International Journal of Digital Information and Wireless Communications (IJDIWC) 6(3): 173-183
The Society of Digital Information and Wireless Communications, 2016 ISSN: 2225-658X (Online); ISSN 2412-6551 (Print)

Encapsulation of Real Time Communications over Restrictive Access Networks

Rolando Herrero
Northeastern University
360 Huntington Avenue, Boston, MA 02115, USA
r.herrero@northeastern.edu

ABSTRACT

The mechanisms of firewall traversal used to overcome the problems that impact Real Time Communications (RTC) in restrictive access networks introduce, among other impediments, excessive latency that results in degraded media quality when frames are dropped by playout buffers running at the application layer. In this paper we summarize the main mechanisms that are used to traverse firewalls through a comparative analysis that concludes with an extensive overview of media encapsulation technologies. One drawback of tunneling, however, is that it involves stream based transport, that is incompatible with datagram based media. Under this scenario and to evaluate how both, speech and video, are ultimately affected by latency and loss typical of mobile networks we use state-of-the-art quality metrics. Moreover, in order to mitigate these negative effects in the context of tunneled traffic, we introduce and assess two separate methods that are optionally applied on top of regular stream based encapsulation.

KEYWORDS

RTC, Tunneling, Encapsulation, Firewall Traversal, Security

1 INTRODUCTION

RTC mechanisms used for transmission of both, speech and video, are an integral part of the backbone of IP Multimedia Subsystem (IMS) networks [1] that rely on different technologies intended to provide reliable and secure real-time delivery of media. One of the most important methods used to accomplish these goals is by means of the Internet Engineering Task Force (IETF) Real-time Transport Protocol (RTP) [2] which is an application layer protocol typically running on top of User Datagram Protocol (UDP). This protocol provides some minimal sequence and timing control but lacks of data integrity protection. Note that because media is time sensitive, data reliability schemes like those that exist on top of Transport Control Protocol (TCP) introduce latency constrains that make their use not as effective.

Firewalls introduce intentional and non-intentional limitations that usually prevent UDP based traffic from traversing them. In general, the efficient traversal of firewalls has been widely studied and several mechanisms have been proposed to overcome these limitations. There are basically two approaches; (1) through relaying and (2) through transport concealment. Relaying involves re-routing media packets so they avoid firewalls by circumventing them through a series of relying servers specially laid out to accomplish this task. Transport concealment, on the other hand, involves changing the transport protocol and port of the media packets so they are compatible and can traverse specific firewalls. Tunneling is the preferred method of concealing transport because the media packet including network and transport layers is encapsulated on top of a, typically TCP, firewall-friendly transport protocol and port. This scenario results on packets that have two sets of transport and network layers; an outer or external one and an inner or internal one.

Each type of firewall traversal method has its own challenges; for example, media relaying requires additional servers distributed throughout the network that impact in the network topology as well as introduce undesired latency caused by the longer path each packet must traverse. Moreover,

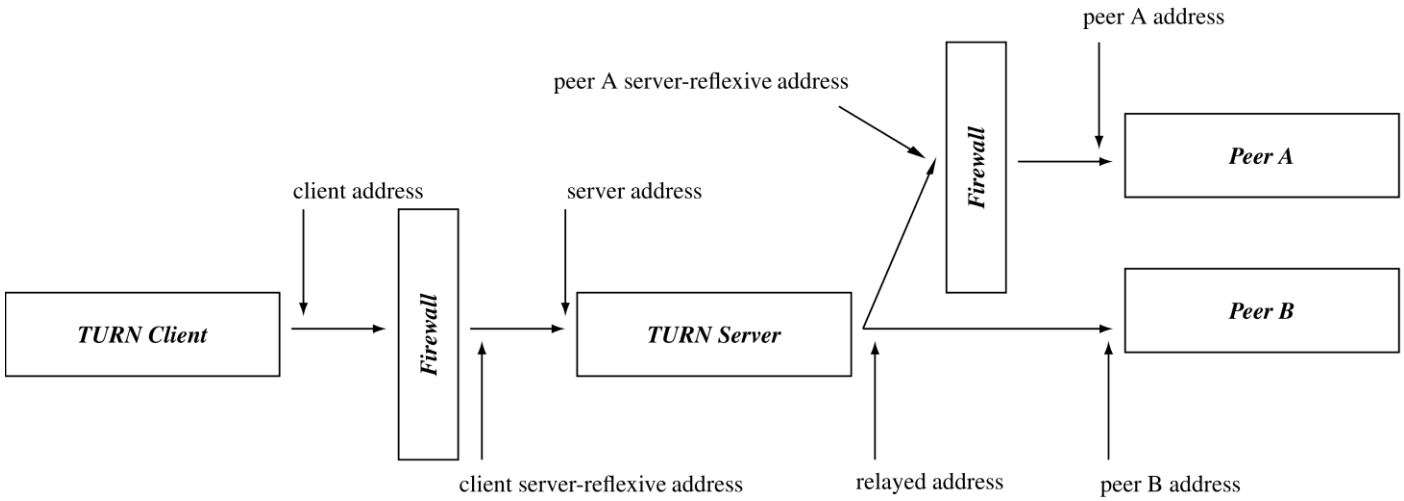

Figure 1: Traversal Using Relays around NAT

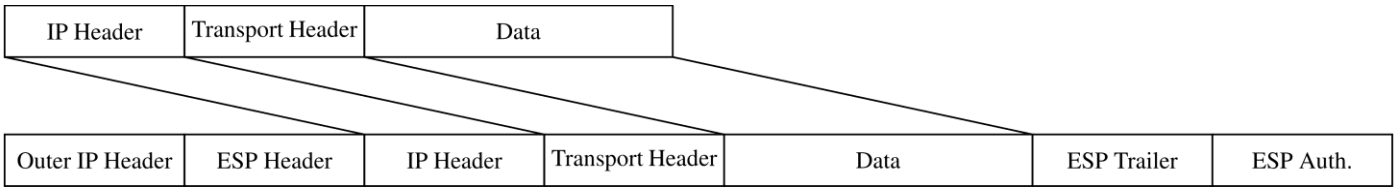

Figure 2: IPSec Tunnel Mode

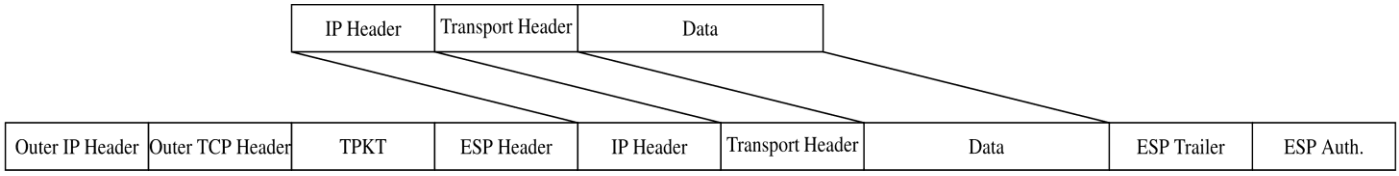

Figure 3: Enhanced Security Gateway (eSEG)

since RTP is negotiated by the Session Initialization Protocol (SIP) [3], a signaling protocol that doesn't suffer the same constrains RTP does, media relying information must also be negotiated requiring application layer changes that lead to increased computational complexity throughout the different network elements of the deployment.

Similarly, media encapsulation introduces a different set of problems; although it is transparent to the application layer, specifically it doesn't require additional changes to either RTP or SIP protocols, it does need support of tunneling client and server functionalities that are usually transparently integrated into media clients and servers respectively. By far the biggest problem with encapsulation is the additional latency introduced by TCP transport that has its origin in two different mechanisms; (1) the Nagle algorithm that is an inherent part of TCP and is used to buffer and group multiple packets before transmission in order to improve the TCP header to payload length ratio [4] and (2) the retransmissions that are triggered by the network packet loss affecting media paths. Note that these retransmissions have a non-linear cumulative effect that ranges from low impact dead air to dropped calls. To a lesser extent, another negative

consequence of tunneling is a higher transmission rate due to the additional overhead introduced by the outer network and transport layer headers.

The remainder of the paper is organized as follows: brief descriptions of the details of media relying and media encapsulation mechanisms are presented in Sections 2 and 3 respectively. In Section 4 the experimental framework is introduced. Comparative results obtained by applying network impairments to the framework and computing media quality scores are given in Section 5. Finally the conclusions are provided in Section 6.

2 MEDIA RELYING

Traversal Using Relays around NAT (TURN) is one of the most well known and established mechanisms for media relying intended to overcome the problems introduced by firewall traversal [5]. TURN allows a client behind a firewall to request that a TURN server acts as a relay. The client can arrange for the server to relay packets to and from certain other hosts and can control aspects of how the relaying is done. The client does this by obtaining an IP address and port on the server, called the relayed transport address. When a peer sends a packet to the relayed transport address, the server relays the packet to the client. When the client sends a data packet to the server, the server relays it to the appropriate peer using the relayed transport address as the source. TURN is an extension to the Session Traversal Utilities for NAT (STUN) protocol [6]. Most, though not all, TURN messages are STUN formatted messages.

Figure 1 shows a typical TURN scheme; the TURN client and the TURN server are separated by a firewall, with the client on its private side and the server on its public side. The client sends TURN messages to the server from a transport address called client address. Clients typically learn the TURN server address via configuration. Since the client is behind a firewall, the server sees packets from the client as coming from an address on the firewall itself. This address is known as the client server-reflexive address. In general, packets sent by the server to this latter address will be forwarded by the firewall to the client address. The client uses TURN commands to create and manipulate an allocation, which is a data structure, on the server. This data structure contains, among other things, the relayed address for the allocation. The relayed address is that on the server that peers can use to have the server relay data to the client. An allocation is uniquely identified by its relayed address.

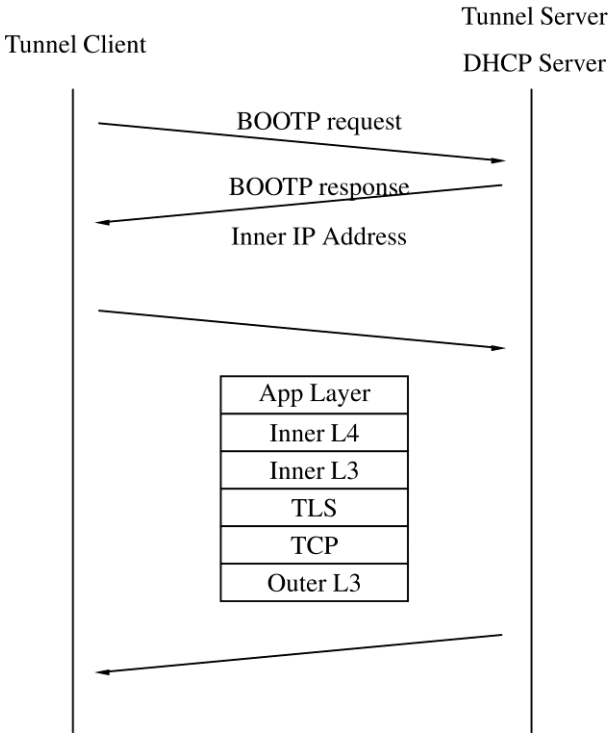

Figure 4: FTT-IMS

Once an allocation is created, the client can send application data to the server along with an indication of to which peer the data is to be sent, and the server will relay this data to the appropriate peer. The client sends the application data to the server inside a TURN message; at the server, the data is extracted from the TURN message and sent to the peer in a datagram. In the reverse direction, a peer can send application data in a datagram to the relayed address for the allocation; the server will then encapsulate this data inside a TURN message and send it to the client along with an indication of which peer sent the data. Since the TURN message always contains an indication of which peer the client is communicating with, the client can use a single allocation to communicate with multiple peers.

When the peer is behind a firewall, then the client must identify the peer using its server-reflexive address rather than its peer address. Each allocation on the server belongs to a single client and has exactly one relayed address that is used only by that allocation. Therefore, if a packet arrives at a relayed address on the server, the server knows for which client the data is intended.

3 MEDIA ENCAPSULATION

There are many media encapsulation mechanisms including, among the most popular ones, Internet Protocol Security (IPSec) that when operating in tunnel mode is typically used to create Virtual Private Networks (VPN) [7]. Figure 2 shows an example of regular IPSec encapsulation, where the IP Encapsulating Security Payload (ESP) layer is used to provide both encryption and authentication [8]. The main problem is that restrictive firewalls typically block IPSec traffic. This limitation is, however, overcome by Enhanced Security Gateway (eSEG) that supports tunneling of IMS services within a TCP encapsulation designed to carry IPSec through restrictive firewalls. Figure 3 shows an example of eSEG encapsulation, where ESP tunnel mode packets are sent over TCP by means of TPKT framing [9]. The drawback of this approach is that because outer or exterior TCP transport is involved, induced latency due to retransmissions can seriously affect the overall media quality.

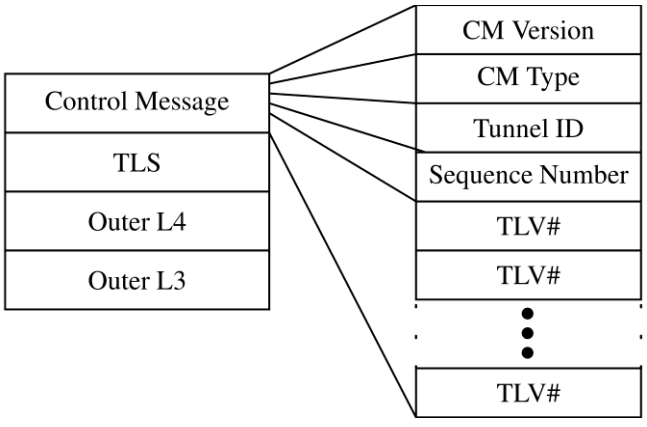

Figure 5: TSCF CM

3GPP Technical Specification 24.322 defines the Enhanced Firewall Traversal Function (EFTF) that relies on the Firewall Traversal Tunnel to IP Network of IMS (FTT-IMS) protocol [10]. Essentially, a TCP connection is created and Transport Layer Security (TLS) is used to encrypt and encapsulate all inner traffic. Both, the tunnel client and tunnel server, implement client and server Dynamic Host Configuration Protocol (DHCP) endpoints respectively. As shown in Figure 4, once an inner or internal IP address is assigned, it is used as source address of all traffic originated at the client. Note that firewall traversal is performed by means of firewall friendlier and more permissive outer TCP transport. This approach makes no difference between reliable high latency inner data traffic and non-reliable low latency inner media traffic and as in the eSEG case it is subjected to the inefficiencies that result from transmitting media over a TCP stream.

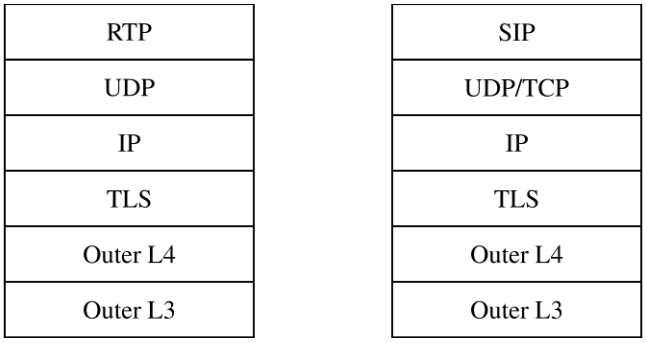

Figure 6: TSCF Inner Traffic

3GPP Tunneled Services Control Function (TSCF) is an attempt to overcome the limitations introduced by previous approaches, where outer TCP based transport is complemented by a series of mechanisms that are intended to mitigate the negative effects of stream transport applied to low latency inner media traffic [11]. TSCF defines Control Messages (CM), shown in Figure 5, as the standard method for tunnel creation, termination and maintenance. CMs are transmitted on top of TLS as a way to accomplish security and each CM includes a number of fields, specifically, (1) version used for negotiation, (2) type needed to signal tunnel actions, (3) tunnel id (TID) intended to uniquely identify the tunnel, (4) sequence number use to keep track of CM requests and responses and (5) a variable number of Type-Length-Value (TLV) parameters. Regular RTP

media and SIP signaling traffic is also sent on top of TLS as shown in Figure 6. A common characteristic of media tunneling is that during call establishment media is negotiated and internal IP addresses are exchanged. In a mobile environment, transition between networks causes the outer layer and specifically its network addresses to be continuously changing. These outer layer changes must not cause the inner layer addresses to change because this would force media addresses to be renegotiated via signaling introducing latency and potentially other more serious impairments like dead air that negatively affects the overall quality of the communication. TSCF introduces a keep alive mechanism that guarantees that the tunnel is kept functional all the time and both parties, client as well as server, are synchronized and aware of the tunnel status even when no traffic is transmitted. This mechanism is independent of outer layer keep alive methods like the one provided by TCP. The idea is to contemplate all cases, including simple low weight TCP implementations that fail to implement basic keep alive functionality.

If mobility or network impairments detected by means of keep alives cause the outer layer to be renegotiated the TSCF persistence mechanism guarantees that inner layer parameters are transparently and efficiently reprovisioned as shown in Figure 7. First, the TCP connection is created including TLS security. Then tunnel parameters are negotiated and TID as well as an IP address are assigned by means of CM Configuration Request/Response exchange. Once the tunnel is created, if connectivity is lost, no response to CM Keep Alive messages is received and eventually both, client and server sides, release associated resources. Simultaneously the tunnel server stores the tunnel information for that specific TID and the tunnel client attempts to restore the tunnel but this time issuing a CM Configuration Request that includes the TID as parameter. When received, this CM triggers the server to retrieve the information and provision the client accordingly.

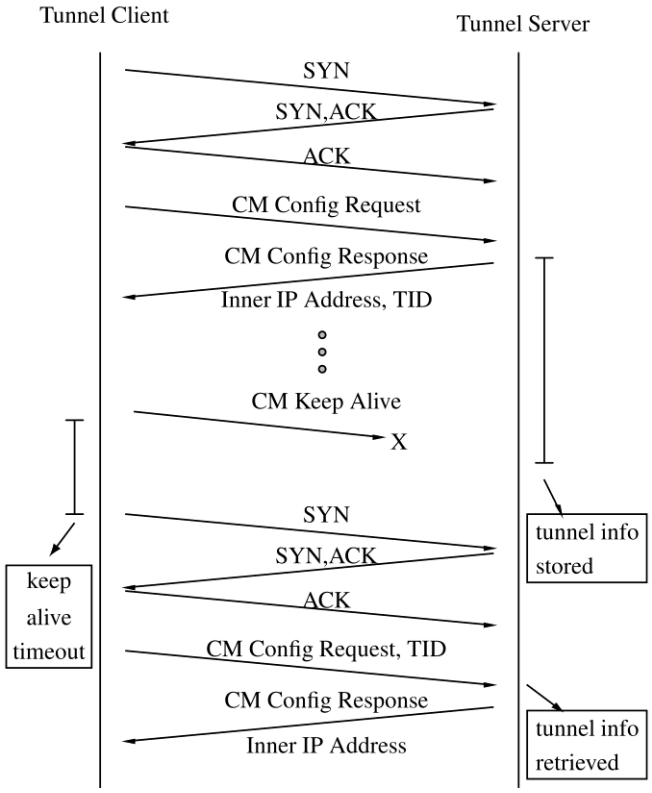

Figure 7: TSCF Tunnel Persistence

As TSCF is intended for media transport, it natively supports a Forward Error Correction (FEC) mechanism that provides multipath transmission of frames in a redundant fashion minimizing the overall latency that results from outer TCP retransmissions in lossy network scenarios. Specifically this mechanism takes advantage of the fact that most modern mobile devices incorporate multiple network interfaces (i.e. WiFI vs LTE) that can be used to transmit traffic simultaneously through many possible paths. Two FEC modes are possible; (1) fan out, shown in Figure 8, where frames are simultaneously sent over multiple tunnels and load balancing, shown in Figure 9, where depending on whether the frame number is even or odd it is sent in the main tunnel or in the redundant one. Both methods share the same set up procedure; once the main tunnel is created a CM Service Request is used to reserve a redundant tunnel. The server responds back with the TID of the redundant tunnel that is used, in turn, by a new client to establish it. When the session is to be terminated both tunnels, the original and the redundant one,

are simultaneously terminated by means of standard tunnel release procedures.

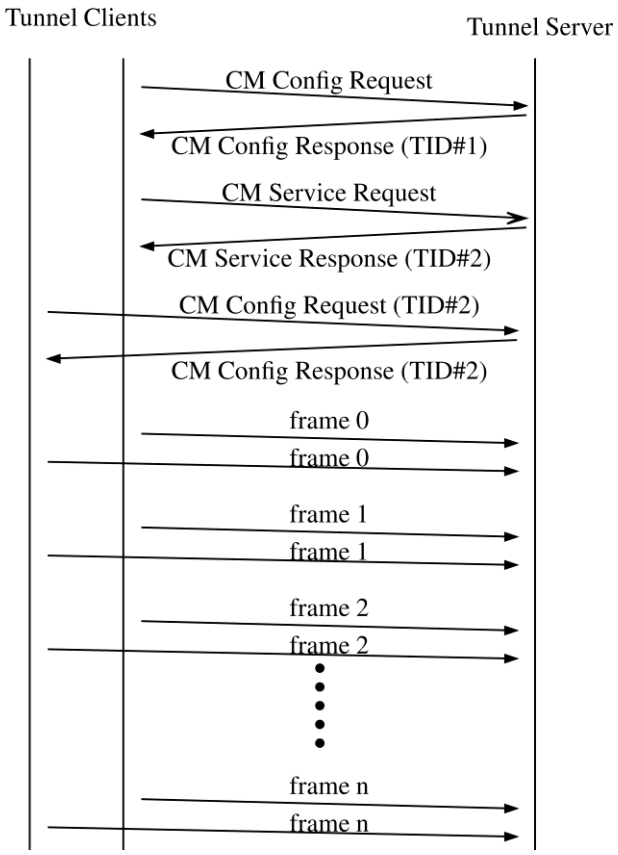

Figure 8: TSCF FEC Fan Out

FEC, however, has the negative effect of a higher transmission rate that depending on bandwidth availability and network topology can result in dropped packets and reduced media quality. TSCF provides an additional mechanism, called Dynamic Datagram Tunnels (DDT) that given a main TCP based tunnel where both signaling and media are encapsulated, a secondary UDP based tunnel is started such that when successfully negotiated all media traffic is transported through it. Of course, successful negotiation of a UDP based tunnel implies that firewalls between the client and the server allow datagram traffic, however, since this is not always possible, DDT is highly dependent on the configuration of the restrictive access networks. Figure 10 shows DDT, specifically its negotiation as well as the signaling and media encapsulation occurring in the main stream based (TID#1) and datagram (TID#2) based tunnels respectively. In general, in order to guarantee the quality of the encapsulated media, FEC and DDT provide an extra line of defense. In the following sections, the performance improvements due to these mechanisms are extensively analyzed.

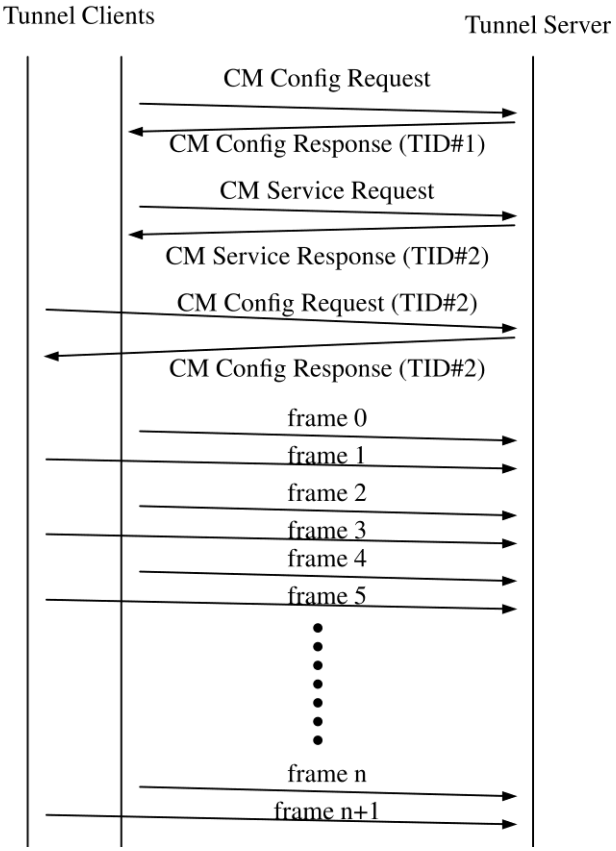

Figure 9: TSCF FEC Load Balancing

4 EXPERIMENTAL FRAMEWORK

In order to evaluate the effects of encapsulation, we introduce a scenario that involves a set of clients and tools [12] modified to support the experimental framework shown in Figure 11. To this end, a media reference, namely a speech or video sequence, is encoded as well as packetized through playback and subjected to controlled network impairments responsible of packet loss and latency before entering the Media over IP (MoIP) cloud either as (1) clear or as (2) encapsulated traffic. On the receiving side, media is decapsulated when coming through a tunnel, depacketized, decoded as well as recorded in a file

that together with the reference are used to obtain offline media quality scores by means of well-known algorithms like PESQ and PEVQ for speech and video respectively.

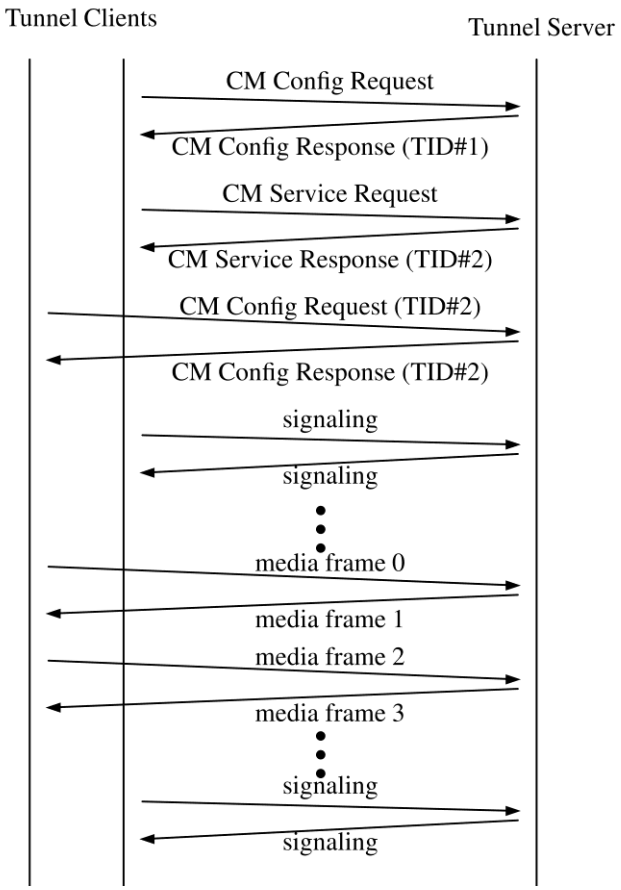

Figure 10: TSCF Dynamic Datagram Tunnels

The media references are (1) a 60-second speech sequence that has 50% of silence evenly distributed and (2) a 15-second color video sequence exhibiting a 352 × 288 Common Intermediate Format (CIF) resolution and recorded at 15 frames-per-second (fps).

For this framework, we consider a group of high-bitrate (HBR) and low-bit-rate (LBR) speech codecs as well as video codecs typically used in MoIP scenarios. In general any speech codec that provides a transmission rate below 32 Kbps is considered LBR and any one providing a rate above this threshold is considered HBR. In addition, speech codecs can be either waveform or parametric depending upon whether they preserve the shape of the original speech wave they are encoding. It can be seen that for the most cases HBR codecs are considered waveform and LBR codecs are considered parametric. In this experimental framework, the following speech codecs are considered (1) G.711 -law, (2) G.729A, (3) AMR-NB, (4) EVRC, (5) AMR-WB and (6) Opus. G.711 is a narrowband (8 KHz sampling rate) HBR codec that preserves the speech waveform through non-linear compansion at the cost of increased transmission rate [13]. G.729A is a narrowband parametric LBR codec that operates at 8 Kbps and relies on linear prediction and prediction error encoding [14]. AMR-NB is also a narrowband LBR codec that provides a wide range of compression rates at different quality levels [15]. EVRC provides narrowband speech compression at three different rates [16]. AMR-WB is the wideband (16 KHz sampling rate) version of AMR-NB that provides multiple rates of operation [17]. Opus is an LBR codec that supports both narrowband and wideband scenarios and a wide range of compression rates as well as very low latency [18]. In this paper the codecs are negotiated to operate at the following rates: G.711 at 64 Kbps, G.729A at 8 Kbps, AMR-NB at 7.95 Kbps, EVRC at 8.55 Kbps, AMR-WB at 8.85 Kbps and Opus at 8 Kbps. On the other hand, speech quality evaluation is performed by means of the well-known PESQ algorithm, standardized under P.842. This mechanism involves a Degradation Category Rating (DCR) test where synthetic speech is contrasted against the reference in order to obtain a score between 1 (bad) and 5 (excellent) [19]. Although PESQ typically operates on narrowband codecs, a wideband version of the algorithm called PESQ-WB is used for those codecs that involve a sampling rate of 16 KHz or above. In addition the following video codecs are considered (1) H.263 and (2) H.264. H.263 is a legacy video codec that relies on intra and inter frame encoding including motion estimation and several scalability modes [20]. H.264, on the other hand, represents the evolution of H.263 and mainly consists of two layers; (1) Video Coding Layer

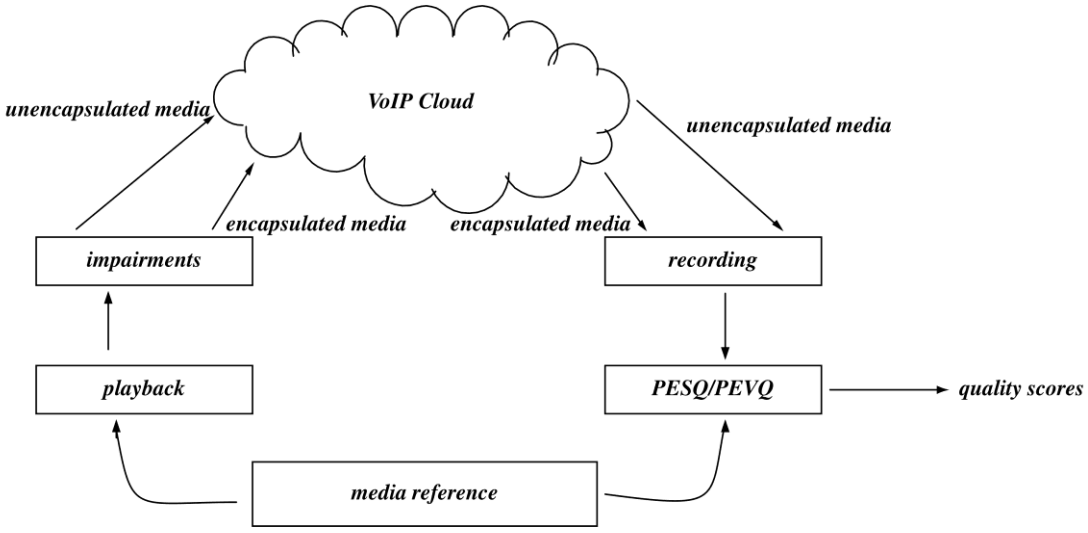

Figure 11: Experimental Framework

Table 1: Clear/Untunneled

p (%)	Δ(ms)	PESQ				PESQ-WB		PEVQ	
		G.711 μ-law	G.729A	AMR-NB	EVRC	AMR-WB	Opus	H.263	H.264
2	25	3.76	3.21	3.05	3.13	3.12	3.05	2.75	3.05
2	50	3.53	3.09	2.90	3.03	3.08	2.97	2.70	2.89
2	150	3.18	2.69	2.63	2.64	2.71	2.72	2.35	2.57
5	25	3.46	3.04	2.78	2.94	2.96	2.95	2.64	2.77
5	50	3.39	2.93	2.70	2.83	2.84	2.88	2.45	2.70
5	150	3.02	2.54	2.42	2.45	2.59	2.46	2.22	2.48
15	25	2.75	2.35	2.19	2.25	2.32	2.31	2.01	2.21
15	50	2.56	2.22	2.18	2.16	2.31	2.21	1.91	2.08
15	150	2.31	2.00	1.89	1.96	1.96	1.90	1.73	1.88

Table 2: Encapsulated

p (%)	Δ(ms)	PESQ				PESQ-WB		PEVQ	
		G.711 μ-law	G.729A	AMR-NB	EVRC	AMR-WB	Opus	H.263	H.264
2	25	2.90	2.59	2.44	2.49	2.51	2.52	2.15	2.44
2	50	2.94	2.43	2.30	2.35	2.47	2.46	2.10	2.40
2	150	2.53	2.23	2.03	2.08	2.17	2.14	1.87	2.03
5	25	2.77	2.36	2.24	2.32	2.36	2.27	2.10	2.21
5	50	2.73	2.36	2.16	2.31	2.37	2.26	2.05	2.15
5	150	2.34	2.07	1.89	1.97	2.08	2.02	1.79	1.92
15	25	2.16	1.83	1.72	1.83	1.82	1.80	1.63	1.79
15	50	2.13	1.81	1.67	1.78	1.85	1.73	1.60	1.68
15	150	1.86	1.58	1.47	1.53	1.62	1.52	1.34	1.48

Table 3: Encapsulated with FEC

p (%)	Δ(ms)	PESQ				PESQ-WB		PEVQ	
		G.711 μ-law	G.729A	AMR-NB	EVRC	AMR-WB	Opus	H.263	H.264
2	25	3.76	3.33	3.07	3.17	3.18	3.24	2.81	3.03
2	50	3.59	3.19	2.93	3.03	3.12	3.02	2.77	2.96
2	150	3.27	2.78	2.69	2.68	2.77	2.74	2.38	2.66
5	25	3.53	3.03	2.84	3.03	3.01	3.04	2.67	2.95
5	50	3.38	3.04	2.85	2.84	3.02	2.83	2.55	2.82
5	150	3.07	2.61	2.43	2.50	2.59	2.50	2.31	2.42
15	25	2.72	2.36	2.28	2.29	2.42	2.36	2.06	2.30
15	50	2.73	2.30	2.17	2.23	2.36	2.26	1.97	2.22
15	150	2.32	2.08	1.90	2.03	2.06	1.97	1.75	1.92

Table 4: Encapsulated with DDT

p (%)	Δ(ms)	PESQ				PESQ-WB		PEVQ	
		G.711 μ-law	G.729A	AMR-NB	EVRC	AMR-WB	Opus	H.263	H.264
2	25	3.65	3.08	2.95	3.09	3.14	3.04	2.67	2.96
2	50	3.57	3.13	2.90	3.00	3.12	3.03	2.66	2.85
2	150	3.07	2.69	2.61	2.59	2.69	2.59	2.36	2.50
5	25	3.38	3.00	2.81	2.89	2.95	2.86	2.50	2.75
5	50	3.28	2.83	2.75	2.72	2.85	2.73	2.48	2.70
5	150	2.89	2.57	2.43	2.44	2.59	2.48	2.23	2.35
15	25	2.69	2.34	2.15	2.21	2.30	2.20	2.02	2.17
15	50	2.53	2.24	2.12	2.13	2.24	2.12	1.96	2.15
15	150	2.31	1.97	1.84	1.91	1.94	1.96	1.70	1.88

(VCL) that provides, among other things, inter and intra frame prediction and (2) Network Abstraction Layer (NAL) that is used to packetize VCL data [21]. Similar to the speech case, video quality is evaluated by means of the PEVQ algorithm that is standardized as J.247 and involves a DCR test where the decoded video is compared against the full reference to obtain a score between 1 (bad) and 5 (excellent) [22]. Since video codecs typically exhibit variable transmission rates, in order to provide a consistent and fair comparison, both video codecs are configured at a fixed transmission rate of 128 Kbps. It is critical to mention that because PEVQ scores, as opposed to PESQ ones, are highly dependent on the video sequence used as reference, relative comparison between codecs is more relevant than absolute evaluation of scores. In the following section, the different codecs under study have their performance compared under two possible scenarios; (1) with and (2) without encapsulation.

5 COMPARATIVE ANALYSIS

In this section the effect of both, clear and encapsulated traffic, are evaluated and more specifically the following transport test cases are considered; (1) clear/untunneled, (2) encapsulated, (3) encapsulated with FEC and (4) encapsulated with DDT. In addition network impairments are incorporated to this testing; (1) 2%, (2) 5% and (3) 15% packet loss (p) as well as (1) 25 ms, (2) 50 ms and (3) 150 ms latency ( ). In order to provide a complete analysis, this comparison assumes a permissive firewall that allows traversal of all transport types. It can be seen that more restrictive access networks prevent datagram based traffic and cause clear/untunneled and DDT scenarios to fail. Under these conditions stream based encapsulation, relying or not on FEC, is the only acceptable solution. Note that encapsulation with FEC implies that traffic is fanned out through three independent redundant tunnels affected by the same network packet loss and latency. All codecs have different transmission rates (i.e. G.711 μ-law at 64 Kbps vs G.729 at 8 Kbps), different inter packet duration (i.e. 20 milliseconds

G.729 vs 30 milliseconds G.723.1) and different Discontinued Transmission (DTX) sensitivities that make their direct comparison fairly irrelevant. The goal of this paper is to compare the overall effects of the aforementioned mechanisms when applied to each of the speech and video codecs independently. As previously mentioned standardized scores are used to this end; (1) PESQ scores apply to narrowband speech codecs like G.711 μ-law, G.729A, AMR-NB and EVRC, (2) PESQ-WB scores apply to wideband speech codecs like AMR-WB as well as Opus and (3) PEVQ scores apply to video codecs like H.263 and H.264. Tables 1, 2, 3 and 4 show the performance results obtained by evaluating the codecs when the scenarios described above are implemented.

6 CONCLUSIONS

Common to all tables, among the narrowband codecs, G.711 μ-law always provides the best quality based on the fact that has a transmission rate that is an order of magnitude higher than that of G.729A, AMR-NB and EVRC. These latter codecs, consequently, exhibit levels of quality consistent with the linear prediction techniques they rely upon. The wideband codecs, AMR-WB and Opus, also show similar quality levels that comply with their also similar transmission rates. On the video front, because of the improvements associated to H.264, this codec exhibits quality that is consistently superior to that of H.263 for the same transmission rate.

When analyzing performance, regular encapsulated datagram media traffic is naturally affected by loss and latency that results in playout buffers skipping missing frames and causing impairments that negatively affect quality. As expected, the larger the loss and the latency the more negative the effect on the quality score. On the other hand, regular stream based encapsulation guarantees that no inner datagram media frames are lost due to TCP retransmissions, however, it introduces extra latency that causes the frames to arrive too late for the playout buffer to play them. The playout buffer is typically dynamic and automatically adjusts itself according to the network latency, however, a latency value of 150 milliseconds is the threshold that most buffers support as higher values degrade the overall user experience. Quality is significantly worse (around 20% in average) when encapsulation is used for transmission as opposed to plain clear traffic. Again, under very restrictive networks, clear traffic is not allowed so bad quality is better that no quality at all.

In order to improve the scores of encapsulated media, FEC by means of multi-tunnels is a feasible solution that provides a similar quality to clear traffic transport for all network restriction scenarios even those that prevent clear traffic from traversing a network. FEC in the context of stream based encapsulation ideally requires the availability of alternative paths for media to flow. On the other hand, the cost of FEC is a higher transmission rate due to the multipath transmission of traffic. An additional technique that can be used to obtain better media quality is via DDT, which encapsulates time sensitive media in a datagram based tunnel. This mechanism exhibits a performance that is slightly inferior to that of clear traffic, mostly due to the overhead of the DDT negotiation that causes traffic to traverse the stream based tunnel until the datagram one is fully established. Since DDT is contingent to the network allowing datagram traffic traversal, when this is not possible, media traffic fallbacks to regular stream based encapsulation.

As opposed to clear media which cannot traverse highly restrictive networks, encapsulated traffic, regardless of its nature, can always go through firewalled networks and dynamically take advantage of FEC or DDT depending on the network resource availability. When network loss and latency are high enough that affect user experience (1) if multipath traversal is available then FEC provides the best solution and (2) if not, DDT support is the next best alternative.

REFERENCES

1. 3GPP TS 23.107: Technical specification group services and system aspects; Quality of Service (QoS) concept and architecture (release 9), v9.0.0 Dec. 2009.

2. Schulzrinne, H., Casner, S., Frederick , R., Jacobson, V.: RTP: A Transport Protocol for Real-Time Applications, RFC 3550 (Internet Standard), July 2003,

Updated by RFCs 5506, 5761, 6051, 6222.

3. Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler , E.: SIP: Session Initiation Protocol, RFC 3261 (Proposed Standard), June 2002.

4. Minshall, G., Saito, Y., Mogul, J., C., Verghese, B: Application performance pitfalls and TCP's Nagle algorithm, SIGMETRICS Perform. Eval. Rev., vol. 27, no. 4, pp. 36–44, Mar. 2000.

5. Mahy, R., Matthews, P., Rosenberg, J.: Traversal Using Relays around NAT (TURN) RFC 5766 (Internet Standard), 2010.

6. Rosenberg, J., Mahy , R., Matthews P., Wing, D: Session Traversal Utilities for NAT (STUN), RFC 5389 (Internet Standard), 2008.

7. Kent, S., Seo, K: Security Architecture for the Internet Protocol RFC 4301 (Proposed Standard), December 2005.

8. Kent, S: IP Encapsulating Security Payload (ESP): RFC 4303 (Proposed Standard), December 2005.

9. Pouffary, Y., Young, A: ISO Transport Service on top of TCP (ITOT) RFC 2126 (Proposed Standard), March 1997.

10. 3GPP TS 24.322: Tunneling of IP Multimedia Subsystem (IMS) services over restrictive access networks; Stage 3,, 3rd Generation Partnership Project (3GPP), 06 2015.

11. 3GPP TR 33.830: Feasibility study on IMS firewall traversal,, 3rd Generation Partnership Project (3GPP), 01 2014.

12. Ecotronics: Kapanga Softphone http://www.kapanga.net.

13. ITU-T G.711: Pulse Code Modulation (PCM) of voice frequencies, Tech. Rep. G.711, International Telecommunication Union, Geneva, 2006.

14. Salami, R., Laflamme, C., Bessette , B., Adoul J: Description of ITU-T recommendation G.729 annex A: Reduced complexity 8 kbit/s cs-acelp codec," in Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)-Volume 2 - Volume 2, Washington, DC, USA, 1997, ICASSP '97, pp. 775–, IEEE Computer Society.

15. 3GPP TS 26.071: Mandatory speech codec speech processing functions; amr speech codec; general description,, 3rd Generation Part- nership Project, 2008.

16. 3GPP2 C.S0014-a: Enhanced variable rate codec, speech service option 3 for wideband spread spectrum digital systems, 3rd Genera- tion Partnership Project 2, 2004.

17. 3GPP TS 26.190: Speech codec speech processing functions; adaptive multi-rate - wideband (AMR-WB) speech codec; transcoding functions, 3rd Generation Partnership Project, 2008.

18. Valin, JM., Vos, K., Terriberry, T.: Definition of the Opus Audio Codec RFC 6716 (Proposed Standard), Sept. 2012.

19. ITU-T P.862.2: Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs Tech. Rep., International Telecommunication Union, Geneva, Switzerland, Nov. 2007.

20. ITU-T H.263: Video coding for low bit rate communication Tech. Rep. H.263, International Telecommunication Union, Geneva, 2005.

21. ITU-T H.264: Advanced video coding for generic audiovisual services," Tech. Rep. H.264, International Telecommunication Union, Geneva, 2014.

22. ITU-T J.247: Objective perceptual multimedia video quality measurement in the presence of a full reference," Tech. Rep. J.247, International Telecommunication Union, Geneva, 2008.