de la documentació (NGTCP2)	Explicació del software a estat de l'art	Setmanes 7 a 19
WP3	T3	Desenvolupament del software necessari per poder fer la emissió en directe	Creació del servidor per enviar la radio IP via unicast	Setmanes 7 a 19
WP4	T1	Lectura de la documentació	Explicació del software a l'estat de l'art	Setmanes 7 a 19
WP4	T3	Desenvolupament del software necessari per poder fer la emissió en directe	Creació del servidor per enviar la radio IP via multicast	Setmanes 7 a 19

WP#	Tasca#	Títol curt	Objectius/Entregables	Data (setmana)
WP5	T4	Comparació de la implementació a nivell de software del servidor unicast i el servidor multicast	Explicació de les avantatges i inconvenients del desenvolupament a l'apartat de resultats del document	Setmanes 7 a 19
WP5	T4	Anàlisi i comparació del tràfic generat pels dos servidors	Gràfica comparativa de la quantitat de tràfic generat per les dues implementacions	Setmanes 7 a 19
WP6	T1	Proposició i pla de treball	Proposició i pla de treball	Setmana 37
WP6	T2	Revisió Crítica	Revisió Crítica	Setmana 48
WP6	T3	Informe final	Informe final	Setmana 19
WP6	T4	Presentació del projecte	Presentació del projecte	Setmana 20

1.5 Diagrama temporal (Diagrama de Gantt)

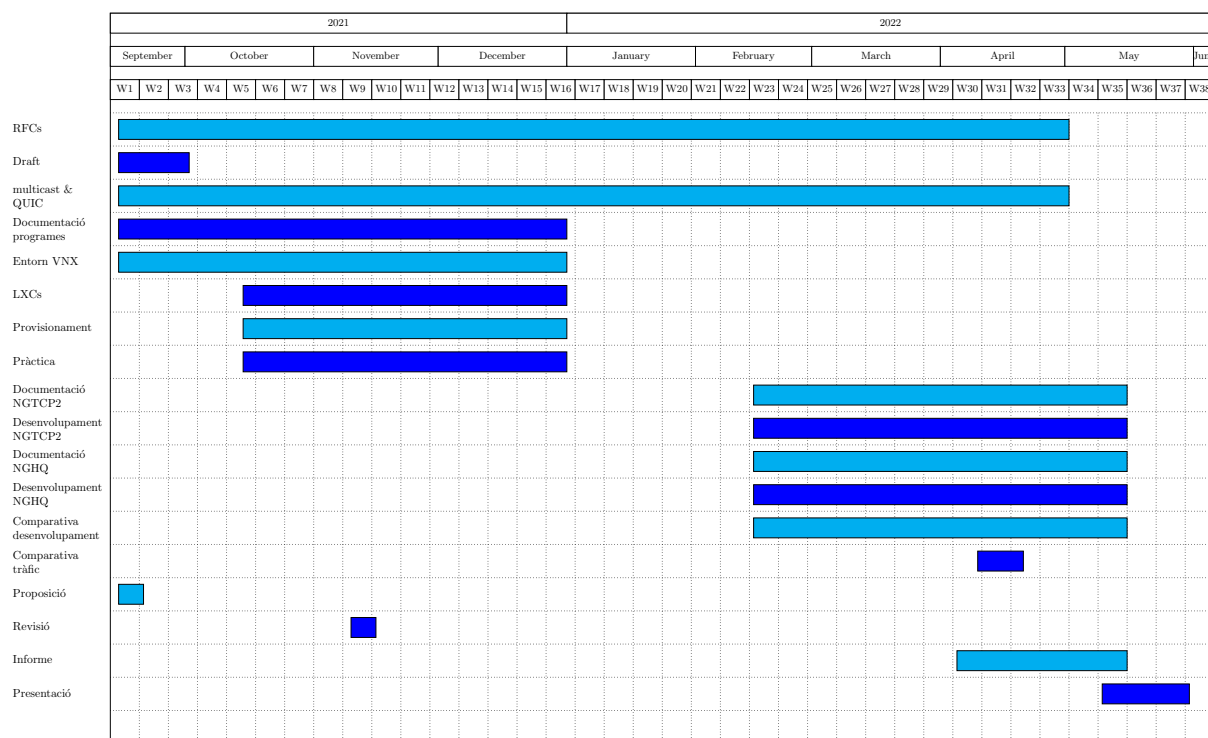


Figura 2: Diagrama de Gantt del projecte

2 Estat de l'art sobre les tecnologies utilitzades

L'interès en les transmissions en directe ha fet que molts esforços d'empreses i grups de recerca es focalitzin en desenvolupar noves tecnologies per fer front a la creixent demanda en termes de qualitat i al mateix temps que donar servei cada cop a més clients. Les tecnologies que permeten aquestes transmissions se centren principalment en dos grups: protocols i codificacions. Això no obstant, en el nostre cas només ens centrarem en els protocols, perquè aquests poden funcionar independentment de les codificacions.

Per tal d'entendre millor el projecte, en els següents apartats s'explicaran els principis i objectius de les tecnologies implicades. A l'annex 1 hi ha un resum extens de l'evolució de les tecnologies implicades fins a arribar la creació de QUIC o l'ús que es fa avui en dia de multicast. Es recomana donar una ullada si hi ha alguna sigla o tecnologia que no s'entén.

2.1 Multicast

IP va néixer com un protocol per interconnectar xarxes d'àrea local amb tecnologies diferents. Aquestes xarxes funcionaven de manera *unicast*, un a un, o *broadcast*, un per a tots. Malgrat això, l'any 1988 es va publicar el RFC 988 anomenat [**Host Extension for IP Multicasting**]. Aquest proposava fer una extensió de la tecnologia IP per permetre la connexió d'un emissor a molts receptors, però no tots, només els que volguessin rebre la informació d'aquest.

Per fer una analogia de multicast amb el nostre entorn quotidià, es podria dir que funciona com la TDT o la ràdio. Es pot pensar com una antena que emet el senyal i si vols rebre el senyal únicament ficat el canal o la freqüència a la qual aquesta emet per poder rebre i veure o escoltar la transmissió, però si algú no la vol rebre no la pot ignorar.

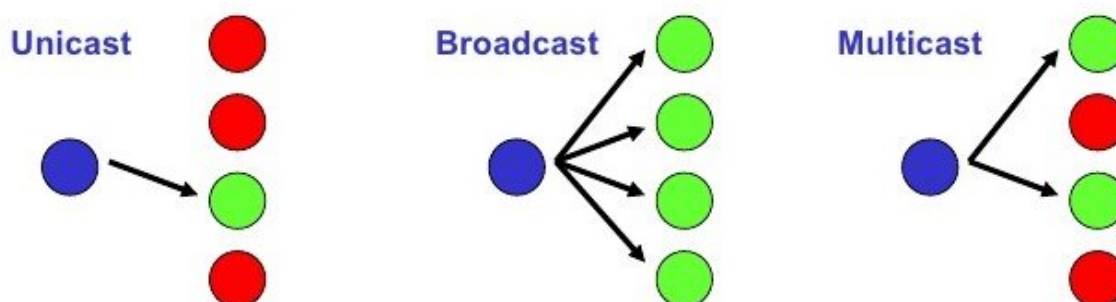


Figura 3: En blau l'emissor de la senyal, en verd els usuaris que reben la senyal i en vermell els que no. Imatge de pc-solucion.es

Dins la tecnologia IP hi ha un rang d'adreces destinades a aquesta tecnologia. El rang és la classe D que va des de **224.0.0.0** fins **239.255.255.255**.

Encara que a priori sembla una idea força senzilla, en realitat, és molt complicada, ja que té diversos problemes associats, encara que la majoria actualment ja tenen solució excepte un que s'esmentarà al final i és el seu gran problema a l'hora d'implementar-lo extensament. Normalment, les qüestions d'IP multicast se solen derivar a capes superiors.

2.1.1 Multicast: IGMP

Per poder dir a un router que pot arribar a rebre l'emissió que es vol rebre aquesta es fa ús d'un protocol anomenat IGMP o Internet Group Management Protocol. La idea d'aquest protocol és implementar un model de subscripció al senyal en el qual mitjançant un paquet que s'envia al router a una distància 1 s'informa que es vol rebre l'emissió d'una direcció destinatària i un port concrets. Aquest protocol permet també dir que la connexió ha d'arribar d'un lloc concret (*source specific*).

2.1.2 Multicast: la seguretat

Un dels majors problemes encara no resolts de multicast és l'encriptació de les comunicacions. El mètode actual per xifrar les comunicacions es basa a canviar la clau d'accés cada poc temps transferint-la via un canal alternatiu (via *unicast*). Malgrat que aquesta solució a priori sembla suficient, si aquesta clau es comparteix a un usuari que no ha de poder veure el senyal (televisió de pagament per exemple) llavors la transmissió multicast no és segura. És més, que es comparteixi la clau per qualsevol agent escoltant el senyal pot implicar altres riscos de seguretat per la transmissió com la suplantació d'identitat per introduir codi maliciós.

2.2 Contexte de QUIC

Transmission Control Protocol (TCP) és un protocol dissenyat per enviar un flux de dades entre dos punts. Les dades són trameses pel sistema TCP que s'encarrega d'assegurar que les dades arribaran a l'altre extrem exactament iguals o que en el cas que no fos així indicar que hi ha un error a la connexió.

Per aconseguir això, TCP parteix les dades en petits paquets i afegeix una sèrie de capçaleres. Aquestes dades extres inclouen un número de seqüència que s'utilitza per detectar quan un paquet s'ha perdut o arribar tard (Out-of-order) i una suma de comprovació s'utilitza per detectar que hi ha hagut un error a les dades. Quan algun problema ocorre, TCP automàticament usa un algorisme que es diu Automatic Repeat Request (ARQ) en el que es notifica a l'emissor que reenvii el paquet perdut o mal transmès.

En la majoria d'implementacions, quan hi ha TCP detecta un error parerà la transmissió fins que l'error s'hagi resolt o consideri que la connexió ha fallat. En cas de fer servir una connexió multiplexada en diversos fluxos de dades, com en el cas de HTTP/2, tots es veuran afectats i es pararan encara que només hi hagi hagut un error en un flux. Per exemple, si hi ha un error carregant una imatge d'una pàgina web, tota la resta de fluxos hauran d'esperar fins que la qüestió s'hagi solucionat. Això es coneix com la qüestió d'encapçalament de línia o head-of-line blocking.

TCP està dissenyat com una canonada de dades en la qual no entén el que s'està enviant. Si es requereixen necessitats addicionals com encriptació (TLS), aquests s'han de fer des de capes superiors a TCP, utilitzant TCP per comunicar-se amb l'altre extrem que utilitza software semblant. Cadascun d'aquests protocols necessita el seu propi *handshake*. Normalment, això implica que hi hagi diversos intercanvis de paquets de sol·licitud i resposta per establir la connexió fins que aquesta finalment s'ha establert. Això pot arribar a ser un molt greu problema en sistemes de comunicacions amb alta latència, ja que pot significar sobrecàrrega de dades de control en el transcurs de la connexió.

2.3 QUIC

QUIC és un nou protocol d'Internet que es diu que canviarà com funciona actualment Internet. Daniel Stenberg, creador de l'aplicació curl i un dels contribuïdors a l'estàndard de QUIC, va dir que fins ara HTTP s'ha fet sobre TCP i això ha implicat grans limitacions, però amb la vinguda d'aquest nou protocol (QUIC), les regles del joc podran canviar millorant les velocitats i altres aspectes de l'Internet actual.

A principis de la dècada passada, a Google varen començar a platejar un nou estàndard pensat per substituir TCP i al mateix temps simplificar i millorar coses com la seguretat o que l'establiment del canal fos el més ràpid possible inclús arribant al punt que en la primera resposta el servidor pogués donar informació útil al client. Amb aquesta idea en ment varen començar a treballar en el que anomenaren Quick UDP Internet Protocol o QUIC.

Avui en dia es tracta d'un estàndard redactat publicat en els RFC 8999 (Version-independent Properties of QUIC), **RFC 9000** (QUIC: A UDP-Based multiplexed and Secure Transport), RFC 9001 (Using TLS to Secure QUIC) i RFC 9002 (QUIC Loss Detection and Congestion Control). L'estàndard de QUIC de Google es coneix com gQUIC a dia d'avui i està pràcticament deixat. Es recomana l'ús de l'estàndard proposat per l'IETF que va sortir en maig de 2021.

A l'inici aquest protocol estava pensat per fer feina pràcticament en la capa de transport (capa L4 del model OSI). Això no obstant, varen trobar un greu problema amb com s'havien implementat els dos protocols predominants fins al moment, TCP i UDP: estaven implementats en la majoria de nuclis dels

sistemes operatius. Això implica que el desplegament o actualització d'un protocol sigui extremadament lent i pugui tardar més d'una dècada. Sobretot també s'ha de pensar que hi ha molts firewalls que no accepten tràfic que no sigui d'aquests dos protocols.

Al veure, aquesta situació varen optar per una solució en una capa superior, la qual normalment està implementada en l'espai d'usuari i sol ser més fàcil d'actualitzar a priori⁴. Donat que UDP és un protocol tan senzill i que delega pràcticament totes les tasques de retransmissions o control de flux a capes superiors, varen decidir fer ús d'aquest per assegurar la interoperabilitat en els nodes intermedis que no entenen aquest nou protocol. Aconseguint saltar aquesta gran barrera, ja varen poder començar a treballar en els aspectes d'aquest nou protocol.

2.3.1 QUIC: establiment de la connexió

Un dels aspectes rellevant de QUIC és que utilitza menys paquets per arribar a establir la connexió inicialment. La filosofia per poder arribar a aconseguir això és que es realitzi el *handshake* tant en l'àmbit de connexió (imitant TLS) com en l'àmbit de seguretat (TLS).

Si comparen el *handshake* de QUIC, que inclou implícitament el *handshake* de TLS 1.3, amb el *handshake* de TCP + TLS 1.2 podrem observar que el nombre de paquets intercanviats entre els dos extrems és molt major en el segon cas. El problema resideix en què en mirar als dos protocols com dos protocols diferents, llavors l'establiment de la connexió primer s'ha de fer per TCP i després per TLS. A QUIC es proposa fer els dos al mateix temps.

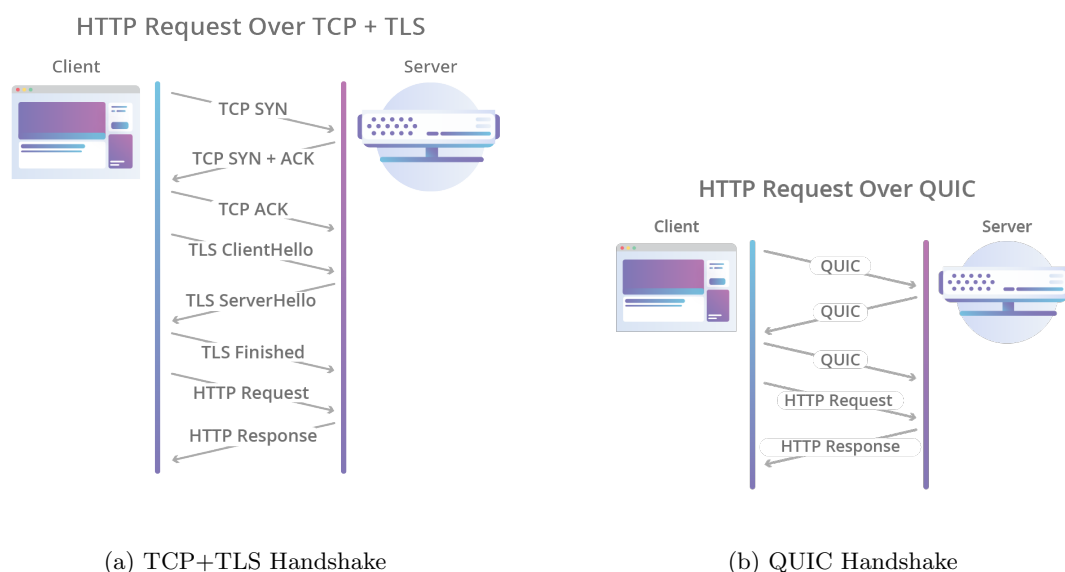


Figura 4: Comparativa establiment connexió entre TCP+TLS i QUIC. Imatge de Cloudflare.

2.3.2 QUIC: Migració

A QUIC al mateix temps s'introdueix un concepte molt interessant que és el **Connection ID**. Bàsicament, es tracta d'un tiquet l'identifica la teva connexió. Això comporta grans avantatges com la fàcil migració de connexió que a continuació s'explica, que en el mateix port puguin córrer diverses aplicacions, principalment a la banda del client o ajudar a evitar atacs d'amplificació (s'explica en detall en el RFC 9000).

⁴En teoria hauria de ser així, però per exemple en el cas de TLS 1.2 va ser una qüestió molt gran la seva actualització. Recomanat l'article de Cloudflare al respecte. Està a la bibliografia

Fins ara, quan s'estableix la connexió, sempre es parlava d'una tupla per identificar a l'altre extrem: una IP i un port. Aquest plantejament de baix nivell funciona molt bé per dispositius que no canvien de xarxa o IP. Tanmateix, amb la vinguda dels dispositius mòbils i sobretot dels smartphones aquest concepte d'identificar a l'altre extrem amb la IP i el port pot arribar a ser un problema, ja que aquest canvien de xarxa mòbil a xarxa wifi bastant sovint. Això no permet reprendre una connexió a on s'havia deixat si s'estaven enviant dades havent de tornar a fer el *handshake* un altre cop. Amb QUIC i el plantejament d'identificadors de connexió o *Connection IDs* canvia completament.

A QUIC s'introdueix el concepte de **0-RTT**; és a dir, que encara que és canvi de xarxa la connexió es pot reprendre a on s'havia deixat evitant fer tants de *handshakes* i pràcticament transmetre dades des de la primera resposta si s'ha establert una connexió anteriorment. Per aconseguir això, tant el servidor com el client ha de mantenir una sèrie de paràmetres de transport (principalment de TLS) i el mateix *Connection ID* tant en la banda del servidor com del client (detalls a la secció 7 de [QUIC-Transport]).

2.3.3 QUIC: capçaleres més dinàmiques

A QUIC hi ha dos tipus principals de paquets en termes de mida i el que duen⁵. Per una banda, estan els paquets amb una capçalera llarga o *long header* que són utilitzats principalment durant els *handshakes* o altres paquets especials. Aquest tipus de paquets inclouen el *SCID* (*Source Connection ID*), el *DCID* (*Destination Connection ID*), altres paràmetres de transport, etc. Aquests paquets també poden dur càrrega útil de capes superiors, encara que aquest tipus no és el més adequat per connexions llargues.

Per altra banda, hi ha un altre tipus de paquets que tenen una capçalera curta o *short header* que sí que són emprats després que la connexió s'ha establert i es transfereixen les dades de manera més extensa. Aquest tipus de paquets només duen el *DCID* i el número de paquet pràcticament, ja que són les úniques dues dades fonamentals perquè la connexió pugui funcionar.

Si s'ha d'enviar un tipus de trama especial, normalment s'envia a través dels fluxos o *streams* que es generen durant el *handshake* o en fases posteriors. Sol haver-hi un canal destinat a aquest tipus de trames.

2.3.4 QUIC: Multiplexat

Ja des de HTTP/2 s'ha plantejat el concepte de multiplexat en l'enviament de dades. Per ficar context, només la pàgina inicial de la gran majoria de pàgines web que podem visitar a través de cercadors estan fetes a base de molts arxius, ja sigui l'arxiu HTML (.html), l'arxiu per decorar (.css), l'arxiu per afegir interactivitat (.js) o fotos i altres (.jpg, .mp3, .ts, etc.). En veure aquesta situació ja en el seu moment es va plantejar que descarregar paral·lelament aquests arxius podia accelerar en gran manera la velocitat de càrrega i descarregar al servidor de feina (sobretot si apliquem també *server push*).

Encara que aquesta idea al final si ha millorat molt les velocitats de càrrega, xoca frontalment amb el concepte d'un únic flux de dades que proposa TCP. En el cas que es perdi un paquet d'un flux de dades HTTP/2, llavors tots els fluxos han d'esperar al fet que aquest paquet es recuperi fent que la millora velocitat que guanyàvem abans es perdi, ja que s'activa el mecanisme de ARQ de TCP. Aquest fenomen s'anomena *Head-of-line blocking*.

Amb HTTP/3 (HTTP sobre QUIC), com a la capa de transport s'utilitza UDP, llavors quan es perd un paquet d'un flux de dades, els altres fluxos no es veuen interromputs, ja que no hi ha mecanismes en capes inferiors com ARQ com si li passa a HTTP/2 amb TCP. De fet, es pot pensar en cada flux de QUIC com un flux TCP independent que incorpora mecanisme de recuperació i validació de paquets.

2.3.5 QUIC: Control de fluxe a nivell de connexió i de paquets de dades

Donat que es realitza un multiplexat individual per cada *stream* llavors el control de flux s'ha de fer per cada *stream* individualment, encara que també es comptabilitza a escala general (mirar més especificacions en la secció 4 de [QUIC-transport]).

⁵S'adjunta el següent enllaç on s'expliquen en una mica de detall tots els tipus: <https://gist.github.com/martinthomson/744d04cbcec9be554f2f8e7bae2715b8>

A manera de resum molt ràpid, les dades s'envien en trames STREAM i quan l'aplicació de capa superior no vol enviar més dades temporalment s'utilitza STREAM_DATA_BLOCKED per mantenir la connexió viva. Els fluxos poden ser unidireccionals, des del client al servidor o viceversa, o bidireccionals. Els *streams* poden tenir prioritats i és important respectar els estats d'aquests perquè la connexió funcioni (es recomana llegir l'apartat 3 de [QUIC-transport] per més detalls).

2.3.6 QUIC: Autenticació i encriptació de la capçalera i càrrega útil

Finalment, per acabar de contar entre altres coses interessants que afegeix QUIC, ens falta parlar dels seus mecanismes d'autenticació i encriptació tant de la capçalera com de la càrrega útil. Fins ara, quan s'envia un paquet via TCP, únicament s'encriptava a la càrrega útil i es deixava a capes superiors (TLS) aquesta feina. El fet de poder encriptar la majoria de la capçalera i verificar que el receptor és qui realment diu ser fa que atacs del tipus amplificació siguin especialment complicats.

2.4 QUIC Multicast

Com s'ha pogut veure fins ara, QUIC és un protocol destinat per ser unicast completament ja que inclou principalment mecanismes de control de fluxe o retransmissió del paquet entre d'altres. No obstant, es pot arribar a fer un perfil concret de QUIC pel cas de multicast, ja que encara que no s'aprofita tant sí que és útil per evitar les *middleboxes* com routers o firewalls a no tallar el tràfic multicast.

La avantatge principal que pot oferir fer un perfil concret de QUIC per multicast és que aquest protocol ja està establert com un estàndard. Un dels majors problemes de les xarxes de comunicacions actuals resideix a la hora fer actualització de software als *middleboxes* i a vegades la mala implementació de certs estàndards (recordar el cas de TLS 1.1 a 1.2 i TLS 1.2 a 1.3). Llavors, el plantejament de fer un estàndard des de 0 normalment no sol ser una bona idea en general.

Per altra banda, el ús de QUIC podria permetre dues avantatges més. Per una part, el multiplexat pot ser força interessant sobretot per *streamings* de video, ja que en un *stream* es pot enviar la imatge, en un altre l'audio i en un altre els subtítols per exemple. Per altra part, el disposar d'un rang tan gran de *Connections IDs*, 2048 bits, permet destinar una part d'aquest rang per direccions multicast i altra part per els homònims en unicast per senyalització; no obstant, aquesta última proposta també es pot fer amb Session IDs com amb NGHQ i proposa el esborrany de Multicast over QUIC, aprofitant el ús de Connection ID 0 (mirar [QUIC Transport]).

De manera molt breu, alguns aspectes tècnics a tenir en compte sobre aquest perfil és que es recomana tenir un canal alternatiu on es faci el *handshake*, la retransmissió de paquets perduts i demés. la connexió multicast únicament ha d'enviar paquets unidireccionalment del servidor al client en format *short header*. Tots els clients han de tenir el mateix *Connection ID* de destí.

Actualment només hi ha un esborrany de l'IETF escrit per Lucas Pardue, Sam Hurst i Richard Bradbury que es diu HTTP sobre QUIC multicast. No obstant, a partir de març de 2022, també l'enginyer de Akamai Jake Holland està escrivint un esborrany al respecte, encara que en aquest cas, actualment està en una fase inicial.

2.5 Software i llibreries importants

2.5.1 VNX: Virtual Network over Linux

És un software general *open-source* de virtualització dissenyat per construir xarxes de proves virtuals automàticament. Permet la definició i desplegament automàtic de escenaris de xarxa construït amb màquines virtuals de diferents tipus (Linux, Windows, FreeBSD, Olive o routers Dynamips, etc) interconnectats seguint una topologia definida per l'usuari, permetint inclús connectar aquesta a xarxes externes.

Esta desenvolupat pel professor David Fernández de l'Universitat Politècnica de Madrid (UPM).

2.5.2 Wireshark

Wireshark és un programari lliure i de codi obert amb la funcionalitat d'analitzador de paquets de xarxes de comunicació. Wireshark s'empra per a solucionar problemes en xarxes, desenvolupament i anàlisi de programari i tasques educatives. Originàriament s'anomenava Ethereal i va ser reanomenat Wireshark el maig del 2006 per causes comercials.

Wireshark és molt similar al programari tcpdump, però amb una interfície gràfica d'usuari i més opcions de menú per a ordenar i filtrar. Es pot emprar per analitzar tot tipus de protocols, des de nivell capa 2 com Ethernet fins a nivells superiors com HTTP.

2.5.3 NGTCP2: QUIC unicast

És una llibreria escrita en C++ que permet enviar missatges unicast utilitzant el protocol QUIC. Està basada en el RFC 9000 [QUIC Transport] i està implementat casi tot l'estàndard. Té una sèrie d'exemples bastant útils encara que complicats d'entendre a primer cop d'ull. Als annexes hi ha un exemple d'ús del servidor i el client.

2.5.4 NGHQ: QUIC multicast

És una llibreria desenvolupada en codi C que permet enviar paquets QUIC sobre multicast. Està basada en el part de l'esborrany 7 d'HTTP sobre Multicast QUIC. Una llibreria una mica complicada d'utilitzar però s'han fet petites modificacions per poder adaptar-la al projecte.

Està escrita per Sam Hurst, Lucas Pardue i Richard Bradbury, enginyers de la BBC.

2.5.5 Mencions importants

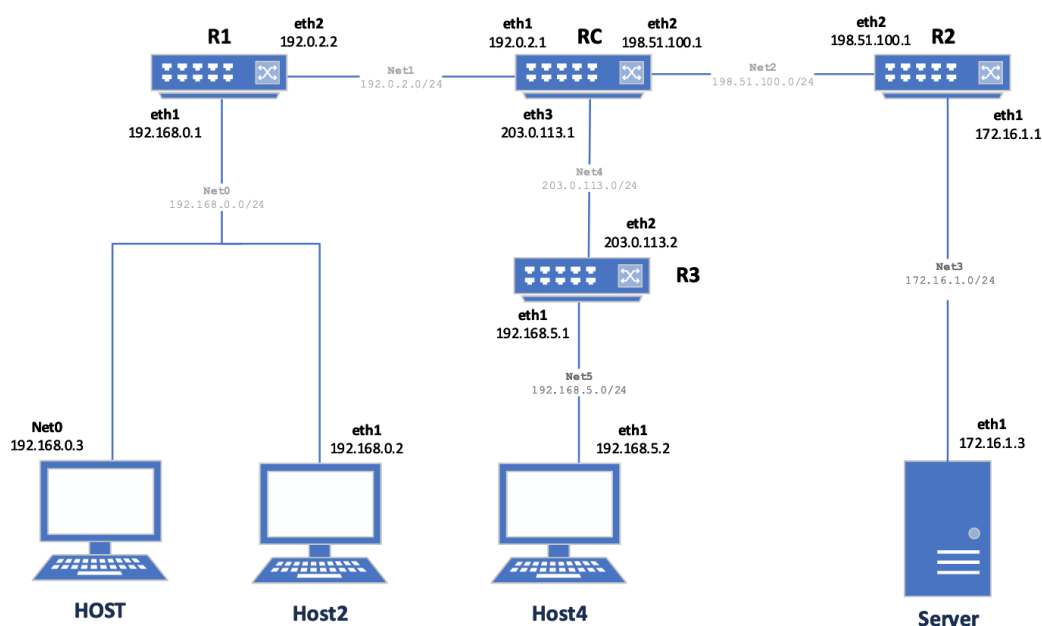
- **ffmpeg**: és una eina que permet modificar, retallar, enviar, etc fluxes de videos principalment.
- **smcroute**: Static Multicast Route és una eina que permet crear rutes estàtiques per enrutar el tràfic multicast.
- **mnc**: versió multicast de la clàssica eina Netcat, encara bastant més limitada.
- **mcsender**: emet una senyal multicast cada segon. Serveix per enviar tràfic multicast i comprovar que funciona l'enrutament.
- **vlc**: Visualitzador de video.
- **lxc**: Linux Containers serveix per crear una màquina virtual però sense haver de crear el kernel. Més eficient que una màquina KVM.

3 Metodologia

3.1 Entorn de proves

L'entorn de proves s'ha creat amb el software VNX. L'escenari està basat en l'entorn de la pràctica d'IP Multicast de l'assignatura de TCGL, ja que és un entorn conegut, además de ser suficient per fer la demostració i ser lleuger a l'hora de fer modificacions i montar-ho l'ordinador base.

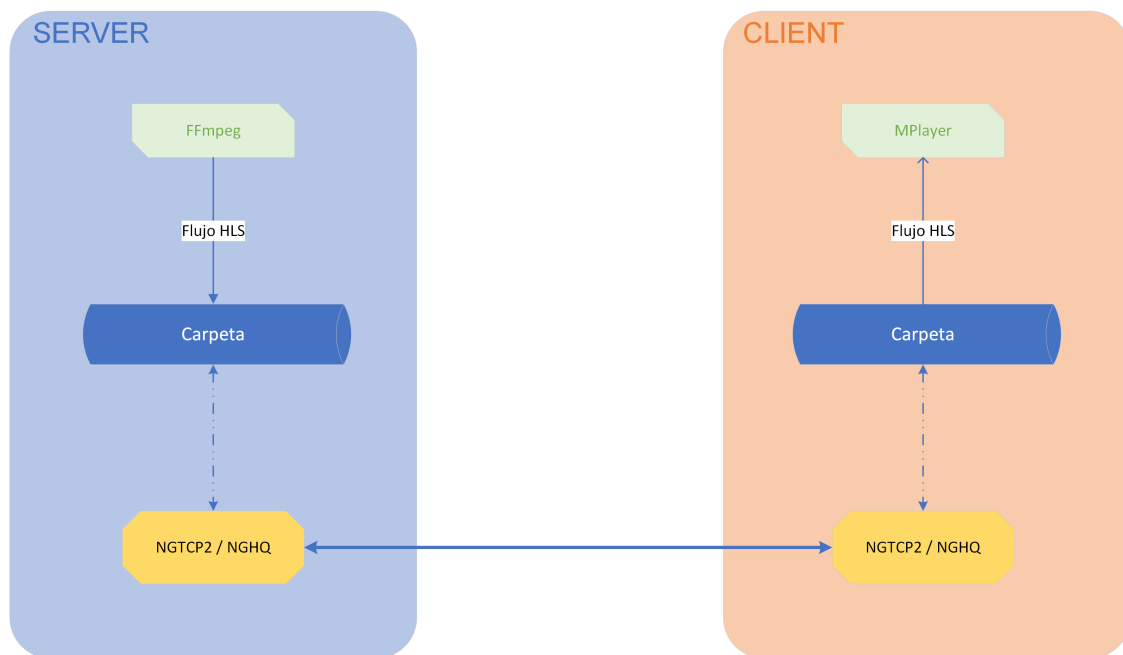
Totes les màquines virtuals de l'entorn son una rèplica d'un LXC amb el sistema operatiu Ubuntu 18.04 LTS. Aquest està provisionat amb tot el software necessari tant pels routers com pels hosts. En el cas de precisar un software en desenvolupament, com pot ser els scripts per fer correr el servidor i els clients, llavors també es poden copiar al montar l'escenari (VNX permet copiar carpeta específiques al terminal indicat).



La topologia de xarxa consisteix en 3 hosts, 2 virtuals i un real (HOST), que rebran la retransmissió; 4 routers que s'encarreguen de distribuir el contingut segons és necessari; i 1 servidor que emetrà el contingut tant per via multicast com per via unicast.

3.2 Disseny del sistema

S'ha dissenyat el sistema perquè sigui el més simple possible, fent que l'estructura tant del servidor com del client siguin pràcticament semblants tant pel cas unicast com pel cas multicast. De cara a l'usuari final s'ha volgut que l'utilització del servidor unicast o del multicast sigui pràcticament igual; és dir, que sigui igual utilitzar un o l'altre a nivell d'API.



3.2.1 Servidor

Per la banda del servidor, primer de tot es genera un fluxe de contingut amb *ffmpeg* que es guarda en una carpeta. Aquest fluxe és un video en bucle reproduït en temps real codificat en HLS⁶. Una vegada s'ha començat a crear el fluxe, aquest es detecta per un script creat en Python que anirà indicant continuament quan s'ha modificat l'arxiu de senyalització (.m3u8) i quin és l'últim troç de fluxe de dades creat (.ts).

En el cas d'unicast, aquest ha de ser demanat pel client (GET) i en el cas de multicast aquest serà enviat directament a l'usuari (server Push).

3.2.2 Client

Per la banda del client, el fluxe creat pel servidor serà rebut (multicast) o demanat (unicast). Aquest s'emmagatzemarà en una carpeta segons vagin arribant. Per poder visualitzar el video live-streaming mentre va arribant es pot inicialitzar un reproductor com pot ser Mplayer. S'ha de pensar que l'arxiu de senyalització (.m3u8) està pensat perquè pugui ser modificat en temps real (on-the-fly), fet que permet de cara a l'usuari final veure un fluxe continu.

3.3 Metodologia de les proves

Per una part s'analitza la quantitat de tràfic generat en el cas multicast i en el cas unicast (prova quantitativa) i, per altra part, s'analitzarà la qualitat del tràfic rebut mirant el video rebut (prova qualitativa). En els dos casos es faran les corresponents taules comparatives i comentaris pertinents.

En el nostre cas, per falta de potència i perquè és un entorn simulat i no un entorn real, no es faran proves de quin és el límit d'usuaris en el servidor unicast. Per la teoria, i pràctica, no té sentit mirar quin és el límit d'usuaris en el cas multicast.

⁶És l'estàndard de codificació de video i audio utilitzat per transmetre contingut en temps real a Internet.

4 Experiments and results

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

5 Pressupost

Donat que de per sí el sistema és purament software i recerca, els costos directes de sistema es podria dir que són 0.

Si tenim en compte el cost de la mà d'obra per fer el disseny, desenvolupament i proves del software durant els 5 mesos per fer un estimació, llavors els costos ascendrien fins als 4500€.

Personal	Preu/Hora	Hores setmanals	Nº Mesos	Sou
Enginyer junior (disseny i desenvolupament)	9€	100	5	4500€
TOTAL				4500€

No obstant, si també tenim en compte que hem virtualitzat el servidor, en un entorn real, aquest tindria també un cost bastant significatiu. Donat que el servidor ha de tenir un accés a disc ràpid i es vol que sigui fiable, el preu d'un servidor d'aquestes característiques pot arribar als 1500€. El millor seria tenir aquest servidor redundat per si acàs s'averia o altres temes, pràctica molt habitual. A n'això se li ha d'afegir el preu per col·locarlo en un centre de dades perquè estigui segur.

Materials i serveis	Preu/Mes	Preu/any
2 Servidors amb SSD 480gb i 8gb de RAM	240€	2880€
Housing	100€	1200€
TOTAL		4080€

El preu total és de **8580€**.

6 Impacte ambiental

Actualment, les emissions de carboni generades per dispositius electrònics ronda el 3.6% de les emissions totals. Només la distribució del tràfic en temps real unicast, com es fa avui en dia, representa unes 300 milions de tonelades de carboni a l'atmosfera, que és el 1%. És més, amb l'augment de la distribució de contingut en temps real aquesta mala dada no farà més que incrementar si es continua fent com avui en dia.

En el cas d'utilitzar multicast, agafant dades de una CDN com Akamai⁷, podriem reduir en casi en un 20% el tràfic generat en un dia normal i fins un 50% en un dia especial com potser un partit de final del mundial de fútbol. És dir, en el cas pitjor podriem reduir casi un 20% les emissions causades per la distribució de contingut en temps real passant de generar 300 milions de tonelades de CO₂ a només 240 milions de tonelades.

⁷[Multicast for the Web]

7 Conclusions

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

8 Future Work

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

Appendices

Appendices may be included in your thesis but it is not a requirement.