國立交通大學

資訊學院　資訊學程

碩士論文

一個智慧型網路語音服務品質提供系統

A Smart QoS Provisioning System for Voice over IP

研究生：張一飛

指導教授：陳耀宗　教授

中華民國九十八年一月

一個智慧型網路語音服務品質提供系統
A Smart QoS Provisioning System for Voice over IP

研究生：張一飛　　Student：Yi-Fei Chang

指導教授：陳耀宗 博士　　Advisor：Yaw-Chung Chen

國立交通大學

資訊學院　資訊學程

碩士論文

A Thesis

Submitted to College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master of Science
in

Computer Science

January 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年一月

一個智慧型網路語音服務品質提供系統

學生：張一飛　　　　　　　　　　　　　　指導教授：陳耀宗 博士

國立交通大學 資訊學院 資訊學程碩士班

摘　　要

寬頻網路服務及家用閘道器的普遍化造就了家庭網路使用者的廣泛相關應用，藉由網際網路承載語音的免費或低價網路電話是即時網路傳輸主要的應用之一。然而當同一個使用下的家用閘道器，即時網路傳輸（如：網路電話）如果和其它的網路相關應用服務在同一段時間發生，必定會造成對內及對外的頻寬資源要求的衝突與競爭，進而造成維持可接受即時網路傳輸服務品質的困難。本論文中提出了方法來解決即時網路傳輸網路服務品質提供系統的三個重要基本需求（計算對外可用頻寬資源，對外頻寬的流量控制以及對內流量壅塞控管），並在家用閘道器上實現智慧型網路語音服務品質提供系統以針對SIP網路電話做即時傳輸的品質控制，藉以驗證所提出方法的正確性及可行性，進而提出一可廣泛地應用在家用或企業用閘道器做其相關之即時網路傳輸品質控制之方案。

A Smart QoS Provisioning System for Voice over IP

Student：Yi Fei Chang　　　　Advisors：Dr. Yaw-Chung Chen

Degree Program of Computer Science
National Chiao Tung University

ABSTRACT

Experiencing the rapid growth of broadband infrastructure and the wide deployment of the residential gateway, various Internet applications are getting more popular for home users. VoIP is one major real-time streaming application used to carry voice data over the IP network and provides the users with free or low cost phone service compared to the traditional PSTN service. To maintain the acceptable end-to-end delay and packet loss for the real-time streaming applications becomes a challenge for the home users behind the residential gateway in case the real-time streaming packets are competing with the non-priority packets to access the limited bandwidth in the egress or ingress directions. Three effective algorithms are proposed to address the egress bandwidth measurement, egress traffic shaping and ingress traffic congestion avoidance, which are commonly required by the residential gateway for the service quality provisioning system for general real-time streaming applications. SQPV (Smart QoS Provisioning System for VoIP) is a service quality provisioning system with these three algorithms implemented and specifically deals with SIP/RTP sessions to demonstrate the result. These algorithms are proposed and able to be leveraged to support the service quality provisioning system for different real-time streaming applications.

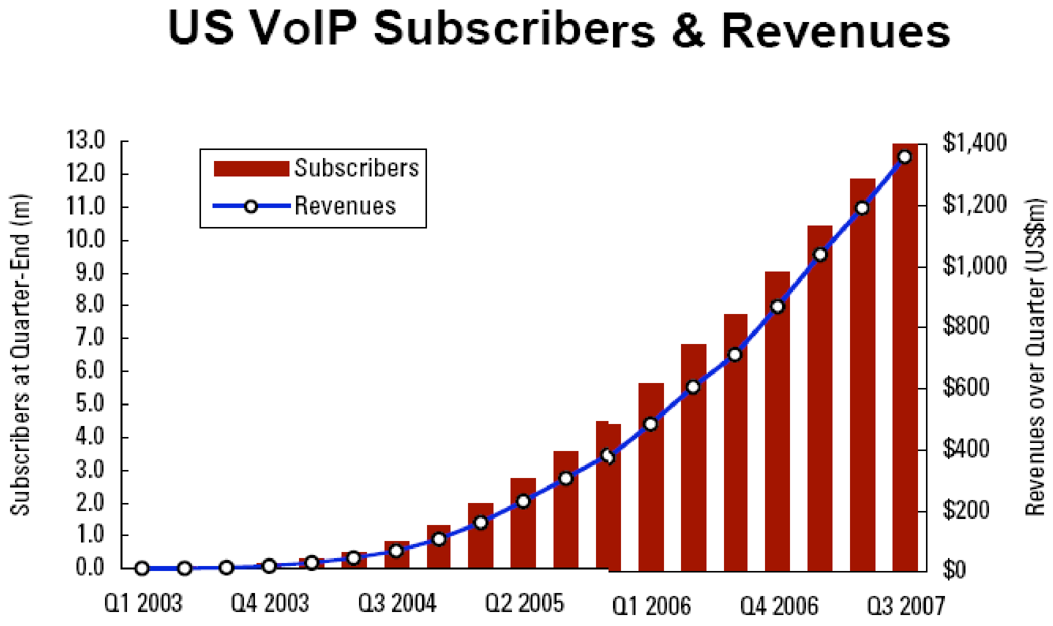
誌謝

首先要致上最深切的感激給我的指導老師陳耀宗教授，在他的熱心及耐心的指導下,讓一個交大資工系畢業十多年並重回母校進修的我能夠具體化和組織化對論文的想法並實現。其次要感謝我的太太美綺對我在寫論文期間的諸多包容及支持，尤其是在我們的女兒瀞文出生以後所承受的多重壓力，更要感謝我母親的付出及分擔家事，才能讓我在工作之餘找出時間來完成我的論文。最後要把本論文獻給在生前一直不斷鼓勵我求學上進的父親，因為有您一路的關懷和教導，我才能一步一步走出自己人生的路。

1 Preliminaries

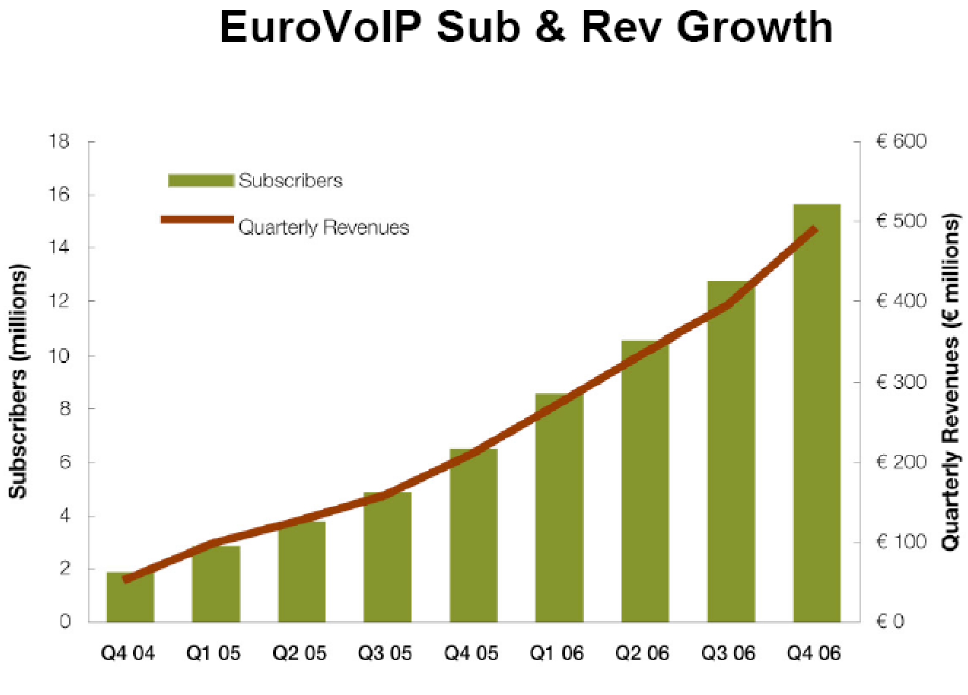
Broadband Internet services have become a rapidly growing market worldwide since year 2000. “Broadband” means the multi-channel data transmission over a physical line which is usually the access link. DSL (Digital Subscriber Line) and cable modem are the major two technologies currently deployed worldwide. The home router, RG (Residential Gateway) or IAD (Integrated Access Device), so-called the CPE (Customer Premises Equipment) are commonly attached to the DSL or cable network to provide the broadband services.

IP Telephony, also called VoIP, is one of the major broadband Internet applications which are used to carry voice data over the IP network and able to provide the home or SOHO (Small Office Home Office) users with the free or low cost phone service compared to the traditional PSTN (Public Switched Telephone Network) services. According to the marketing analysis report generated by Paul Brodsky, TeleGeography Research [12] depicted in Figure 1 and 2, the number of subscribers reaches around 13 millions in U.S in Q3, 2007 and 15 millions in Europe in Q4, 2006. And also the good projection of the growth of VoIP subscribers can be made according to deployment of broadband subscriptions as depicted in Figure 3.



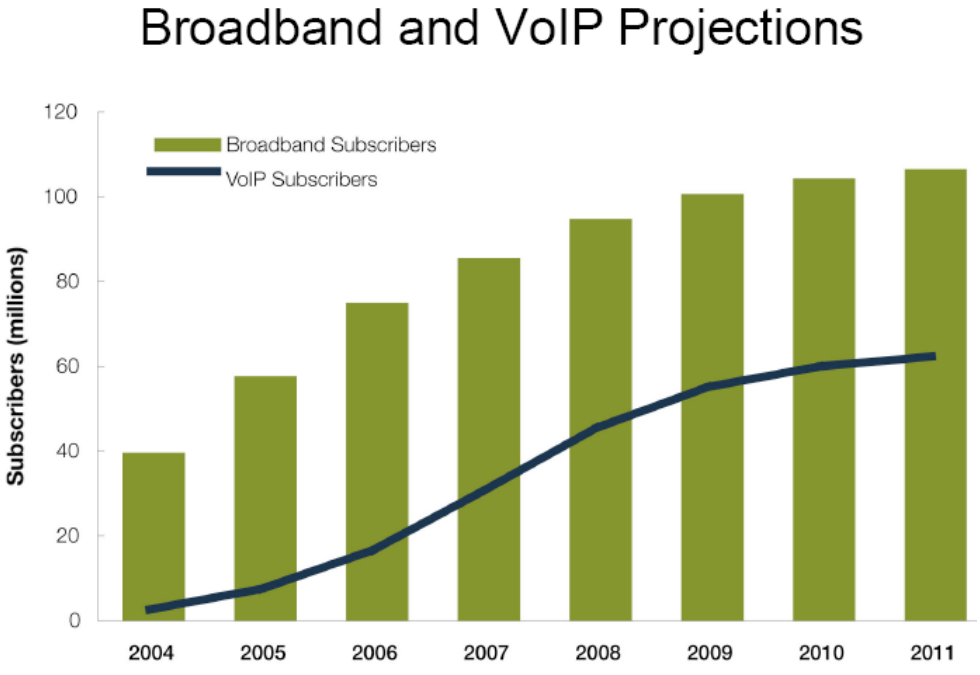
www.telegeography.com

Figure 1 : U.S. VoIP Subscribers & Revenues Growth.



www.telegeography.com

Figure 2 : Europe VoIP Subscribers and Revenues Growth.



www.telegeography.com

Figure 3 : Broadband and VoIP Projections.

Due to the rapid growth of various kinds of Internet applications (such as the FTP and P2P) which are competing against real-time streaming applications to use the shared available egress and ingress bandwidth at home, it becomes very difficult for real-time streaming applications to manage and maintain the acceptable network latency, packet loss and jitter. VoIP requires very low jitter (delay variation), a one-way delay no larger than 150 milliseconds and the guaranteed bandwidth in the range of 8Kbps to 64Kbps, depending on the codec used. However, the voice quality is one of the major challenges to implement and widely deploy the VoIP service which is acceptable by the PSTN users. Here below is one use case which can be commonly observed by the residential gateway (RG) users. Let's assuming one network user is sitting behind the RG and establishing multiple FTP sessions to upload and download files. In the meantime, another network user sitting behind the same residential gateway trying to make a VoIP call may have a hard time because each FTP session will try as fast as possible to upload or download the files and make it unavoidable to damage the network latency, packet loss and jitter and impact the quality of the real-time applications like VoIP.

To address the issues depicted as above, the targeted service quality provisioning system needs to at least consist of the following elements in order to handle the service quality of various kinds of real-time applications.

- Measure the maximum bandwidth in egress direction
- Classify and shape the traffic in egress direction
- Avoid the traffic congestion in the ingress direction
- Admission control by interpreting the correlative signaling protocol and allocating the required bandwidth

2 Background

The architecture of QoS-Enabled residential gateway was proposed by Deepak Bansal, Jeffrey Q. Bao, and Whay C. Lee, Motorola Labs in the paper titled "QoS-Enabled Residential Gateway Architecture" [15]. Figure 4 presents a functional block diagram of the residential gateway next-generation (RGng). Both data and control plane functional components which are required for QoS provisioning are depicted accordingly. For the data plan, the classifier identifies different categories of packets coming from different network interfaces and classifies them into various classes so that QoS mechanisms designed for each class can be applied. The BIB (Broadband Intelligent Bridge) interconnects various home LAN segments as well as the WAN segment, providing QoS support to data flows through scheduling, queue, management, and resource adaptation. For the control plane, the RSVP module provides signaling support of IntServ-based QoS management. The SBM (Sub-network Bandwidth Manager, defined in RFC2814 [16]) module implements admission control and provides DSBM (Designated SBM) functionality to the home network segments. This paper shows a good perspective of next generation residential gateway with QoS supported but it highly relies on the correspondent deployment in service provider side like CableHome mentioned in this paper as an example. This motivates the study to develop a QoS provisioning model at residential gateway which can work independently as much as possible.

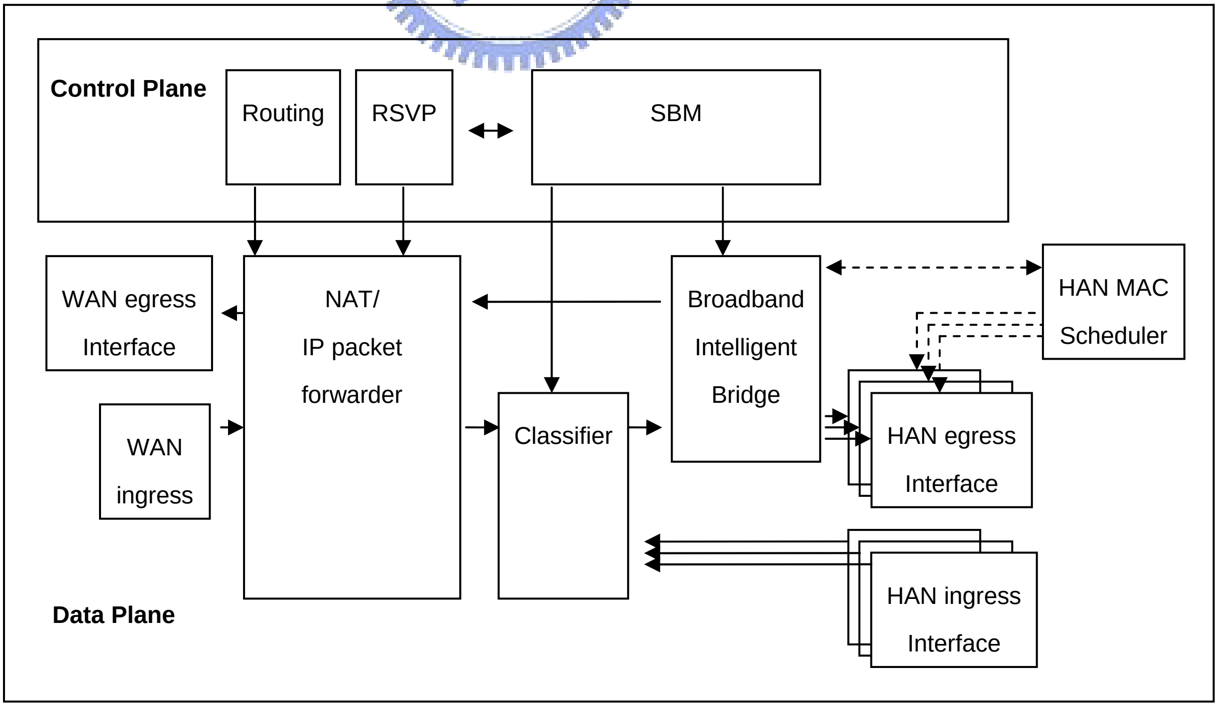


Figure 4 : A functional Block Diagram of the RGng Architecture

Another service model of a QoS-aware residential gateway was proposed by Wen-Shyang Hwang and Pei-Chen Tseng in the paper titled “A QoS-aware Residential Gateway with Bandwidth Management” [13]. The network architecture was depicted in Figure 5 as below. The key motivation to trigger this research is that the high percentage of family PC’s time is spent for real time application or media use. This new market trend is emerging from the fusion of PC and A/V (Audio & Video) multimedia. Since the RG is a core device which is transporting data, A/V, home control and multimedia streams in the integrated heterogeneous home networking environment, it’s required for RG to provide sufficient QoS and meet the demands of both existing and emerging multimedia applications, which is called QRG (QoS-aware RG) in the referred paper.

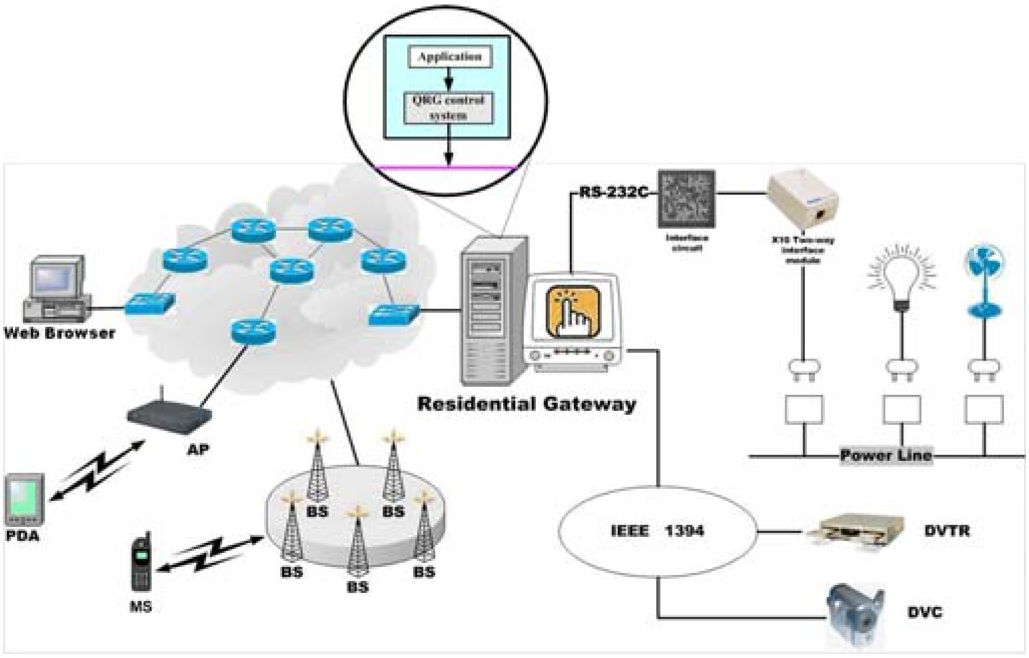


Figure 5 : ORG Network Architecture

The bandwidth management proposed in ORG is CBQ (Class Based Queuing), which is a method of classifying, allocating and sharing network bandwidth among classes of traffic. Traffic is classified into different classes and the bandwidth is allocated to them accordingly. By referring to the “Note on CBQ and guaranteed service” [14], the CBQ is using the token bucket concept in the manner of PRR (Packet-by-packet Round-Robin) or WRR (Weighted Round-Robin) to deal with traffic shaping.

Since the tokens are generated in a specific rate and the resolution of system timer will directly impact how the tokens are generated and packets are sent. As depicted in Figure

6, R is the number of tokens generated per second, M is the maximum number of tokens generated and T is the time interval to check and update the status of tokens. The packets started at the beginning of each time interval will be allowed to be transmitted until it's running out of tokens. All of the packets after running out of tokens will be dropped until the next time interval is coming as shown in the first half of Figure 6. If the timer resolution is not good enough (that is, time interval T is longer), the worse distribution of transmitted (or dropped) packets may happen. The second half of Figure 6 showed a better distribution in terms of the packets to be transmitted or dropped which we should expect regardless of the timer resolution. (This is why the Random-Prorated Drop Token Bucket is proposed to address this portion and it will be explained in further detail in the following section)

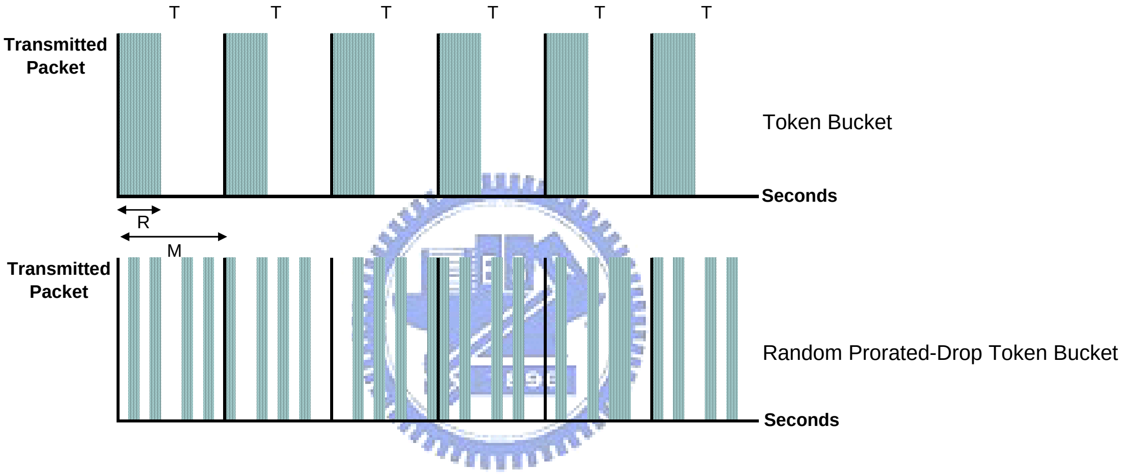


Figure 6 : Token Bucket and Random Prorated-Drop Token Bucket

The congestion detection mechanism implemented on QRG is to use the GNU Wget to periodically retrieve traffic flow data files of incoming and outgoing traffic flow. The retrieved data includes the immediate moment as well as 5 minutes. If the immediate traffic is greater than the 5-minute average, the congestion will be treated happened and the correspondent traffic shaping will be triggered to be enabled. This congestion detection mechanism cannot address precisely if the priority data streams are getting satisfied. The other approach to detect the congestion needs to be figured out in our SQPV system. The detection and measurement of the maximum available data bandwidth in upstream and downstream directions were missed in the ORG proposal. This is also required to be investigated in SQPV as well.

3 SQPV Service Model

3.1 Introduction

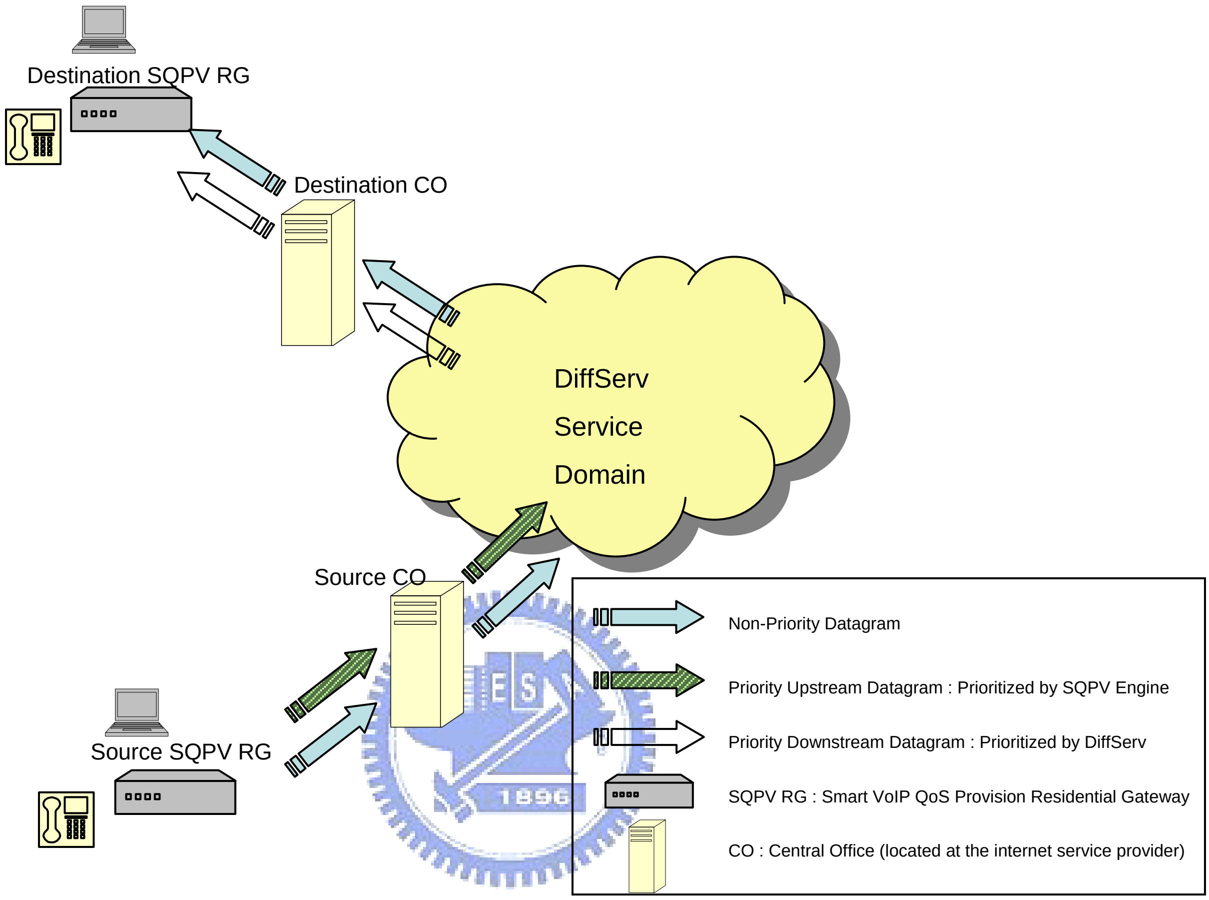

Destination SQPV RG
Destination CO
DiffServ
Service
Domain
Source CO
Source SQPV RG
Non-Priority Datagram
Priority Upstream Datagram : Prioritized by SQPV Engine
Priority Downstream Datagram : Prioritized by DiffServ
SQPV RG : Smart VoIP QoS Provision Residential Gateway
CO : Central Office (located at the internet service provider)

Figure 7 : SQPV Service Model

The service model called SQPV (Smart QoS Provisioning system for VoIP) was established. Three key algorithms were proposed accordingly to address the egress bandwidth measurement, egress traffic shaping and the ingress traffic congestion avoidance. The objective was to demonstrate the functionality and feasibility of these algorithms via the implementation of SQPV and make a proposal to leverage these algorithms for a common quality of service provisioning model which could be applied for all real-time streaming applications.

RG (Residential Gateway) is a networking device connecting the home network to WAN (Wide Area Network) or Internet. RG may sit between the modem (DSL or Cable) and internal network, or the DSL (or Cable) modem can be integrated into the RG. Routing, NAT, firewall, multiple Ethernet switch and WiFi access point are commonly supported by GW.

CO (Central Office, in the field of telecommunication) refers to the physical building equipped with telephone switches which can make phone call connections and relay the speech information. In the past, the ISPs (Internet Service Provider) were usually run by the phone companies. So CO now is a common term standing for the Internet service provider. Various kinds of technologies (such as dial-up, DSL, cable modem, broadband WLAN access, FTTH/FTTB and ISDN) provide consumer or business users the access to the Internet. With the increasing popularity of downloading music and video, the general demands for the higher bandwidth connections are becoming more popular.

Diffserv [2] is a coarse-grained, class-based mechanism for traffic management. Compared to the IntServ (Integrated Service), which provides the mechanism to allocate resources to individual flows, the DiffServ will only allocate resources to a small number of classes of traffic. Each data packet is classified and placed into a limited number of traffic classes and each class can be managed differently to make sure the good treatment of higher priority traffic on the network.

In the SQPV service model described in Figure 7, the SQPV RG is the RG with SQPV engine implemented. SQPV engine is implemented to meet the requirement as described above. The source SQPV RG will classify the priority upstream datagram and make sure that it'll be forwarded to the CO with the minimum delay. The implemented congestion avoidance mechanism in SQPV engine/SQPV GW will make sure the forwarded (by destination CO) priority downstream datagram won’t be congested by means of slowing down the ingress non-priority datagram.

CO is the common term of the equipments deployed at CO site and able to classify and forward the priority datagram received from SQPV RG, mark the DSCP bits for each priority datagram before sending to Diffserv service domain and forward the priority datagram received from Diffserv service domain to the SQPV RG accordingly. Once the priority datagram reaches the source CO, it'll also be classified and marked for EF (Expedited Forwarding, defined in DSCP, the 6-bit value that identifies a particular PHB to be applied to a packet) treatment which should be forwarded by the router with minimal delay and loss in the Diffserv service domain until it reaches the destination CO. The destination CO will then forward the priority datagram with the minimal delay and loss to the destination SQPV RG.

The implementation of CO and DiffServ service domain to meet the requirement described above is beyond the scope of my research and presumed in place to make sure the priority datagram can always be serviced and forwarded with the minimal delay and loss.

3.2 SQPV Engine

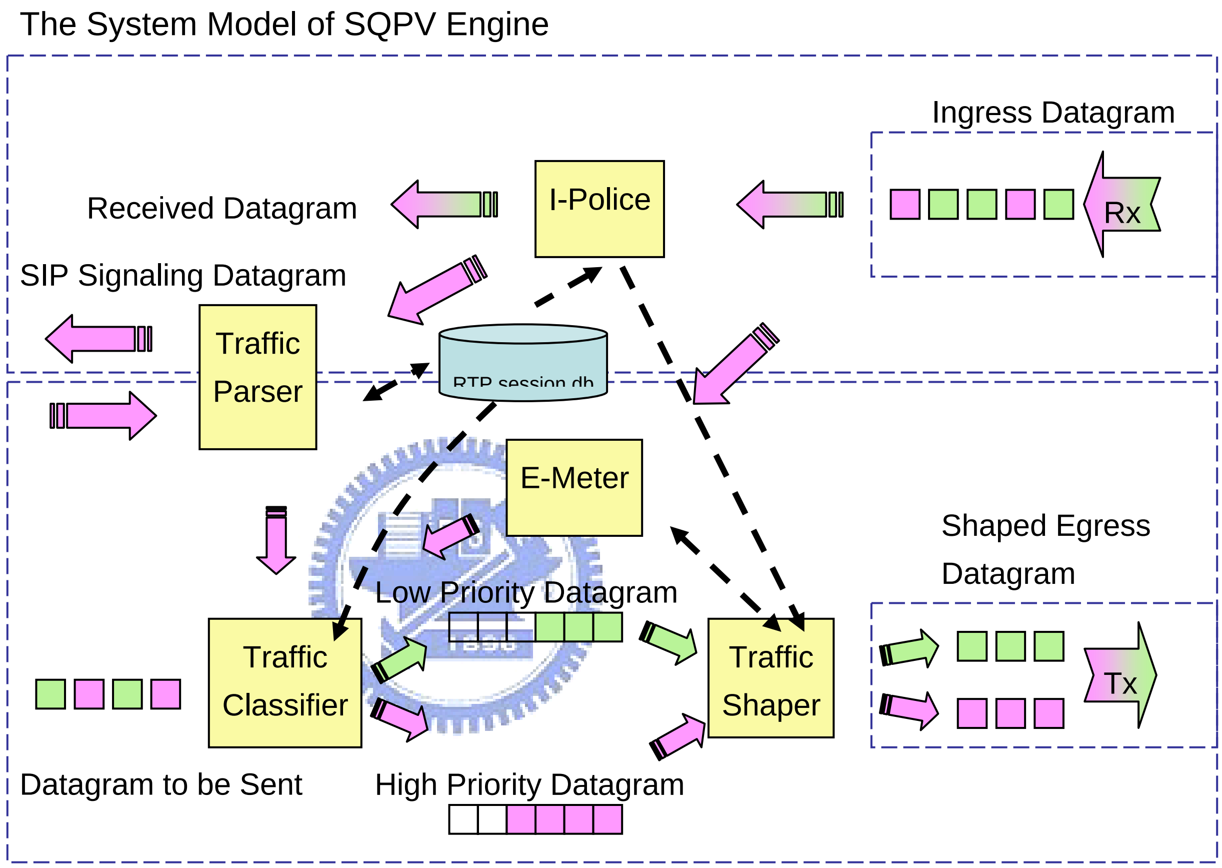


Figure 8 : System Model of SQPV Engine

The SQPV engine consists of I-Police, E-Meter, Traffic Parser, Traffic Classifier and Traffic Shaper. The egress traffic classification is counting on the information obtained from Traffic Parser. The Traffic Shaper will shape the egress traffic for high/low priority datagram before sending them out. E-Meter provides the Traffic Shaper with the maximum egress bandwidth by checking and measuring the maximum allowed egress traffic periodically. The rate of incoming priority datagram will be calculated by I-Police and if it’s lower than the expected rate, the I-Police will trigger a mechanism to slow down the non-priority traffic until the expected rate of priority datagram is met.

The system model of SQPV engine shown in Figure 8 gives a quick overview of the data flows and the control plane as described above. More details to model and implement each key element will be addressed in the subsequent sections.

3.2.1 E-Meter

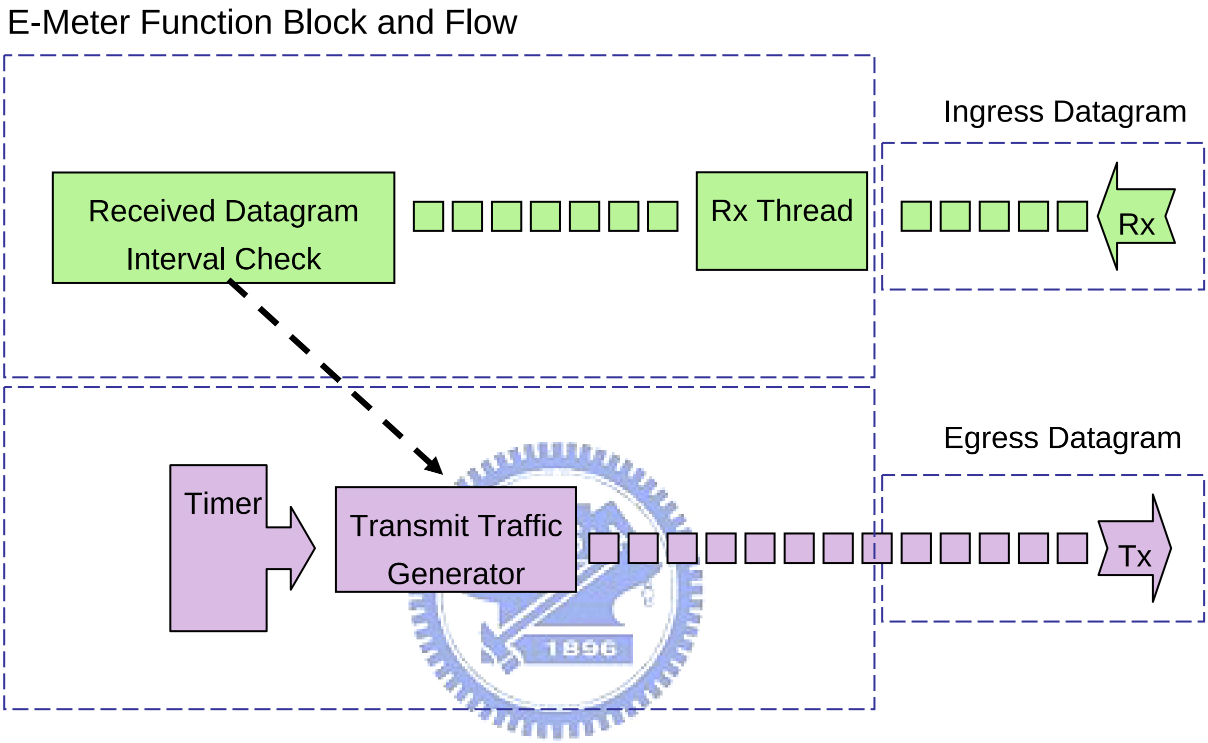

E-Meter Function Block and Flow
Ingress Datagram
Received Datagram Interval Check
Rx Thread
Rx
Egress Datagram
Timer
Transmit Traffic Generator
Tx

Figure 9 : E-Meter Functional Block and Flow

E-Meter is standing for "Egress traffic Meter", which measures the maximum possible data bandwidth in the egress direction and keep the information to be referred by the Traffic Shaper later on. The maximum possible egress data throughput is one of the major parameters required by the Traffic Shaper to shape the bandwidth for the low priority and high priority traffic.

The general way to measure the maximum egress data bandwidth is to install the proprietary application to send the UDP traffic at one end and another application at the other end to receive the traffic. The sending application will try to send out the traffic as fast as it can without having any packet drop on the receiving end. The maximum rate that the sending application can reach without having any packet drop at the receiver will be the maximum egress bandwidth. However, this is not a good approach due to the following reasons. First, the proprietary implementation needs to be in place at both sending and receiving sides. This creates the complexity to deploy the receiving application on the CO side as well as the limitation to deploy the SQPV engine on the CPE end. Secondly, it may impact the normal egress traffic if the sending application keeps sending traffic in the upstream direction and tries to approach the maximum egress data bandwidth. The longer time it takes, the larger the impact on the normal egress traffic is expected.

Functional blocks and flows of M-Meter are shown in Figure 9. The minimum possible time interval will be configured for the timer. Each timer interrupt will trigger an event and the callback routine will be called accordingly. In the timer callback routine, the packet of ICMP echo will be generated. The destination IP of the ICMP echo message will be the IP address of the default gateway, which is located at the end of ISP. Once a packet is prepared OK, it'll be transmitted immediately as shown in the function block "Transmit Traffic Generator". The size of the ICMP echo packet can be prepared according to the feedback of the result which comes from the function block "Received Datagram Interval Check". There is a thread created "Rx Thread" which is always listening in the incoming ICMP echo reply message. This message is correspondent to the ICMP echo message sent previously, and supposed to be sent by the default gateway at the end of ISP. The "Rx Thread" will also track the time for each ICMP echo reply message received which will be referred by the "Received Datagram Interval Check" to measure the time interval of each received packet. By means of increasing the payload size of the sent packets and comparing the time interval of the received packets with the original time interval of the sent packets, we can detect the turning point of the time interval of received packets which are getting started to be changed. The maximum possible data egress bandwidth will be calculated based on the packet size at this turning point.

E-Meter needs no proprietary application to be deployed at the CO/ISP side as long as the ICMP echo reply is supported at the default gateway of CO/ISP. E-Meter also occupies very limited bandwidth to measure the maximum egress data bandwidth compared to the UDP sending/receiving mechanism. Two earlier indicated points are now well addressed on page 11 by the E-Meter by giving better approach to measure the maximum egress data bandwidth.

The model established by E-Meter to measure the egress bandwidth is depicted as below.

- One outbound packet destined to the default gateway on CO side will be generated by E-Meter in each T seconds
- Default gateway on CO side will relay each received outbound packet to the original sender, which is E-Meter

TO: Time interval of contiguous outbound packets
TI_i: Time interval of i^{th} and $(i+1)^{th}$ inbound packets
P: Outbound packet size
O: Outbound traffic rate
B: Egress bandwidth (to be measured)
U_{Max}: Maximum transmission data length
U_{Min}: Minimum transmission data length

O = 1 / TO (packets/sec) * P (bytes/packet) * 8 (bits/byte) = 8 P/ TO (bps)

Since O is proportional to P and inverse proportional to TO, we can either increase P or decrease TO and make a bigger amount of output of O. If we fix the value of TO to be TO_f, the generated outbound rate can be depicted as below:

$O_i = 8P_i / TO_f$

- $U_{Min} \leqq P_i = U_{Min} + (i * K) \leqq U_{Max}$
    - $i \geqq 0$ (i is an integer)
    - K is a constant which is greater than 1

  (Smaller K makes better precision for egress bandwidth calculation but increases time to locate B)

If a given integer j meets the below conditions, B is calculated to be $8P_{j-1} / TO_f$

- $U_{Min} \leqq P_j = U_{Min} + (j * K) \leqq U_{Max}$
- $TO_f = TI_{j-1}$ (equal or pretty close)
- $TO_f < TI_j$

Since SQPV engine was implemented on the Linux operating system and HZ defined in the Linux system provides the frequency of the system timer, which is basically the tick rate. Linux, as well as most other operating systems maintain a sense of time using a periodic interrupt from a timer chip, which is known as the "heartbeat" of the system. The heartbeat of the Linux kernel is 10 ms for the i386. Jiffies is the global variable that holds the number of ticks which start after the system is booted. On each timer interrupt, the value of this variable is increased by

1. Jiffies and HZ are related in the sense that if there are HZ timer interrupts in a second, there are HZ jiffies in a second. The HZ was defined to be 100 and the time interval of each jiffies will be 1 / 100 second = 10 ms, which will be the TO_f as depicted earlier. Due to the fact that the MTU is always limited, the better resolution (beyond 10ms) will directly increase the outbound traffic to be generated and enhance the capability of E-Meter to measure the maximum data bandwidth. However, the temporal granularity of Linux is beyond the scope of this thesis.

3.2.2 I-Police

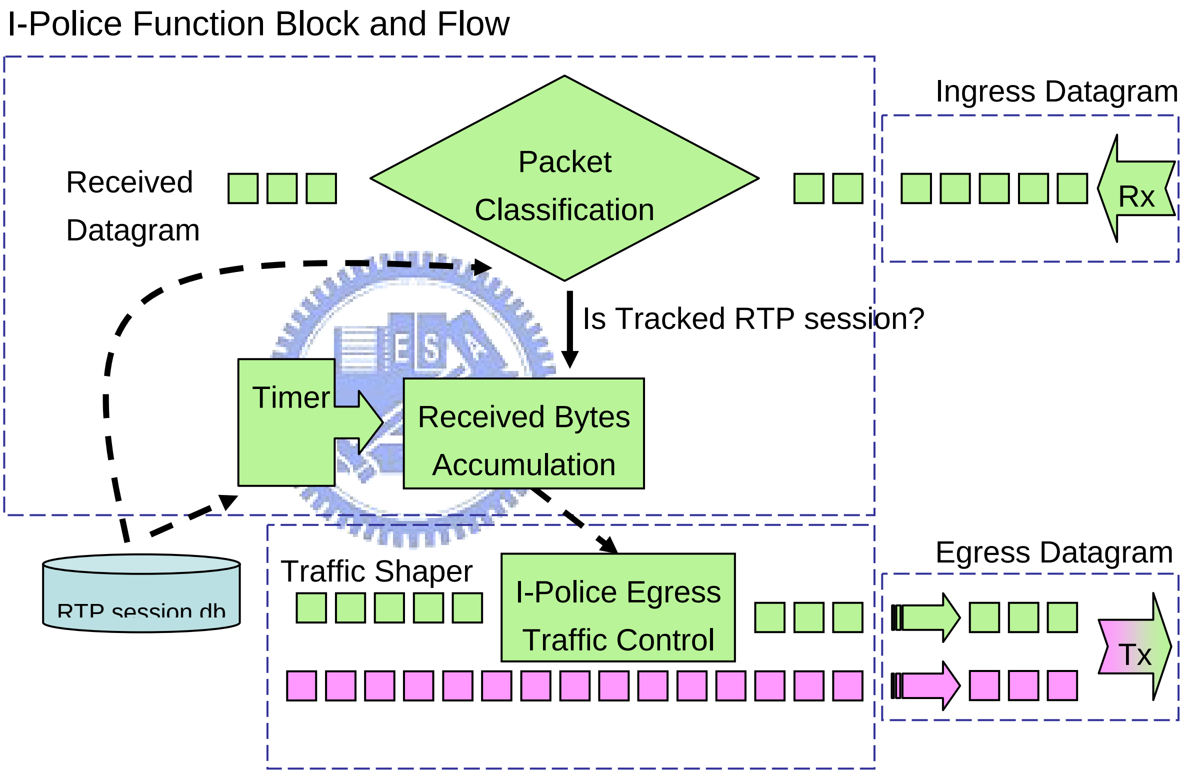


Figure 10 : I-Police Functional Block and Flow

As depicted in Figure 10, I-Police in SQPV engine plays the role to avoid the congestion in the ingress direction and make the incoming TCP traffic not exceed the maximum allowed bandwidth. The incoming lower priority TCP datagram will be slowed down by I-Police once it has detected the received bytes of the established RTP [5] sessions not matching the outgoing bytes of RTP sessions or speeded up otherwise.

Each packet arrives at I-Police will be classified and the amount of received byte of priority datagram will be accumulated. The timer will periodically check the amount of the received bytes of priority datagram and if it’s less than expectation, the I-Police traffic congestion avoidance mechanism will be activated. Please be noted that if the silence suppression is supported and enabled, the congestion may not be able to be detected and activated correctly. The correspondent enhancement on the congestion detection should be happened accordingly, which is outside of the scope of this thesis.

The way I-Police controls the ingress traffic and avoid the congestion (through I-Police traffic congestion avoidance mechanism) is described as below.

- If the local end is the TCP sender and the remote end is the TCP receiver, the advised window size of the TCP datagram sent from local end TCP sender will be decreased to slow down the incoming TCP traffic if the accumulated bytes of priority packets are not matching the expectation.
- If the local end is the TCP receiver and the remote end is the TCP sender, the acknowledgement to be replied by the local end TCP receiver will be removed. The remote end TCP sender will slow down the TCP traffic without receiving the expected acknowledgement.

The ground rule to minimize the impact of incoming lower priority TCP datagram on the priority datagram is to make the advised window size slowly increased and fast decreased. If we plot the current value of the advised window size as a function of time, we get a saw tooth pattern as illustrated in Figure 11. The implementation concept of I-Police is to reduce the advised window size at a faster rate rather than the rate to increase. The reason to decrease the advised window aggressively and increase it conservatively is that the consequences of having too large advised window’s size is worse than the one being small from the perspective of the better QoS provisioning for the priority datagram. Please be noted that the I-Police won’t work with TCP Vegas.

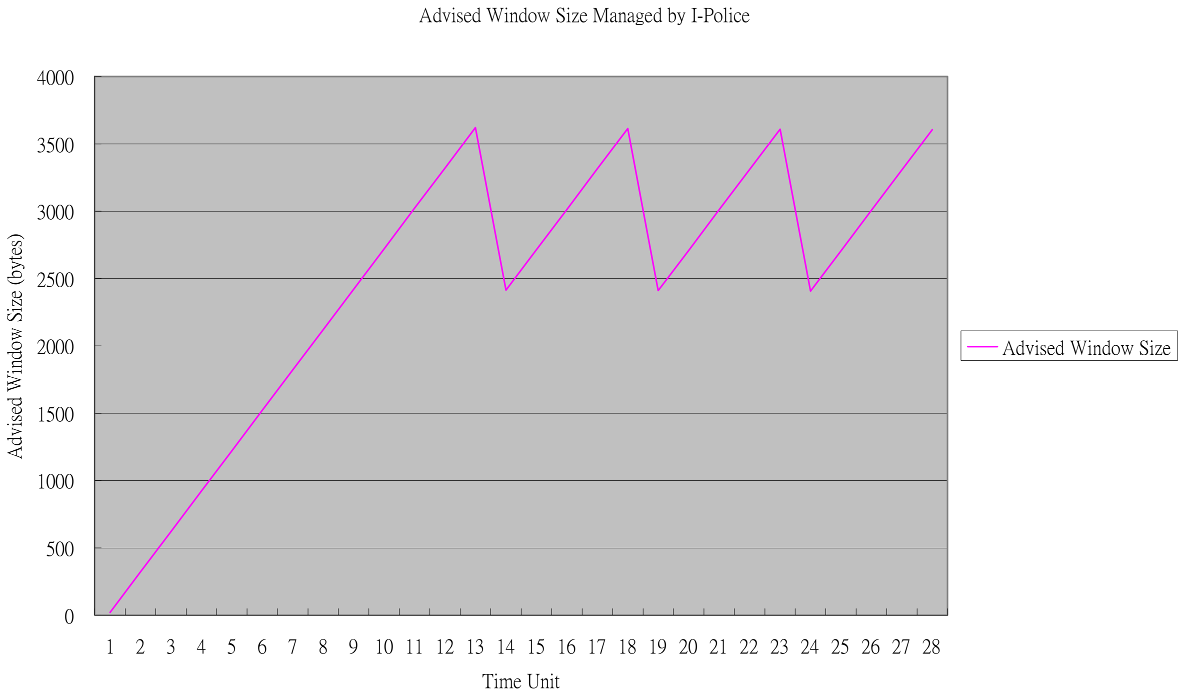


Figure 11 : Advised Window Size Managed by I-Police

3.2.3 Traffic Classifier

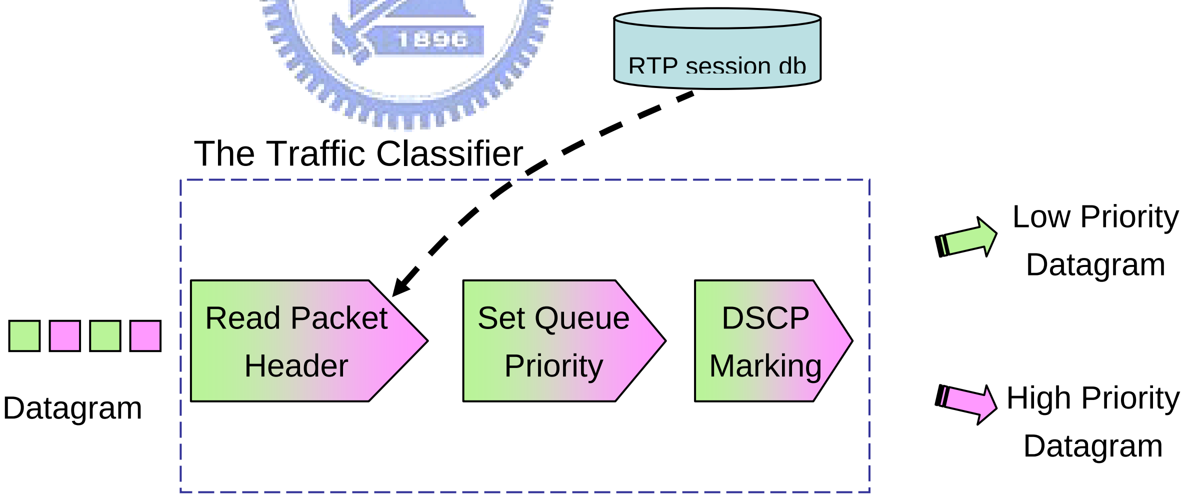


Figure 12 : Traffic Classifier Function Blocks

Traffic Classifier in the SQPV engine will classify the outgoing datagram according to the type of each packet. If the packet type is the UDP/RTP, it'll refer to the RTP session db and check if the source IP, source port, destination IP and destination port match one of the established sessions. If it matches, the packet will be classified into the priority packet by setting the queue priority to high and inserting the packet into the high priority queue in order to make sure it can be serviced in the egress direction first. Two priority queues (high and low) are implemented accordingly. This allows for the prioritization of the egress traffic. At any time instant, the low priority traffic can only be sent if there is no high priority traffic needs to be serviced. That is, any packet remains in the low priority queue won't be serviced if the high priority queue is not empty. All of the packets remain in the low priority queue will be serviced in best-effort mode.

In order to make sure each priority egress packet can be processed first after being sent out from the SQPV RG to the Diffserv service domain as illustrated in Figure 7, the classifier needs to fill in the 6-bits DSCP (Diffserv Code Point) field in every egress priority packet and make sure it'll be forwarded to the destination with the minimum delay

3.2.4 Traffic Parser

The signaling protocol supported by SQPV engine is SIP (Session Initiation Protocol) [8]. SIP is a light weight and transport independent protocol which is widely used currently. It acts as a carrier for the SDP (Session Description Protocol) [3] which describes the media content of the session. The sessions of SIP are the packet streams of RTP (Real-Time Protocol) which is the carrier for the actual content to be transported.

SDP provides a standard representation to convey media details, transport addresses, and other session description irrespective of how that information is transported. It intends to be general purpose and uses different transport protocols as, including the SAP (Session Announcement Protocol), SIP (Session Initiation Protocol), RTP (Real Time Streaming Protocol), MIME based electronic mail and the Hypertext Transport Protocol. The SIP messages are used to create sessions and carry session descriptions that allow participants to agree upon a set of compatible media types. These session descriptions are commonly formatted using SDP.

An SDP session description includes the following:

- Session name and purpose
- Time while the session is active
- The media comprising the session
- Information needed to receive those media which includes the addresses, ports, formats, etc.
- Information about the bandwidth to be used by the session (optional)

- Contact information for the persons responsible for this session.

The SIP (Session Initiation Protocol) is the signaling protocol developed to set up, modify and tear down the multimedia session and instant message over the internet. The main functions of the signaling protocol are to locate the end point, contact the end point to determine the willingness to establish the session, exchange the media information and modify the existing media sessions. Figure 13 below shows a simple SIP session establishment example.

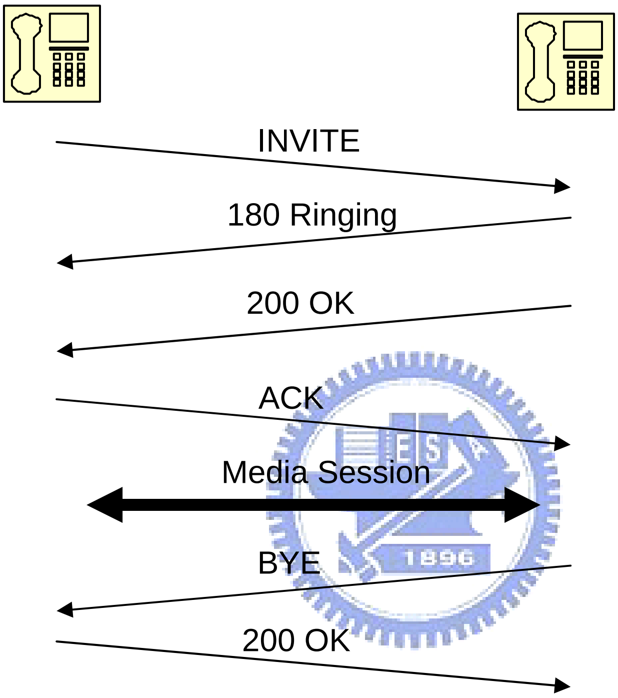


Figure 13 : SIP Session Establishment

The calling party will begin the message exchange by sending the SIP INVITE message to the called party. The INVITE message contains the details of the type of the session that's requested. The Via, Max-Forwards, To, From, Call-ID, and CSeq header fields represent the minimum required header field set in any SIP request message.

The "Via" header contains the SIP version number, the transport method, the host name (or IP address) and the port number. The "branch" parameter is a transaction identifier and the response can be correlated with the request if they have the same transaction identifier.

The "Max-Forwards" header field contains the integer and is decremented by each

SIP server which receives and forwards the request. The loop detection mechanism can be provided based on this integer.

The "To" and "From" header fields show the originator and destination of the SIP request. The "tags" is a random identifier for each party in the "To" and "From" header fields of the session.

The "Call-ID" header field is an identifier to track a particular SIP session and it consists of the local unique string generated locally along with the host name to make it globally unique.

The "CSeq" contains a number followed by the method name, INVITE in this case. This number is incremented for each new request sent.

Here below shows an example of SIP/INVITE message:

```
INVITE sip:maquire@nctu.edu.tw SIP/2.0
Via: SIP/2.0/UDP lab.csie.nctu.edu.tw:5060;branch=yoklPlchuRsoq
Max-Forwards: 70
To: J. Maquire <sip:maquire@nctu.edu.tw>
From: Dan Karr <sip:d.karr@nthu.edu.tw>;tag=89763
Call-ID: 1122345@nthu.edu.tw
CSeq: 1 INVITE
```

Here below is an example of the SIP message body which is SDP. The key information delivered in the message body are the connection IP address (100.102.101.100), media type (audio), port number (49170), media transport protocol (RTP), media encoding (PCMU) and the sampling rate (8000).

```
v=0
o=Dan 4098765409 0976409432 IN IP4 nthu.edu.tw
s=Phone Call
c=IN IP4 100.102.101.100
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

The "180 Ringing" SIP response message, which is informational response used to convey the non-critical information about the progress of the call.

When the called party decides to accept the call, a "200 OK" SIP response message is sent. The "200 OK" message body contains the media information like end-point IP address, media format, port number, media transport protocol, media encoding and the sampling rate which is communicated in a SDP message body.

After receiving the "200 OK" SIP response message sent from called party, the acknowledgement is sent by the calling party to confirm the media session.

If the called party decides to terminate the call, the "Bye" SIP request is sent to the calling party and the "200 OK" SIP response message is sent to confirm the termination of the call.

Traffic parser, as illustrated in Figure 14, plays the role of admission control for the egress traffic and makes the decision whether the new SIP session is allowed to be established or not. It captures the SIP messages and fetches the identification of the SIP session, the IP address and port number of the transport media in order to make sure whether the session has been established and tracked in the RTP session db. If it’s matched one of the established tracked sessions in the RTP session db, the correspondent SIP messages will be allowed to come in or be sent out. If this is a new SIP session which cannot be found in the existing RTP session db, Traffic Parser will need to figure out whether there is enough egress bandwidth available for this new SIP session. If the left available bandwidth is enough for the new SIP session to be established, the Traffic Parser will update the RTP session db by adding the new SIP session, let the correspondent SIP messages go and make it be able to complete the SIP session establishment. If it's not allowed due to the lack of available egress traffic bandwidth, the Traffic Parser will block the correspondent SIP messages of this session and make sure the new SIP session is impossible to be established.

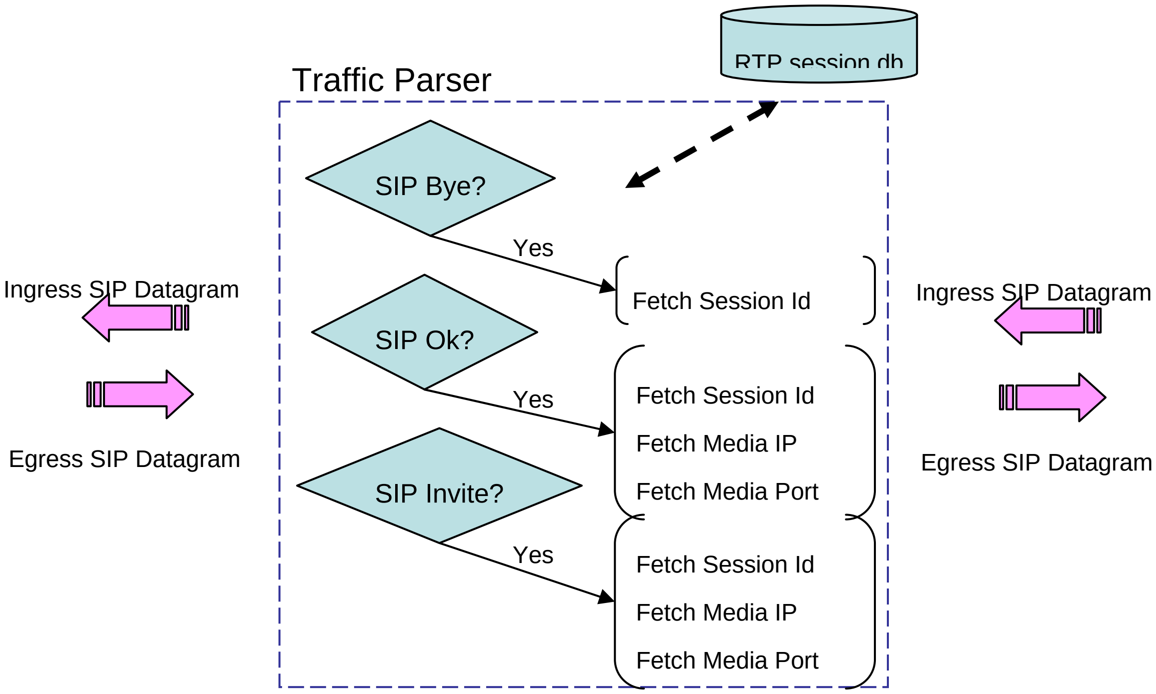
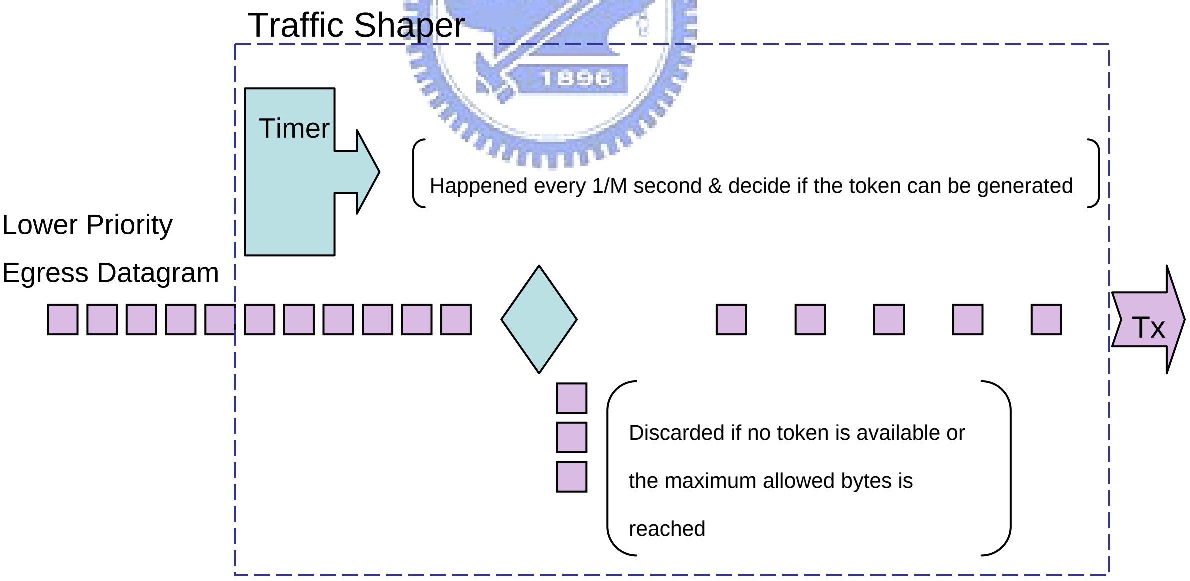


Figure 14 : Traffic Parser Functional Block

3.2.5 Traffic Shaper


Traffic Shaper
Timer
Happened every 1/M second & decide if the token can be generated
Lower Priority
Egress Datagram
Tx
Discarded if no token is available or
the maximum allowed bytes is
reached

R: Tokens generated per second.

B: The maximum number of token that the bucket can hold

M: The maximum possible number of token generated per second

Figure 15 : Traffic Shaper Functional Blocks

The “Random Prorated-Drop Token Bucket” (RPDTB) is the algorithm invented to support Traffic Shaper to limit data transmission rate of the non-priority traffic as illustrated in Figure 15.

- On the platforms lacking the better clock resolution: The established working model can work out no matter the clock resolution is better or worse.
- Packet based check instead of byte based check: Each packet (instead of byte) will be checked before it’s sent out.

Here below are the parameters defined and used in RPDTB algorithm.

- Token rate R: The number of token generated per second.
- Bucket depth B: The maximum number of token that the bucket can hold.
- Maximum token rate M: The maximum number of token generated per second.
- Polling interval T: Timer interval to check the status of token.

One available token can allow N bytes of packet to be sent. The maximum rate of transmission is (M * N) bytes per second and the allowed rate of transmission is (R * N) bytes per second according to the given token rate R.

The algorithm can then be conceptually described as below.

- During the period of each T, the random integer number x is generated in the range from 1 to 100 prior to each packet to be sent and the packet is allowed to be sent out only if
    - x ≦(100*R)/M and
    - The accumulated amount of bytes during time interval T is smaller than (R * N * T)
- Bucket depth B is configured R.

Compared to token bucket, RPDTB can provide a fair distribution for the packets transmitted during the period of T, regardless of the resolution of T.

3.2.6 RTP session db

RTP was developed by IETF. Two protocols, RTP and RTCP, are defined in the IETF standard. RTP is used for the exchange of multimedia data and RTCP is used to periodically send the control information associated with a certain data flow. RTP can provide the ability to negotiate the choice of the coding scheme and identify the type of application (voice or video).

<table>
<tr><td>V</td><td>P</td><td>X</td><td>CC</td><td>M</td><td>PT</td><td>Sequence Number</td></tr>
<tr><td colspan="7">Timestamp</td></tr>
<tr><td colspan="7">Synchronization Source Identifier</td></tr>
<tr><td colspan="7">Contributing Source Identifier</td></tr>
<tr><td colspan="7">Extension Header</td></tr>
<tr><td colspan="7">RTP Payload</td></tr>
</table>

Figure 16 : RTP Header Format

The header format of RTP packets is described in Figure 16. The first 2 bits (V) are the version identifier. The next bit is the padding (P) bit which is set in circumstances in which the RTP payload has been padded. The extension (X) bit is used to indicate the presence of an extension header. CC (4 bits) is used to count the number of the contributing source if any are included in the header. The mark bit (M) is for the frame indication. The payload type (PT, 7 bits) indicates what type of multimedia data is carried in this packet. The sequence number is used to enable the receiver of an RTP stream to detect missing and mis-ordered packets. The timestamp is to enable the receiver to play back samples at the appropriate interval and make different media streams to be synchronized. The synchronization source identifier is a 32-bit number which can uniquely identify a single source of an RTP stream. The contributing source is used only when a number of RTP streams pass through a mixer, which is used to reduce the bandwidth requirements for a conference by receiving data from many sources and sending it as a single stream.

One of the three major functions provided by the control protocol is to the feedback on the performance of the application and the network. This is quite useful for the rate-adaptive applications which may use performance data to decide using a more aggressive compression scheme to reduce congestion or higher-quality stream when there is little congestion.

RTP session db is a double linked list data structure that keeps track of source IP, source port, and destination IP, destination port of the established SIP/RTP sessions. There is also a timer associated with each session to make sure that the timed-out sessions can be removed from the RTP session db accordingly. The RTP session db will be referred by I-Police, Traffic Parser and Traffic Classifier in the SQPV engine.

4 Test Setup and Result Analysis

4.1 Goals

Voice traffic is particularly time-sensitive, which cannot be queued long. In case that the voice datagram gets lost during transmission, the conversation will be broken and choppy. VoIP applications are also very throughput sensitive and CPU-intensive. They'll almost disrupt other traffic on your network once they're in use.

The main goal of the tests trying to carry out is to evaluate how well it's possible for the SQPV engine to control the process for the SIP/RTP priority traffic in both egress and ingress direction. The key indicators which help quantify and determine the overall output of SQPV engine are described as below.

- The data throughput which includes: average, minimum and maximum throughput
- Lost data, which includes the bytes sent, bytes received, bytes lost and consecutive lost datagram
- One-way-delay (or network delay) is composed of the average, minimum and maximum end-to-end delay

The data throughput is calculated by measuring the total number of bytes sent and received by the two endpoints divided by the elapsed time. For the multiple established data session, the average data throughput, minimum and maximum data throughput will be calculated and shown accordingly.

The lost data is shown by the bytes lost sent from endpoint 1 to endpoint 2 for a test consisting of pairs running data streaming. The consecutively lost datagram means how often data losses within a certain period occurred and it's placed into ranges based on the number of consecutive loss and is shown by histogram.

One-way-delay (or network delay) is one of the major key factors in determining the quality of time-sensitive applications and composed of processing delay, queuing delay, transmission delay and propagation delay.

4.2 Test Setup

The test setup was illustrated in Figure 17 which is an isolated network consisting of

5 Intel Pentium based PCs. The prototype of SQPV enabled Residential GW (SQPV RG) was implemented on the Infineon® XWAY™ TWINPASS-VE reference platform, which is MIPS® 24KEc™ based network processor running Linux (kernel version 2.4.31) as a kernel loadable module as showed in Figure 18.

In order to emulate the limited data bandwidth in both downstream and upstream directions and the end-to-end delay between host C and host E(or host C and host F), the NIST Net [9] was introduced as a network emulation package to emulate the end to end performance characteristics. Host D is running Linux (kernel version 2.4.20) with NIST Net installed to emulate the real internet service provider by having the fixed bandwidth. It's also emulating the real internet network situation by having the fixed end-to-end delay. The fixed bandwidth was emulated to be 1 Mbps in both downstream and upstream directions and it was used to crosscheck the result that E-Meter demonstrated. The end-to-end delay was emulated to be 50 ms, which was the delay between each two endpoints connected to host D. If we're presuming the processing, queuing and the transmission delays on host A, B, E and F as depicted in Figure 17 are small enough to be ignored, the one way delay can be calculated as shown below:

One way delay (or network delay) =
(Processing+Queuing+Transmission) delays on the RG +
(Processing+Queuing+Transmission+Propagation) delays on the emulated Internet

As indicated earlier, the processing, queuing, transmission and propagation delays on the emulated Internet was fixed to be 50 ms, the one way delay will depend on the processing, queuing and transmission delays on the RG. We used it to demonstrate the result how SQPV was able to manage the one way delay by controlling the processing, queuing and transmission delays on the RG.

The NetIQ Chariot was used to generate multiple RTP sessions between host B and host E in both egress and ingress direction. In order to evaluate, measure and quantify how SQPV engine capable of QoS provisioning, the Iperf [11] was used to generate the UDP datagram in egress direction and the FTP was used to send and receive the TCP datagram in both egress and ingress directions to compete with the RTP traffic for acquiring available system and network resource. Host A and host F are the two endpoints with Iperf and FTP client/server installed to emulate the TCP/UDP datagram.

To evaluate the functionality and accuracy performed by Traffic Parser in SQPV engine to recognize the new coming SIP sessions, the SIPp [10] was introduced. SIPp is a test tool for traffic generation of SIP protocol which is able to establish and release multiple calls with the SIP INVITE and BYE methods.

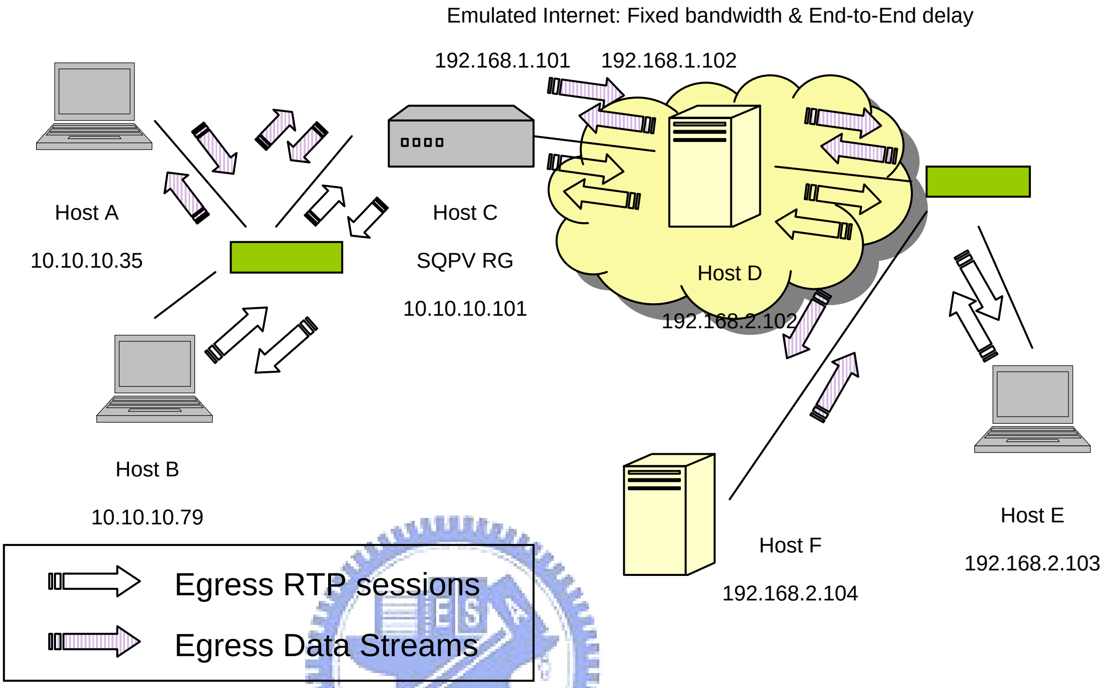


Figure 17 : Test Setup

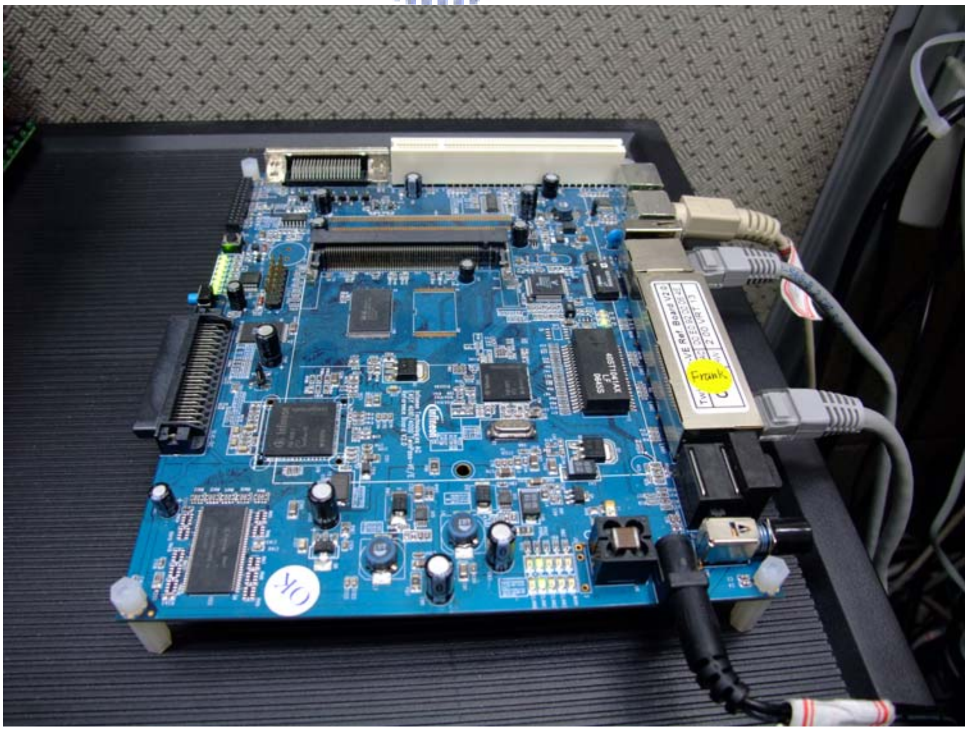


Figure 18 : SQPV Prototype

4.3 Test Result and Analysis

4.3.1 Egress Traffic Bandwidth Measurement - E-Meter

1024 Kbps in both downstream and upstream direction was preconfigured at NIST Net to emulate the fixed bandwidth in downstream and upstream direction on the WAN side. E-Meter will generate the packets with different packet size in every predefined time interval, which is described in previous chapters to measure and approach the maximum possible egress bandwidth. As illustrated in Figure 19, the time intervals of sent packets and received packets were shown in the Y-axis and the packet size of each packet sent and received were shown in the X-axis.

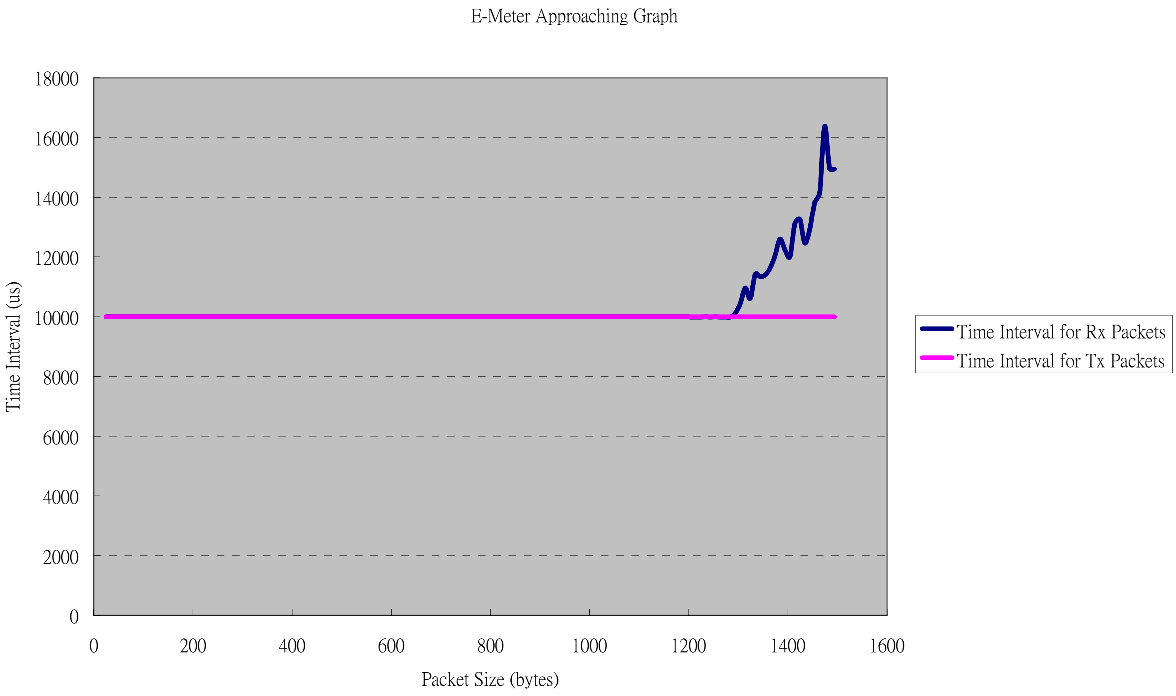


Figure 19 : E-Meter Egress Data Bandwidth Approaching Graph

As per the established model of E-Meter to measure the egress bandwidth, TO (The time interval of each two contiguous generated outbound packets) needs to be properly defined. As indicated earlier, SQPV engine was implemented on the Linux operating system and HZ defined in the Linux system provides the frequency of the system timer, which is defined to be 100 on i386. So each packet will be sent out in every 10 ms (= 10000 us), for all different possible packet sizes (from 64 bytes to 1500 bytes for Ethernet packets).

TO_f = 10000 us
The maximum/minimum transmission length is 1518/64 for Ethernet.
K is defined to be 10

P_i = (64 + i * K) bytes for i from 0 to k can be calculated and TI can be measured as showed in Figure 20. By observing the timer interval of the received packets, TI starts getting increased at 1294 bytes and the maximum egress data bandwidth will be:
$8P_i$ / TO_f = 8 * (1294 – 10) / 10000 (us) = 1027200 (bits/sec).

1,027,200 bps is pretty close to the original preconfigured value of egress bandwidth by NIST Net) and shows the bandwidth measured by E-Meter is pretty close to the actual egress bandwidth.

Please be noted that there is one prerequisite assumption that the maximum ingress data bandwidth should be no less than the maximum egress data bandwidth, and this is true in most residential environments. Otherwise the bottleneck of the data flow will be in the ingress direction and make it not possible to approach the maximum egress data bandwidth as we expected.

i	P_i (bytes)	TI_i (us)	TO_f (us)	Packet Loss (byte)
112	1184	9998	9999	0
113	1194	9998	9999	0
114	1204	9985	9999	0
115	1214	9985	9999	0
116	1224	9985	9999	0
117	1234	9999	9999	0
118	1244	9985	9999	0
119	1254	9998	9999	0
120	1264	9985	9999	0
121	1274	9985	9999	0
122	1284	9985	9999	0
123	1294	10121	9999	0
124	1304	10446	9999	0
125	1314	10960	9999	0
126	1324	10608	10000	0
127	1334	11421	9999	0
128	1344	11340	9999	0

129	1354	11394	9999	0
130	1364	11624	10000	0
131	1374	12045	9999	0
132	1384	12600	9998	0
133	1394	12235	9999	0
134	1404	12017	9999	0
135	1414	13115	9998	0

Figure 20 : Packet Size and Time Interval Table

4.3.2 Egress Traffic Shaping/Classification – Traffic Classifier/Shaper

Five test scenarios (from A to F) were defined, created and executed to show how the RTP egress traffic was impacted by injecting the UDP and TCP datagram and what level of improvement could be given by introducing SQPV

There were 5 pairs of RTP sessions predefined and established in egress direction (from host B to host E). The source and destination port of each session were different. The Codec preconfigured in each RTP session is G.711, which requires 64 kbps (64000 bps) per session. The end-to-end delay on the WAN side was emulated to be 50 ms by NIST Net at host D.

To emulate the non-priority UDP datagram in egress direction, the Iperf client was installed on host A to generate the fixed packet size (1470 bytes per packet) and fixed bandwidth (1 Mbps) egress UDP datagram towards host F as the destination with Iperf server installed. The non-priority TCP datagram will be emulated by having the FTP client installed at host A which connects to and upload the file to host F with FTP server installed, and the Iperf client/server installed at host A and host F.

Please be noted that the duration of each test was preconfigured to be one minute for each different scenario described as follows, all of the measurement result shown after 1 minute of elapsed time is ignored.

Scenario-A: 5 RTP sessions only

Five RTP sessions were created for generating the traffic in egress direction from host A to host F and no any other non-priority traffic was generated to compete with these 5 established RTP sessions. The amount of bandwidth required for these five RTP sessions is only 320 Kbps, which is less than the emulated maximum egress bandwidth 1 Mbps. The data throughput for each session was 64Kbps, as shown in Figure 21.

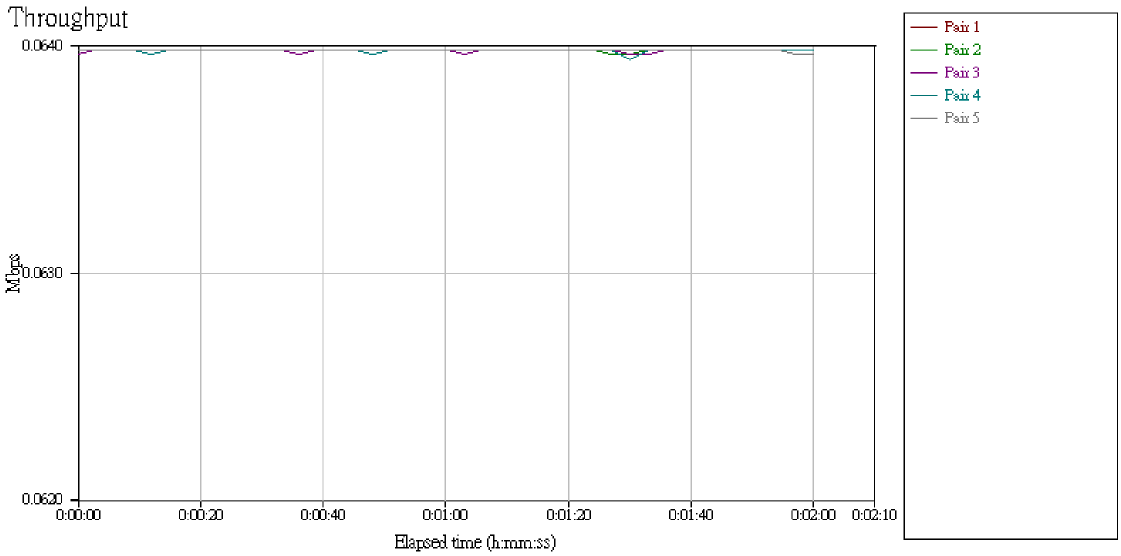


Figure 21 : Throughput Result Graph for scenario-A

The original emulated one-way end-to-end delay on the WAN side is 50 ms and the measured one-way delays are in the range between 51 ms and 56 ms as shown in Figure 22, which are quite close to the original preconfigured one-way delay 50 ms.

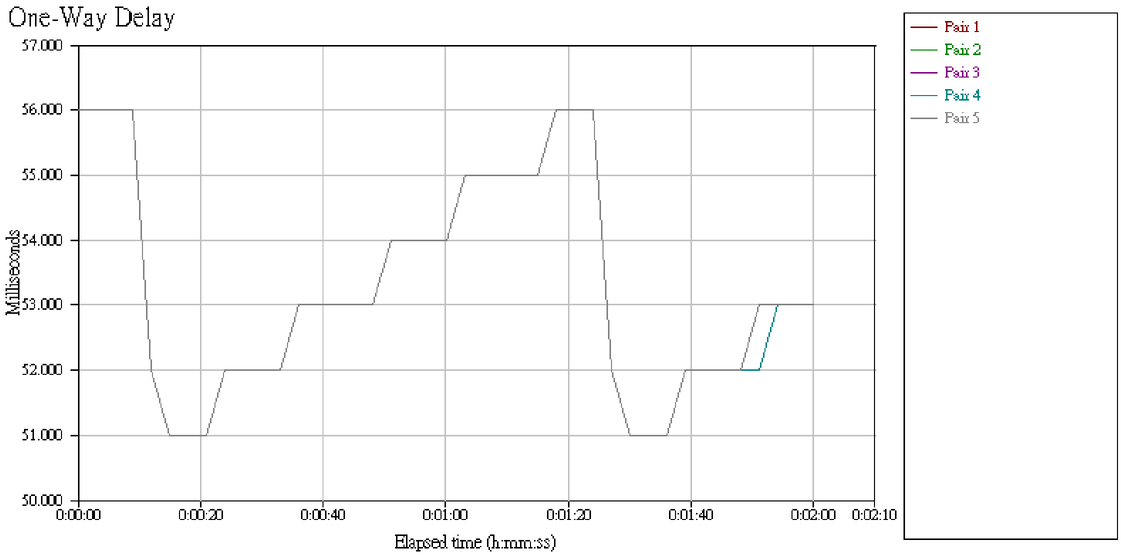


Figure 22 : One-Way Delay Result Graph for scenario-A

The lost datagram and the consecutive lost datagram shown in Figure 23 and Figure 24 are zero, which also meets the original expectation.

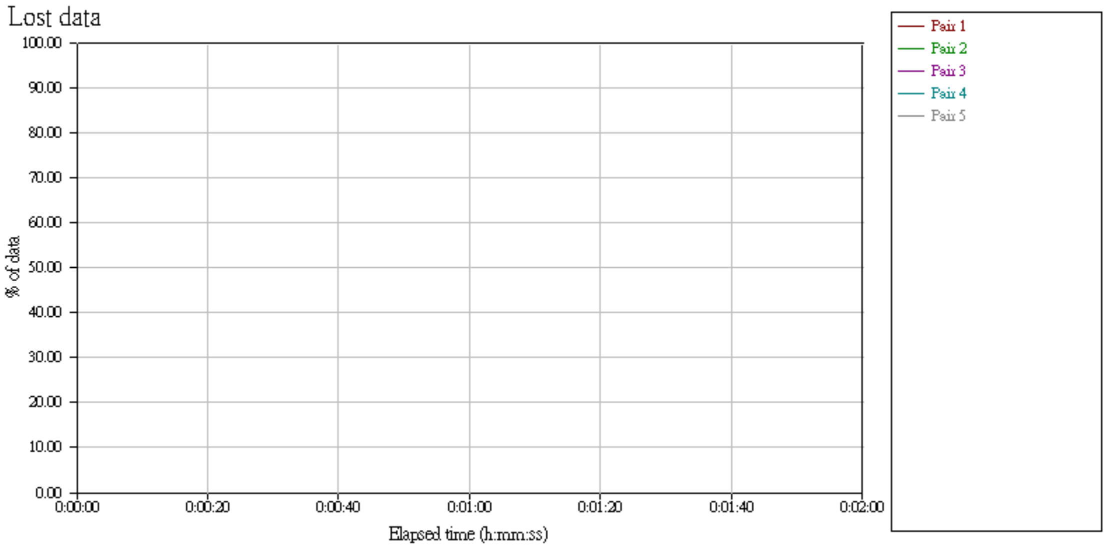


Figure 23 : Data Lost Result Graph for scenario-A

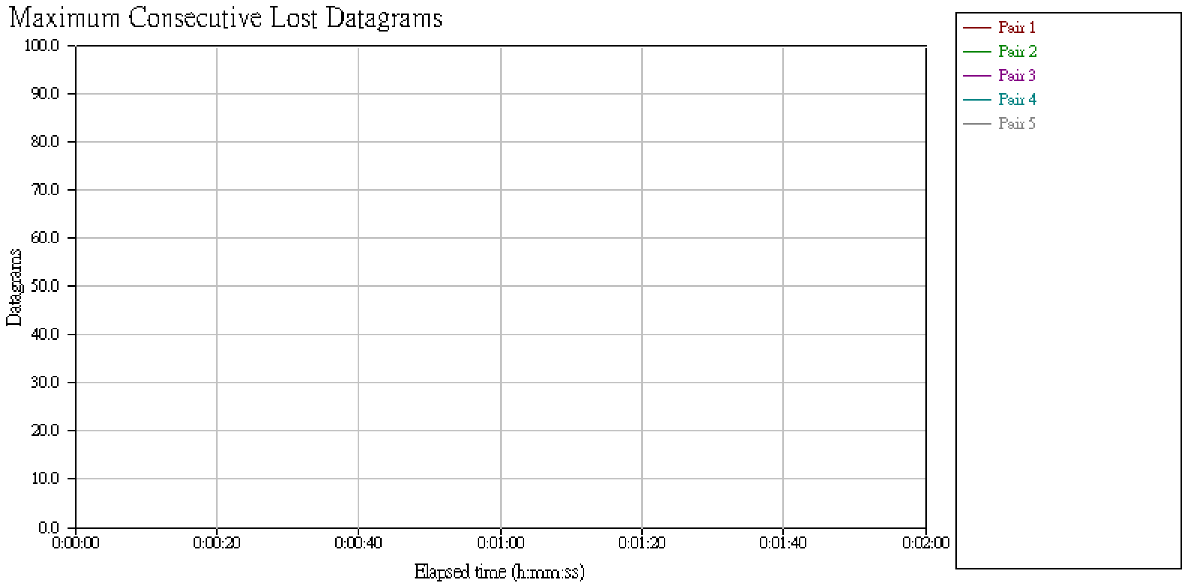


Figure 24 : Maximum Consecutive Lost Datagram Graph for scenario-A

Scenario-B: 5 RTP sessions along with UDP datagram

Besides five same RTP sessions as indicated in scenario-A, the UDP datagram were injected to compete against each other in the egress direction. In the case the UDP datagram was sent from the source endpoint host A to the destination endpoint host F at the fixed rate 1 Mbps, only around 47 Kbps can be reached for each RTP session as shown in Figure 25.

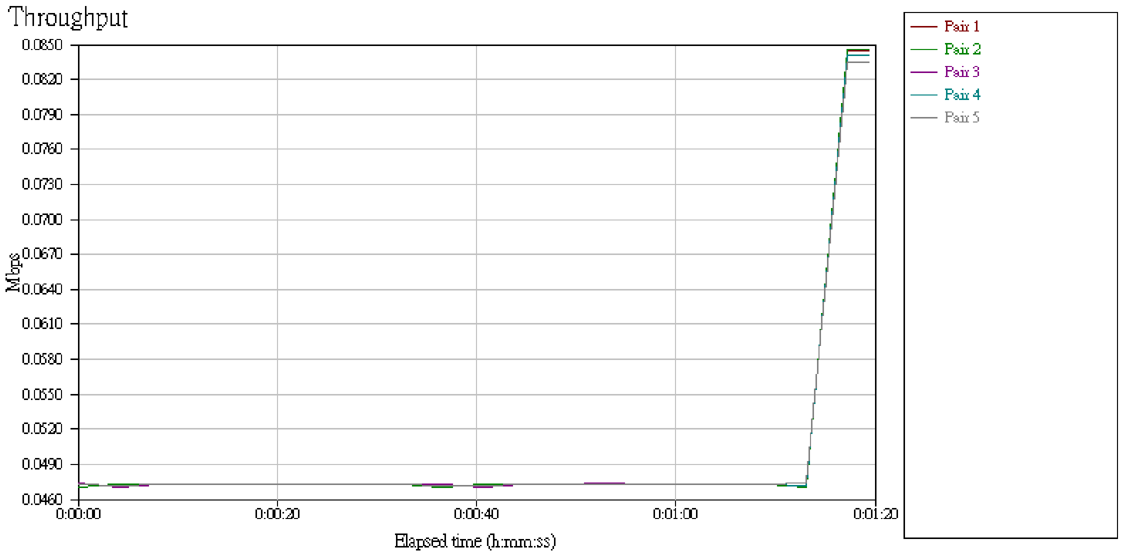


Figure 25 : Throughput Result Graph for scenario-B

The measured one-way delays were increased significantly from 590 ms to 16,590 ms as showed in Figure 26. For the normal audio application, it’s hard to carry a conversation if the time between a speaker and a listener is more than 300 ms makes it unacceptable for making VoIP calls with quality of service.

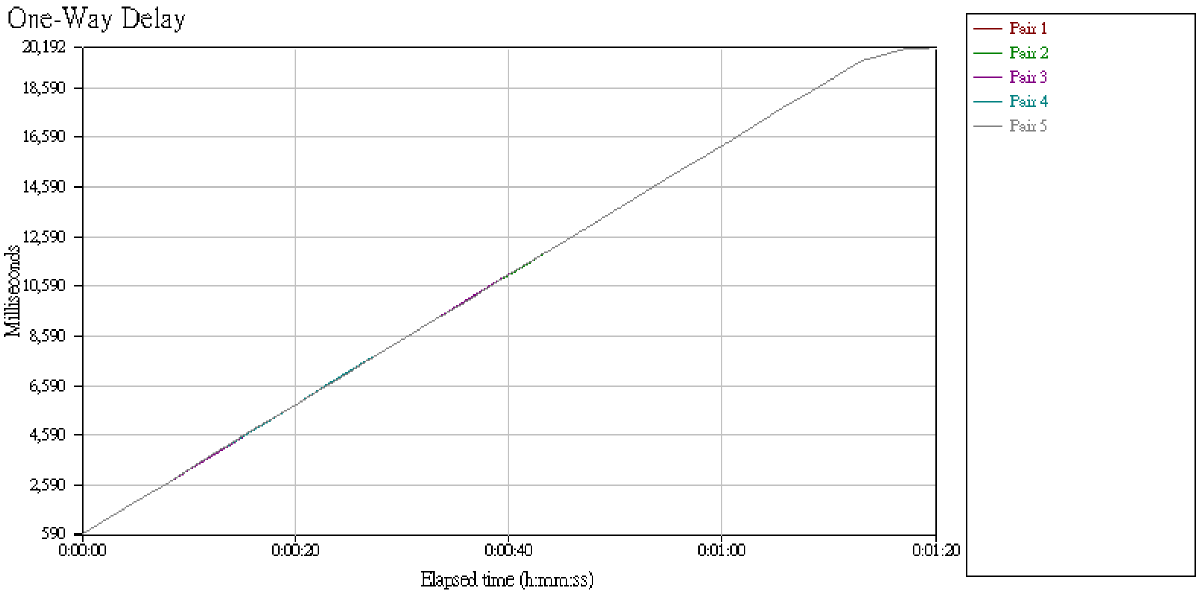


Figure 26 : One-Way Delay Result Graph for scenario-B

The lost datagram and consecutively lost datagram shown in Figure 27 and Figure

28 are also zero, which shows that the length of both transmitting and receiving queues in all network nodes are sufficient large to accommodate the deferred datagram.

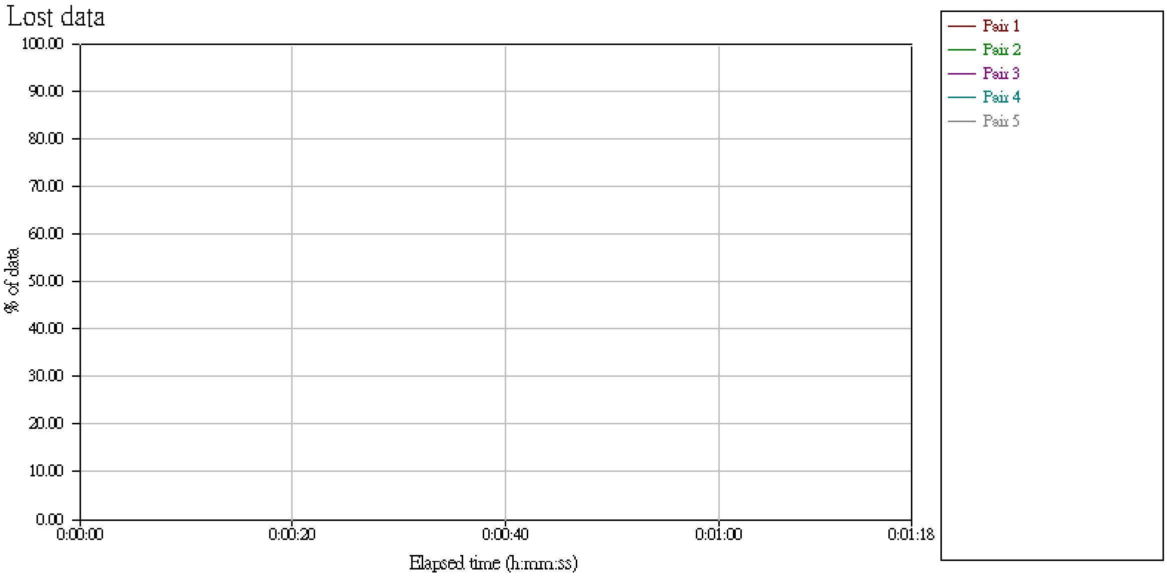


Figure 27 : Data Lost Result Graph for scenario-B

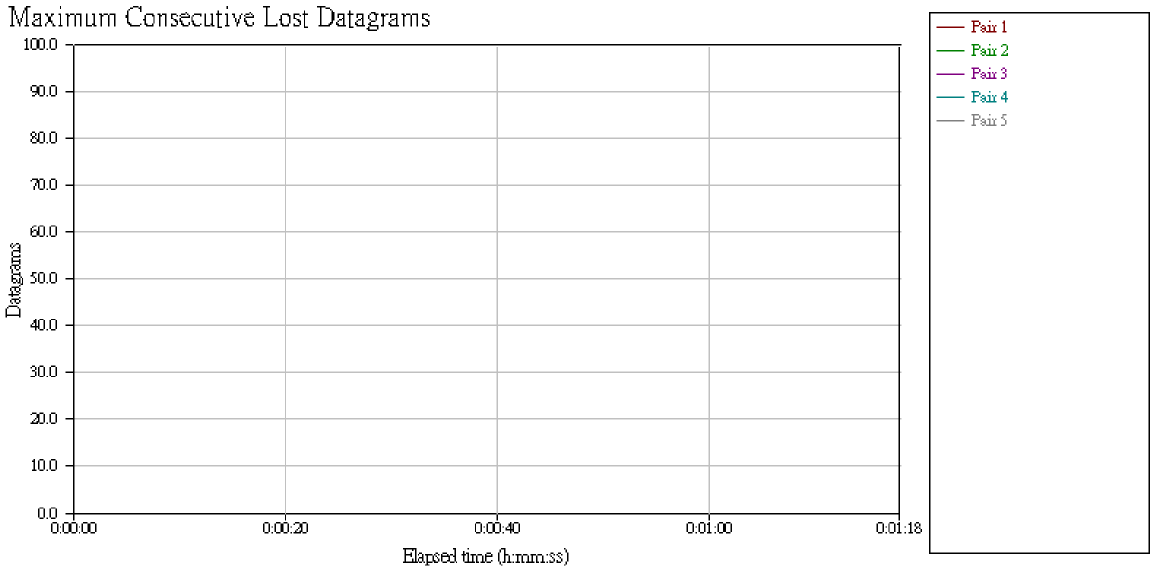


Figure 28 : Maximum Consecutive Lost Datagram Graph for scenario-B

The bandwidth of the non-priority egress UDP datagram keeps decreasing from 1 Mbps at the very beginning and sustains between 729 Kbps and 753 Kbps as shown in Figure 29. Since the maximum egress bandwidth is only 1 Mbps, each RTP session won’t be expected to have enough bandwidth (64 Kbps) to support the voice transmission quality.

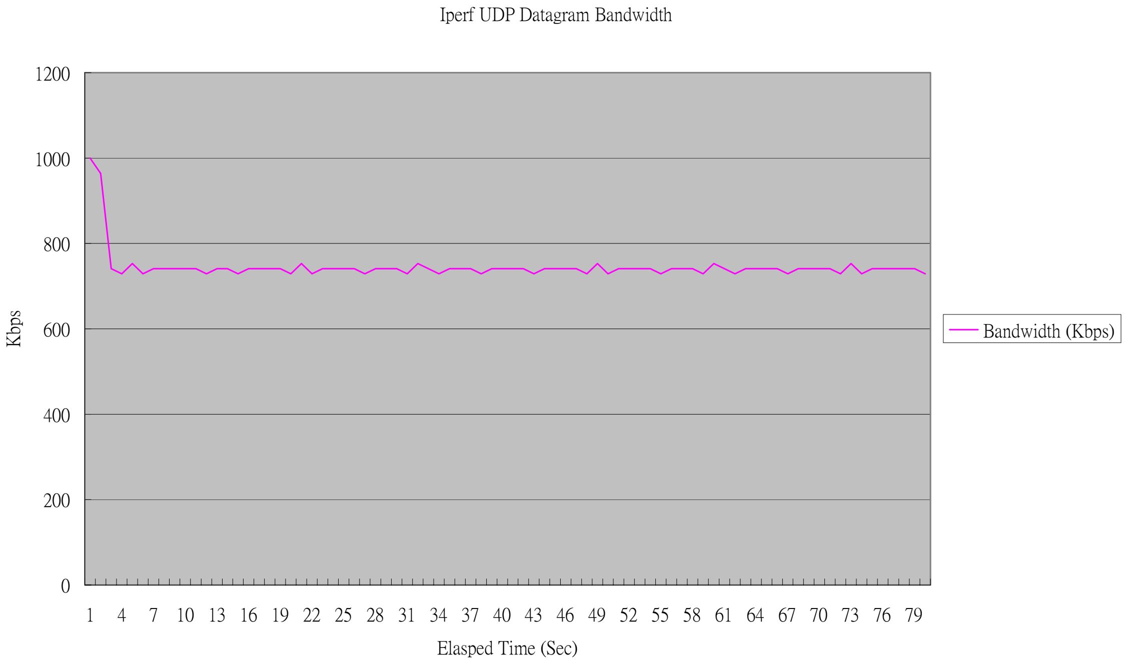


Figure 29 : Iperf UDP Datagram Bandwidth for scenario-B

The zero packet loss percentage of the egress UDP datagram generated by Iperf as depicted in Figure 30 can be explained due to the length of both transmitting and receiving queues in all network nodes are large enough to accommodate the deferred datagram.

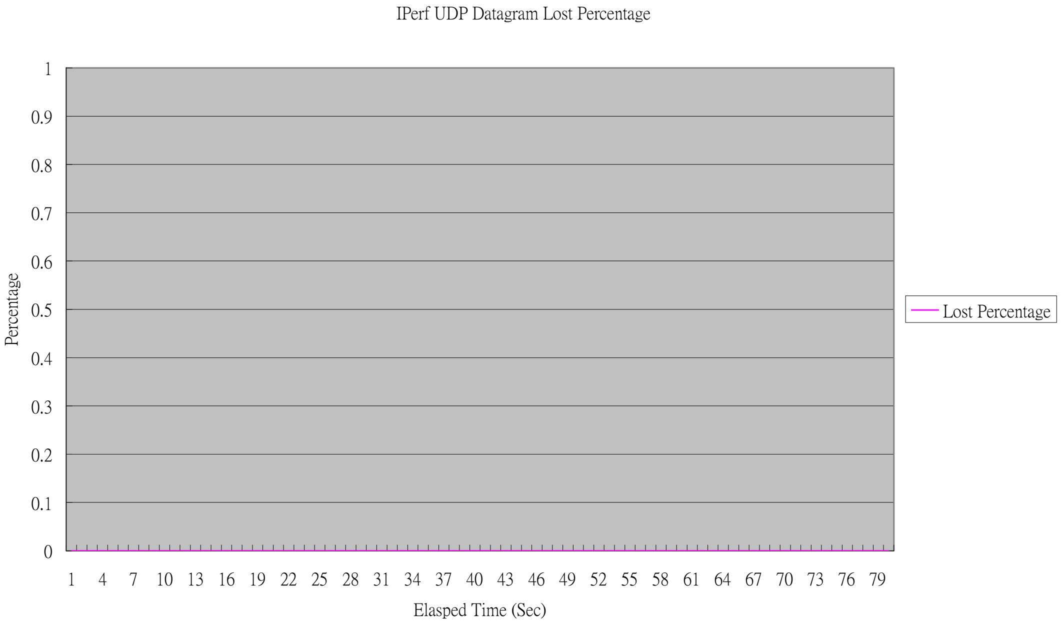


Figure 30 : Iperf UDP Datagram Lost Percentage for scenario-B

Scenario-C: Five RTP sessions along with single FTP session

Single FTP session to upload files from host A to host F was added together with 5 RTP sessions. The big variation in throughput numbers (between 59 Kbps and 86 Kbps) for each RTP session was observed and shown in Figure 31.

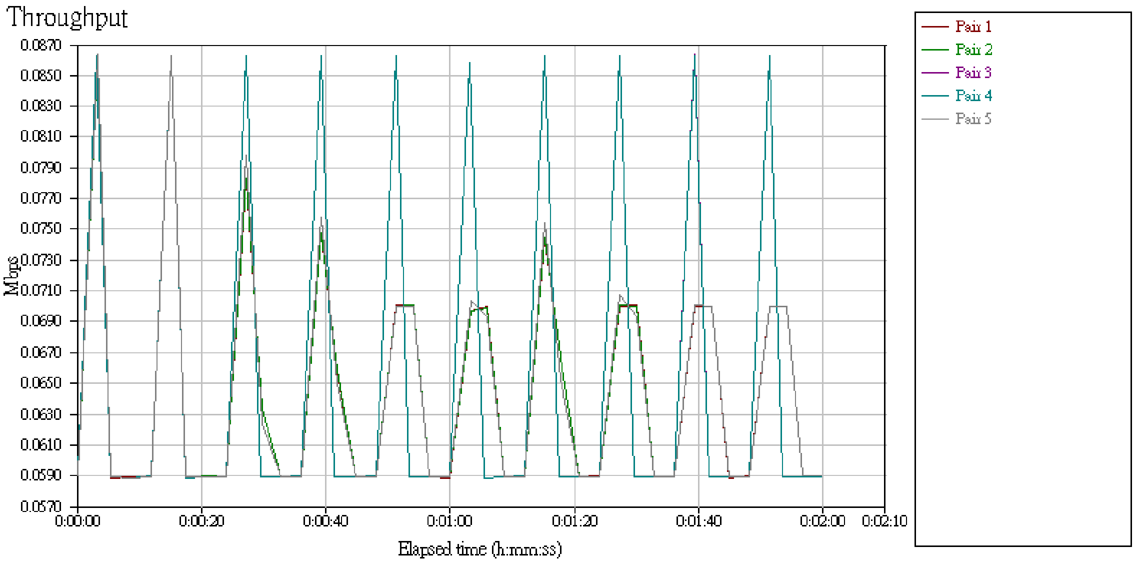


Figure 31 : Throughput Result Graph for scenario-C

The one-way delays were observed in the range between 745 ms and 820 ms as shown in Figure 32. The lost datagram and consecutively lost datagram as shown in Figure 33 and Figure 34 are all zeros, concluded to be the same explanation given in scenario-B.

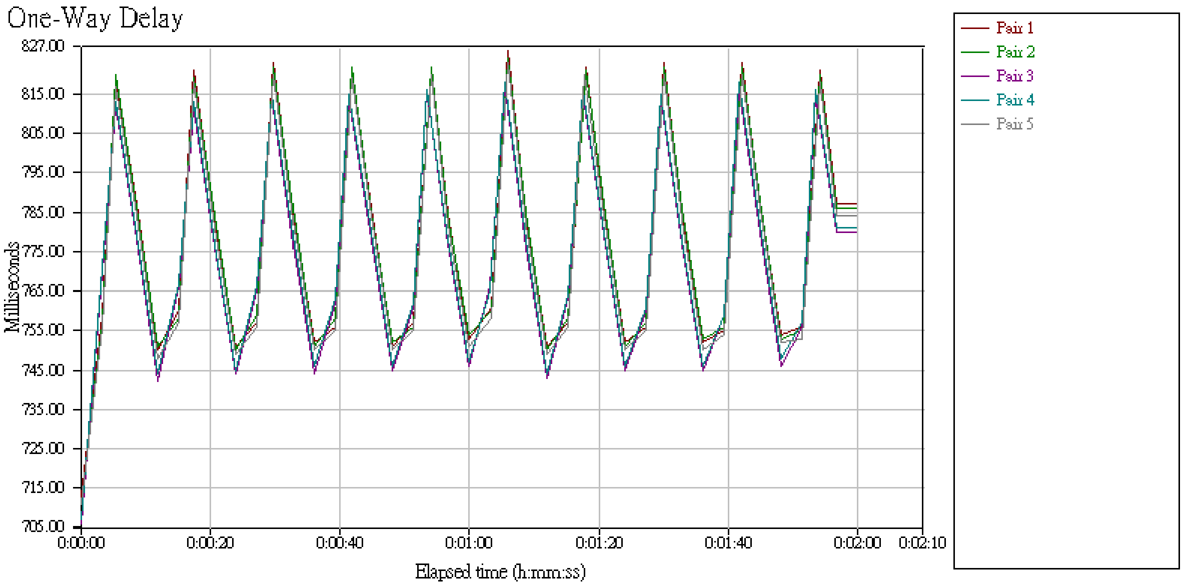


Figure 32 : One-Way Delay Result Graph for scenario-C

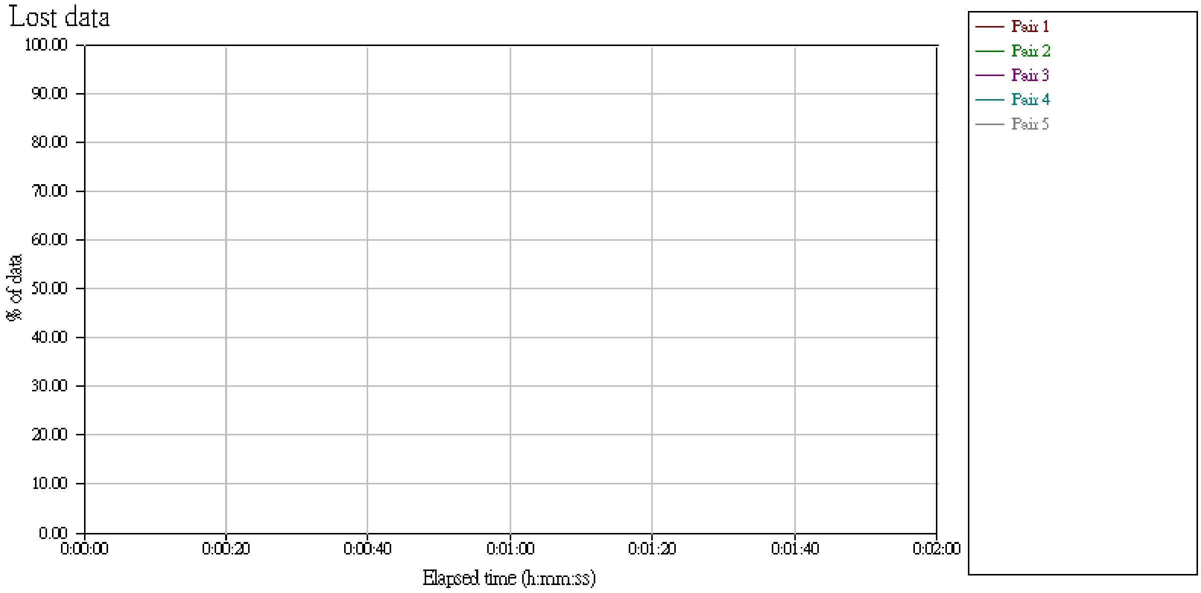


Figure 33 : Data Lost Result Graph for scenario-C

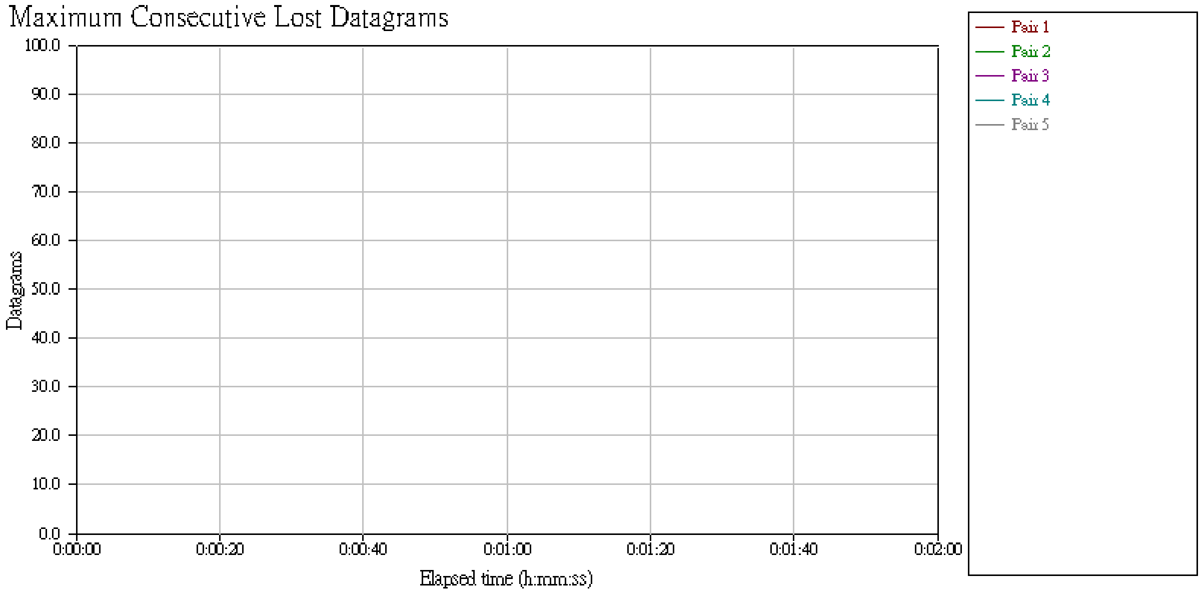


Figure 34 : Maximum Consecutive Lost Datagram Graph for scenario-C

Scenario-D: Five RTP sessions along with UDP datagram with SQPV enabled

In order to crosscheck the difference after the SQPV is introduced, here we kept the same test setup as in scenario-B but made SQPV enabled to redo the same test. The throughput number of each RTP session was observed around 64 Kbps as shown in Figure 35. The measured one-way delays are in the range between 61 ms and 65 ms as shown in Figure 36. The lost datagram and consecutive datagram losses shown in Figure 37 and Figure 38 are still zero. It showed remarkable improvement after SQPV was introduced and enabled.

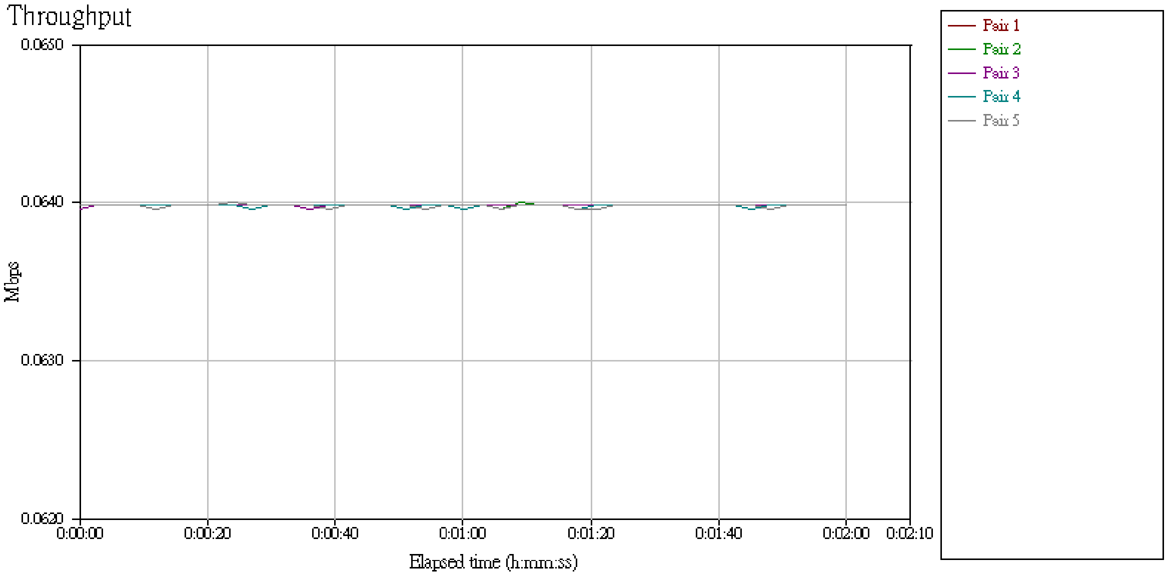


Figure 35 : Throughput Result Graph for scenario-D

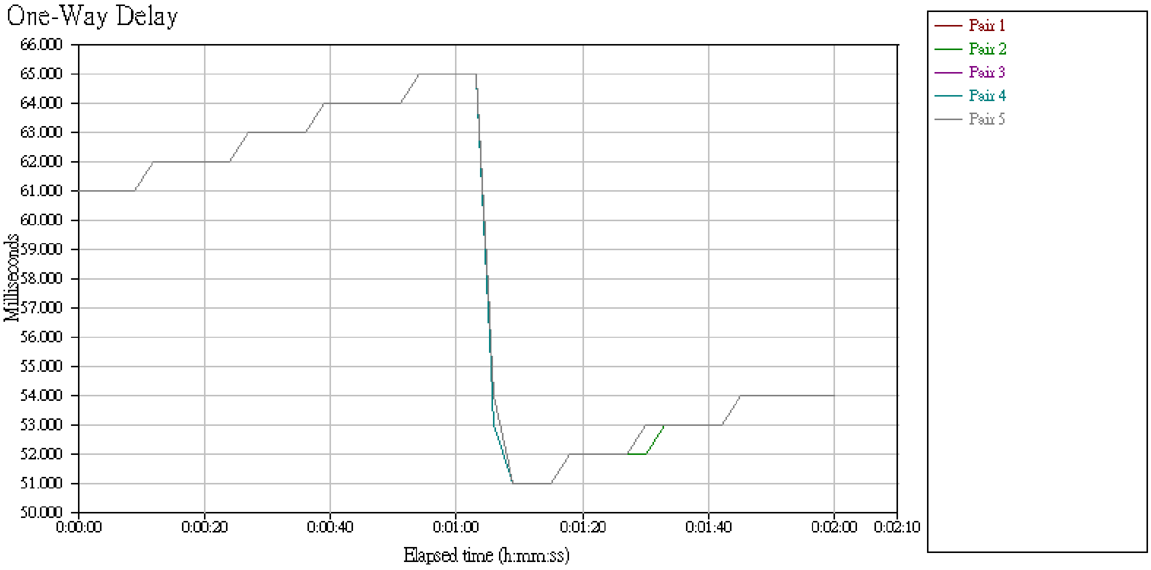


Figure 36 : One-Way Delay Result Graph for scenario-D

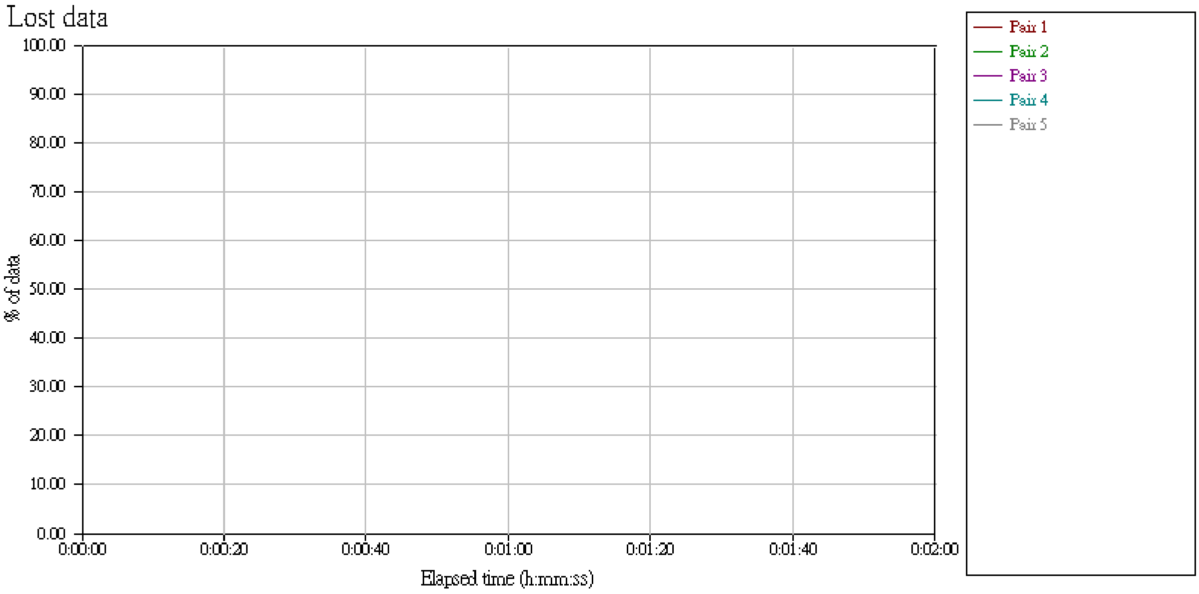


Figure 37 : Data Lost Result Graph for scenario-D

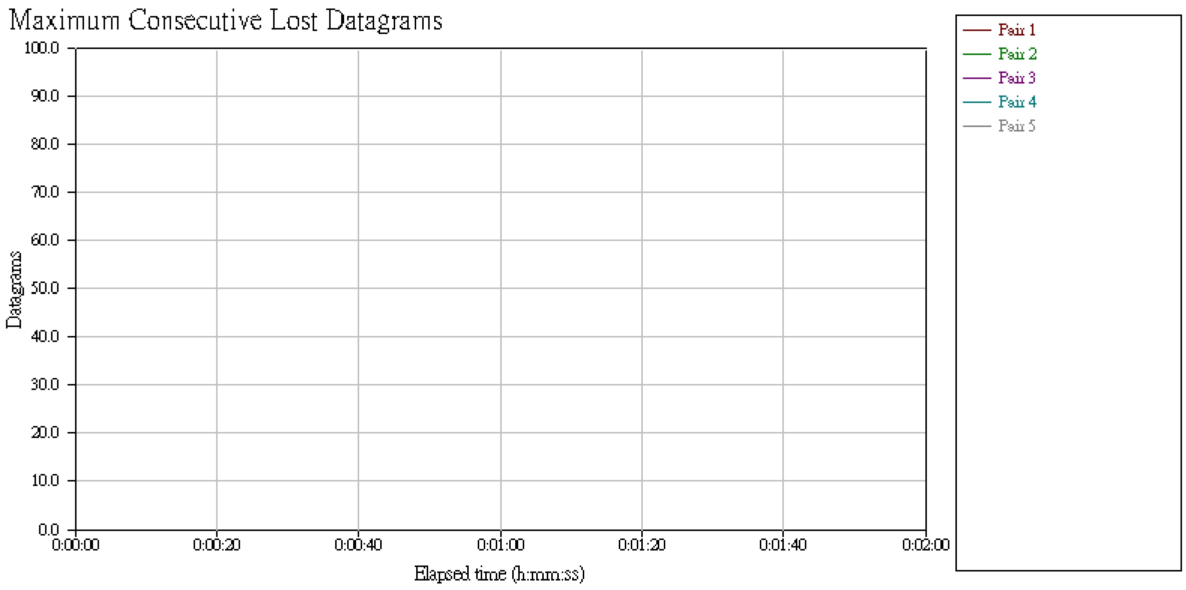


Figure 38 : Maximum Consecutive Lost Datagram Graph for scenario-D

The UDP datagram bandwidth was shaped in the range between 175 Kbps and 475 Kbps as shown in Figure 39 and 35% to 75% of the egress UDP datagram was also observed lost as shown in Figure 40.

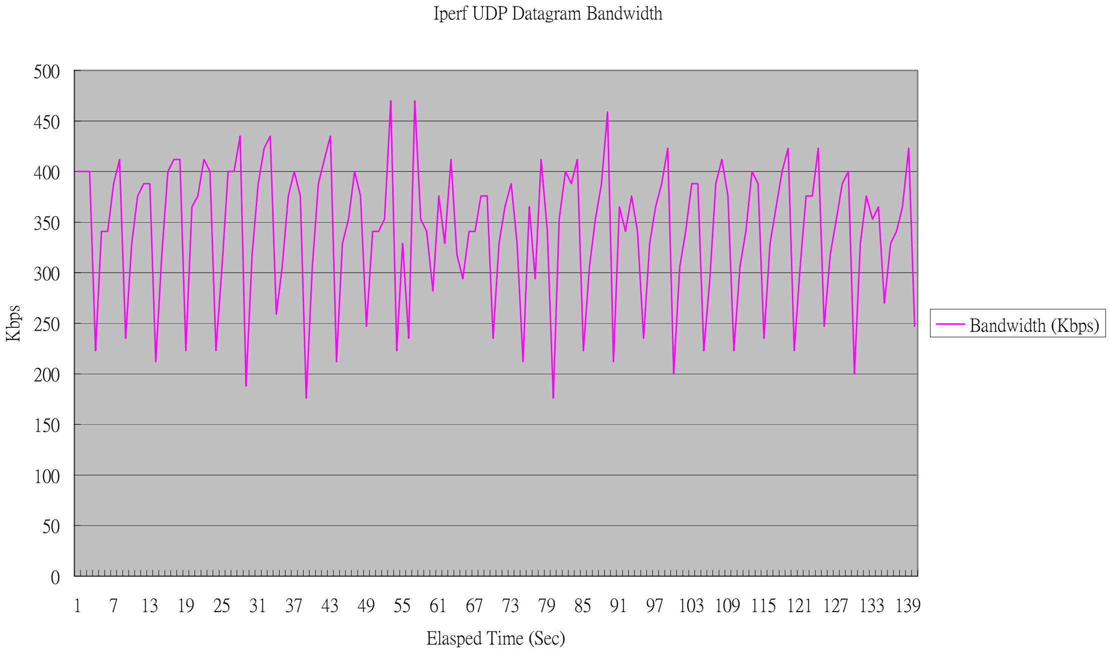


Figure 39 : Iperf UDP Datagram Bandwidth for scenario-D

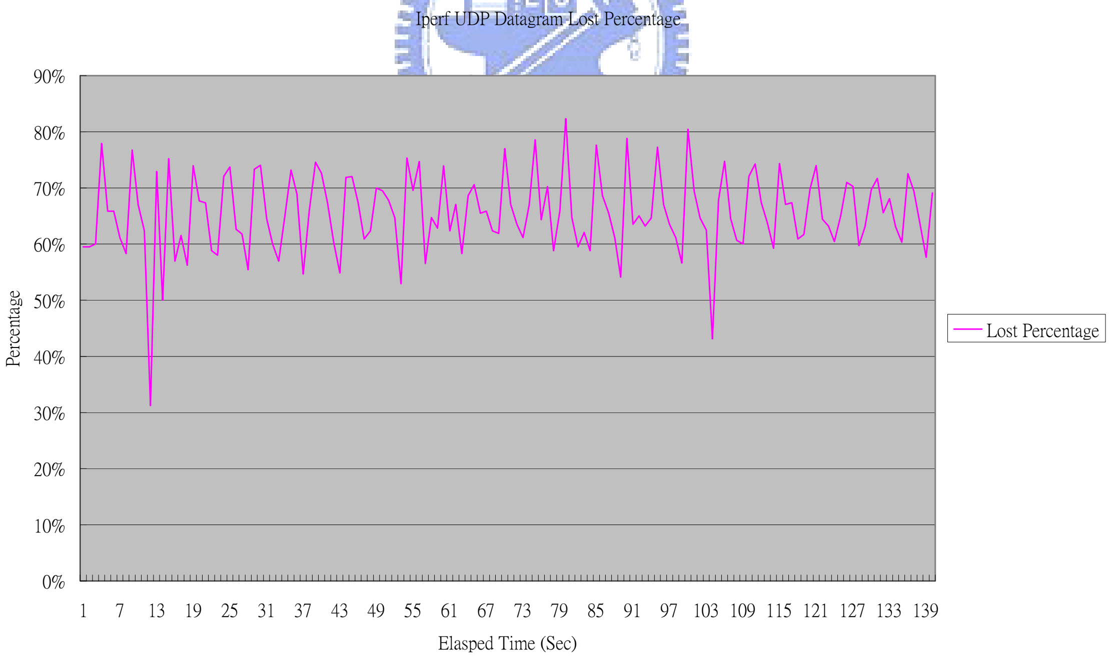


Figure 40 : Iperf UDP Datagram Lost Percentage for scenario-D

Scenario-E: Five RTP sessions along with single FTP session with SQPV enabled

Again, we'd like to see the improvement that SQPV can bring. The same test setup in scenario-C was established again but the SQPV was enabled in order to redo the test and measure the result again. The throughput number of each RTP session was pretty close to 64 Kbps as shown in Figure 41. The one-way delays were in the range between 51 ms and 56 ms as shown in Figure 42. The lost datagram and consecutive datagram losses were all zeros as shown in Figure 43 and Figure 44. It showed the great improvement compared the demonstrated result in scenario-C.

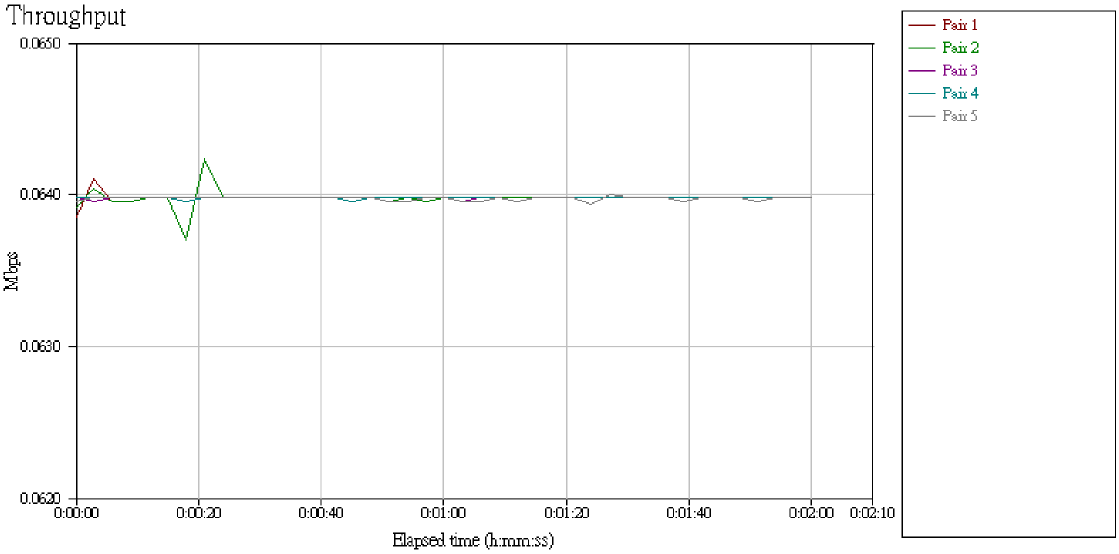


Figure 41 : Throughput Result Graph for scenario-E

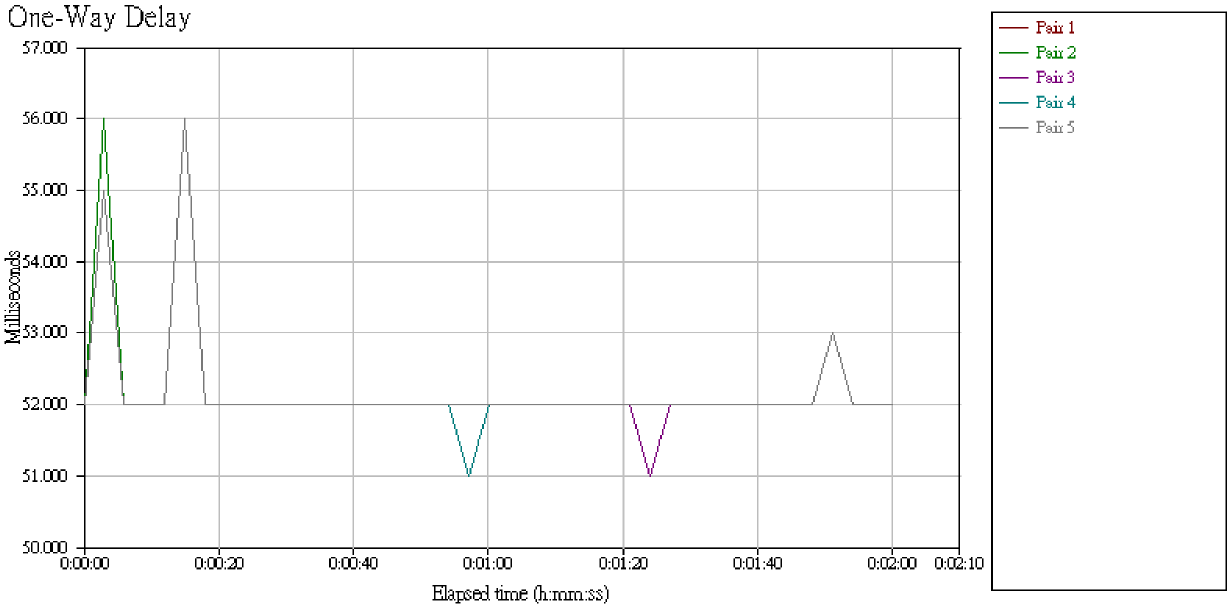


Figure 42 : One-Way Delay Result Graph for scenario-E

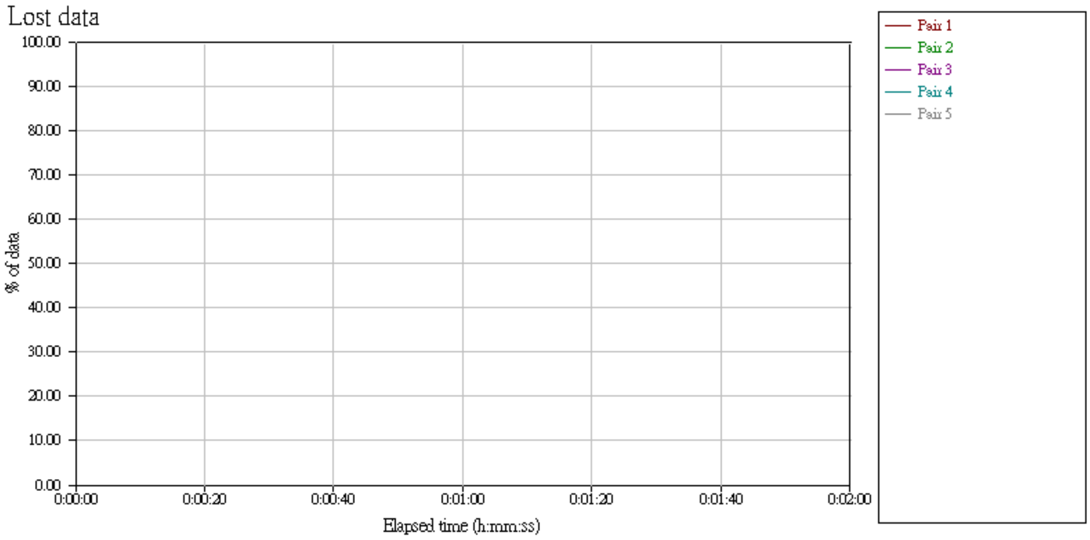


Figure 43 : Data Lost Result Graph for scenario-E

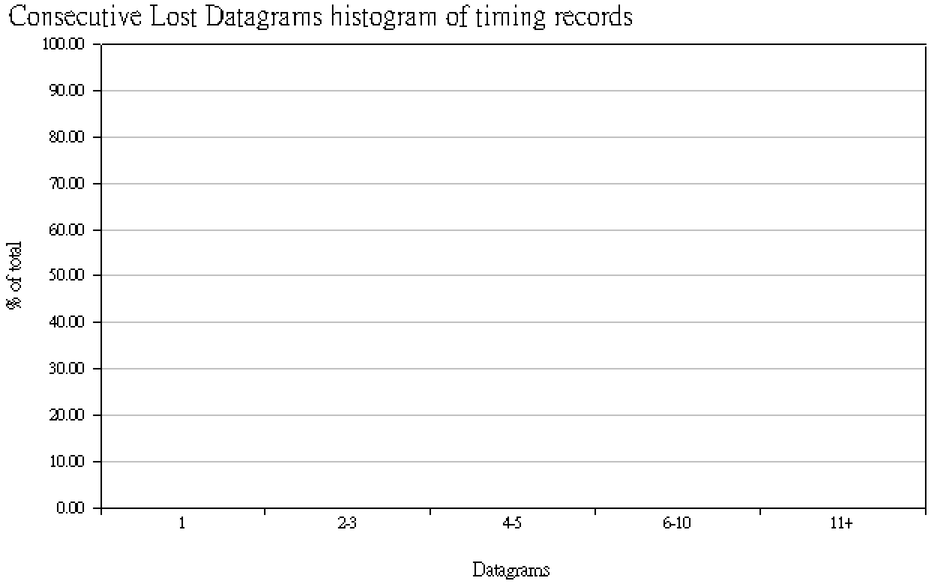


Figure 44 : Maximum Consecutive Lost Datagram Graph for scenario-E

4.3.3 Ingress Traffic Shaping – I-Police

Besides the improvement which had been shown as indicated in earlier section 3.3.2, we'd like to check if SQPV could also work it out in the ingress direction by having 3 more scenarios (from F to H) defined and executed.

Scenario-F: Five RTP sessions only

Five RTP sessions were created in the ingress direction from host E to host B and the target was to demonstrate the result if there was no other traffic competing with these 5 established RTP sessions. The data throughput for each session was 64Kbps as shown in Figure 45.

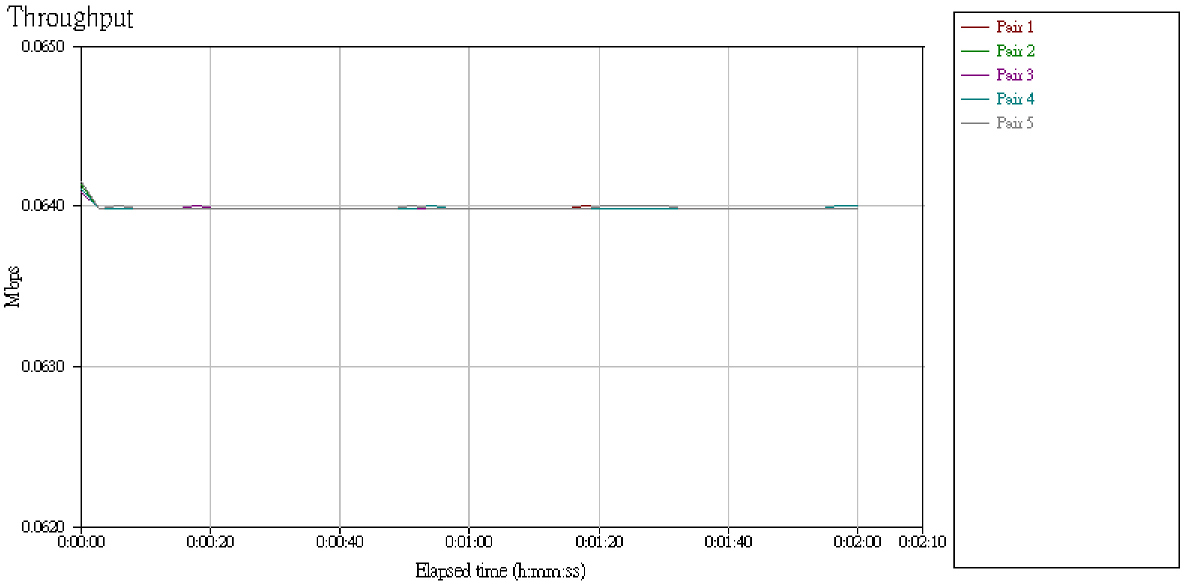


Figure 45 : Throughput Result Graph for scenario-F

The one-way delays were observed in the range between 38 ms and 50 ms as shown in Figure 46 and the lost datagram and the consecutively lost datagram shown in Figure 47 and Figure 48 are all zeros.

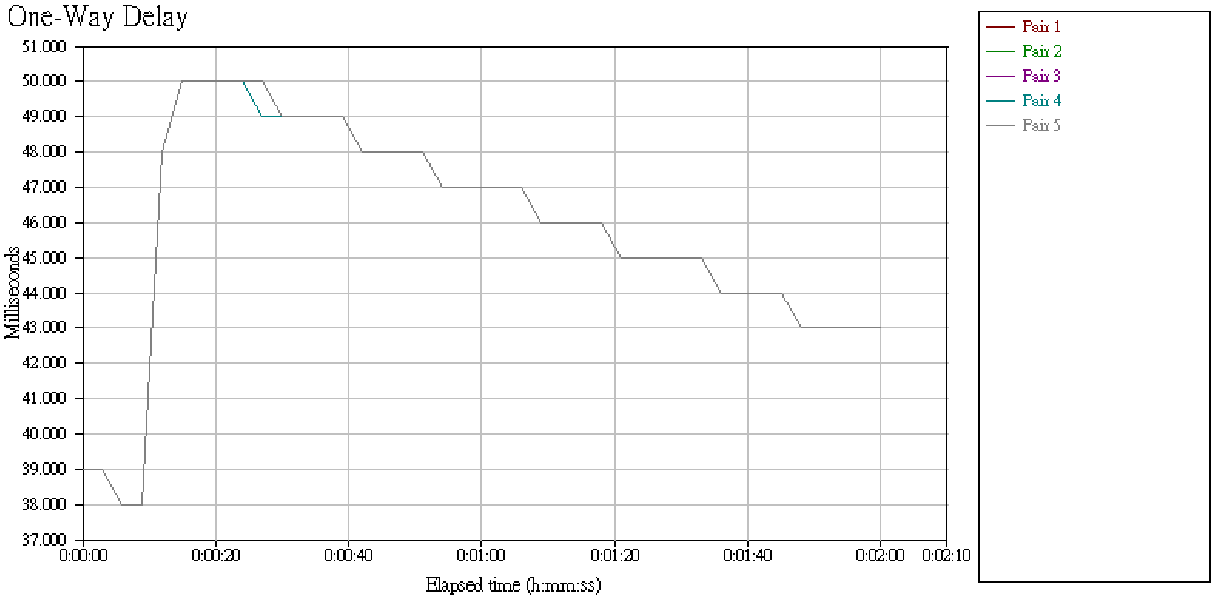


Figure 46 : One-Way Delay Result Graph for scenario-F

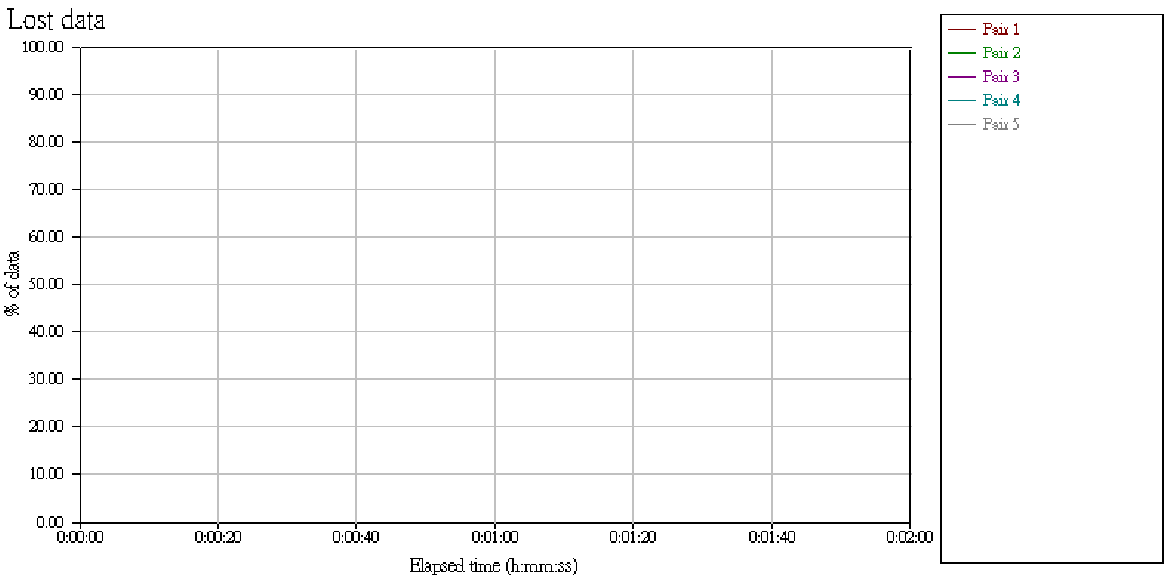


Figure 47 : Data Lost Result Graph for scenario-F

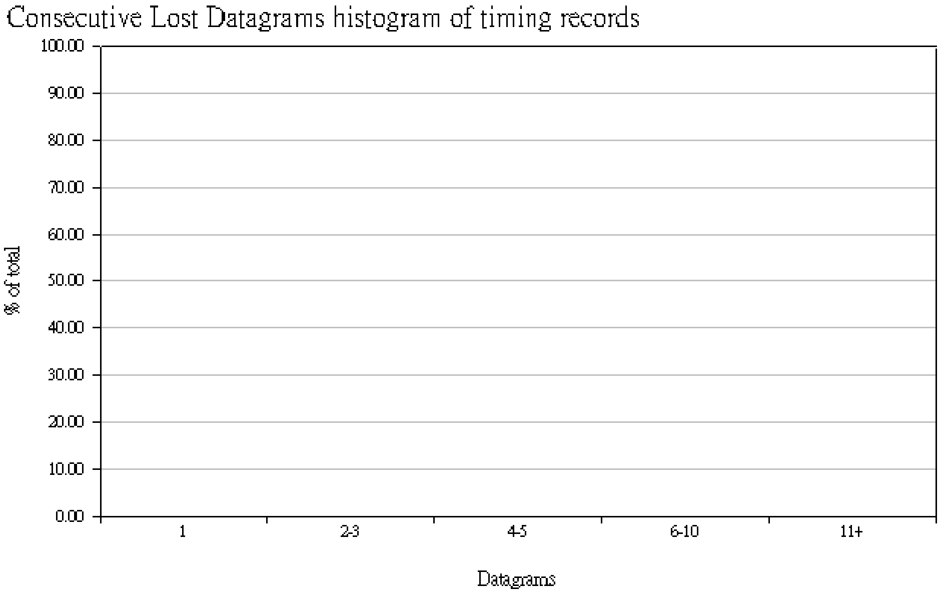


Figure 48 : Maximum Consecutive Lost Datagram Graph for scenario-F

Scenario-G: Five RTP sessions along with single TCP session without SQPV

The TCP datagram was injected into the network to compete against 5 established RTP sessions in the ingress direction. The Iperf client installed at host F kept sending the TCP datagram to the Iperf server installed at host A. The throughput varied in the range between 61 Kbps and 69 Kbps as shown in Figure 49 for each RTP session.

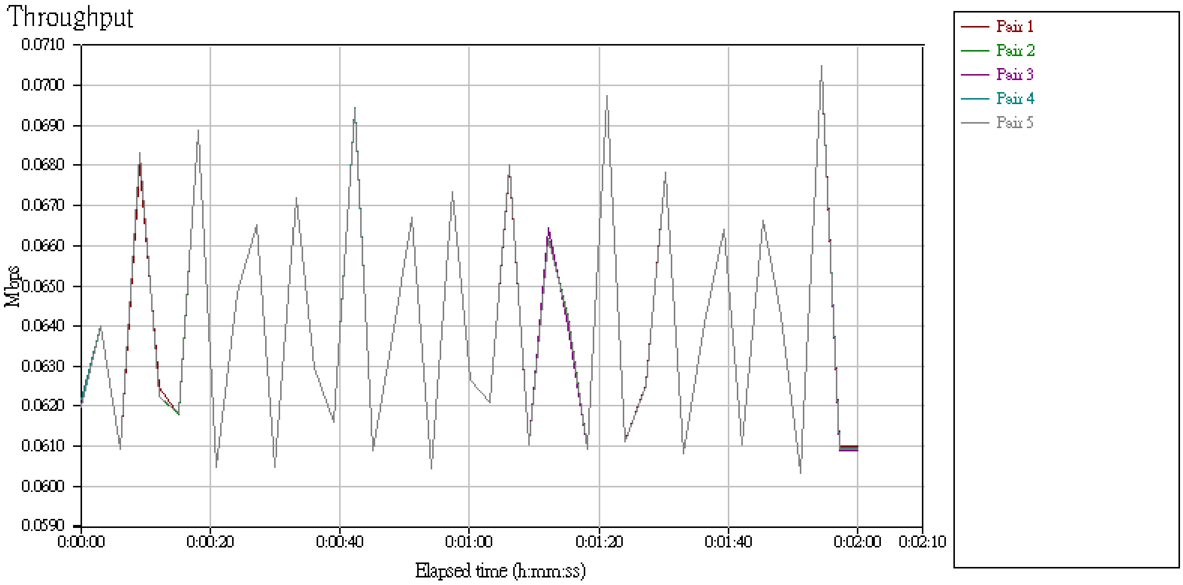


Figure 49 : Throughput Result Graph for scenario-G

The one-way delay was in the range between 389 ms and 422 ms as shown in Figure 50 The lost datagram and consecutive datagram losses shown in Figure 51 and Figure 52 are also zeros. The one-way end-to-end delays demonstrated the poor result while 5 RTP sessions were running together with TCP datagram simultaneously in the ingress direction.

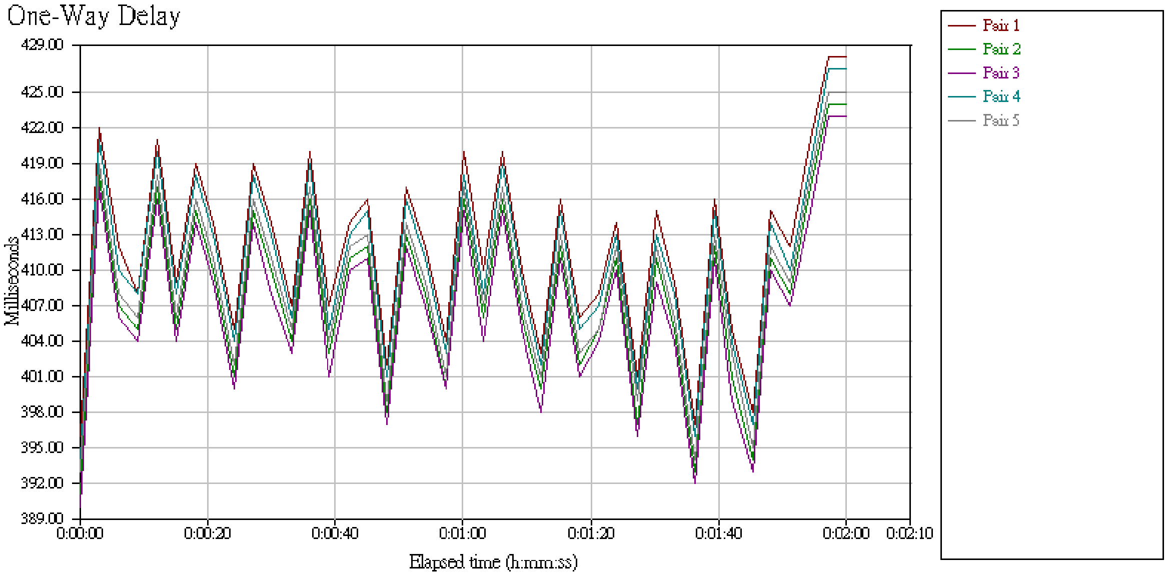


Figure 50 : One-Way Delay Result Graph for scenario-G

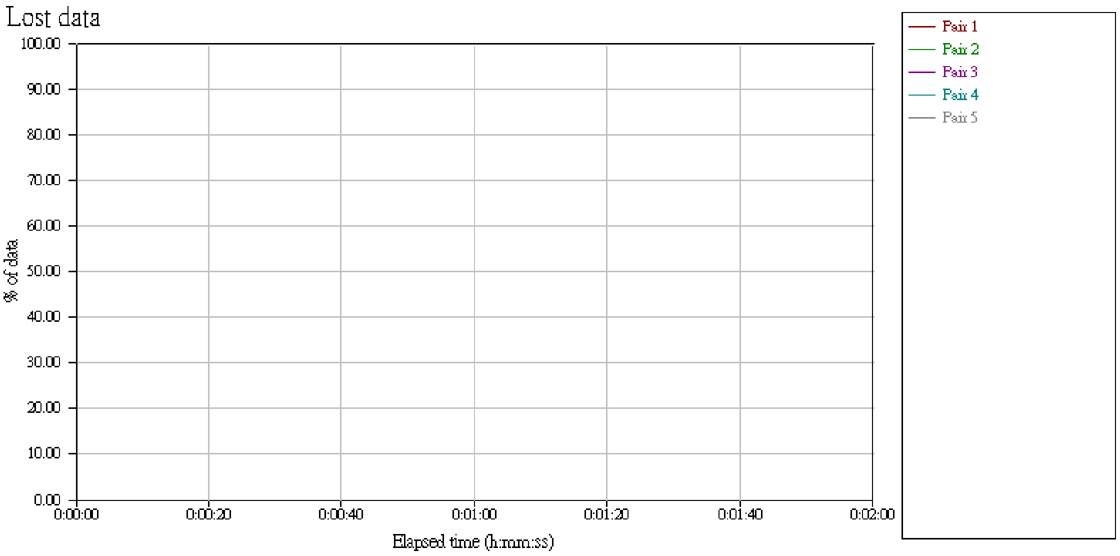


Figure 51 : Data Lost Result Graph for scenario-G

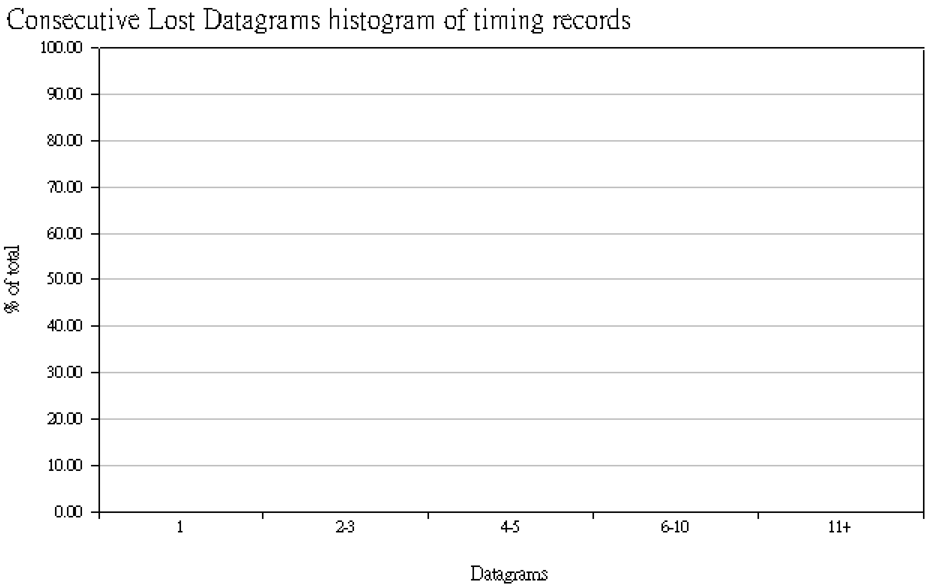


Figure 52 : Maximum Consecutive Lost Datagram Graph for scenario-G

Scenario-H: Five RTP sessions along with ingress TCP datagram with SQPV enabled

The SQPV was enabled in the scenario to see if any improvement can be happened. First of all, the throughput of each RTP session was in the range between 63.5 Kbps and 64.5 Kbps as shown in Figure 53. The one-way end-to-end delays are in the range between 42 ms and 63 ms as shown in Figure 54. The lost datagram and consecutive datagram losses shown in Figure 55 and Figure 56 are zeros. The bandwidth of the non-priority ingress TCP datagram was able to be shaped at most 650 Kbps as shown in Figure 57. Compared to the result shown in test scenario-F and scenario-G, the remarkable improvement was demonstrated.

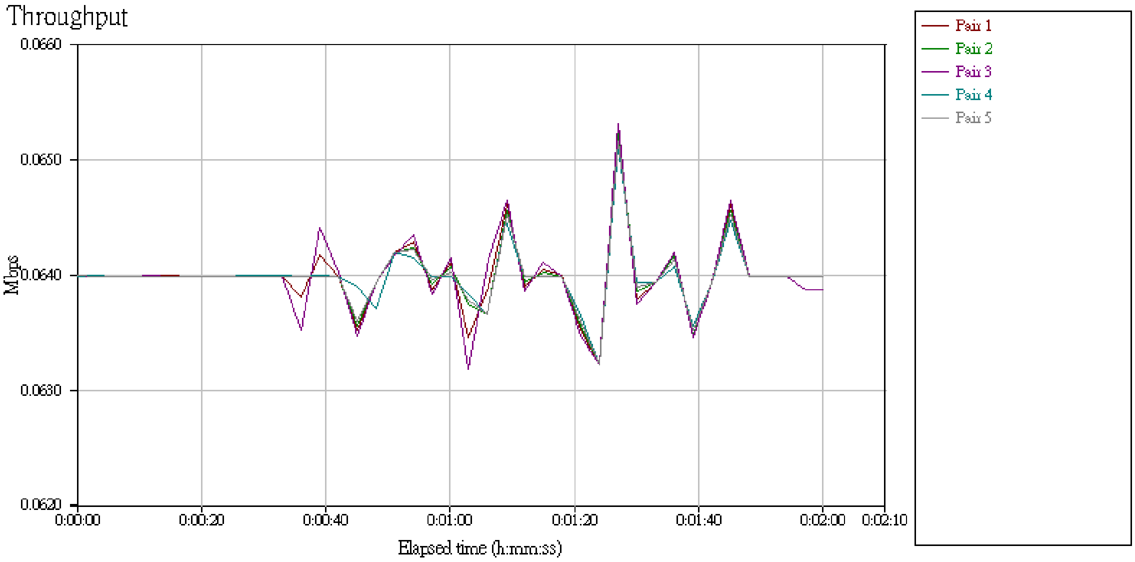


Figure 53 : Throughput Result Graph for scenario-H

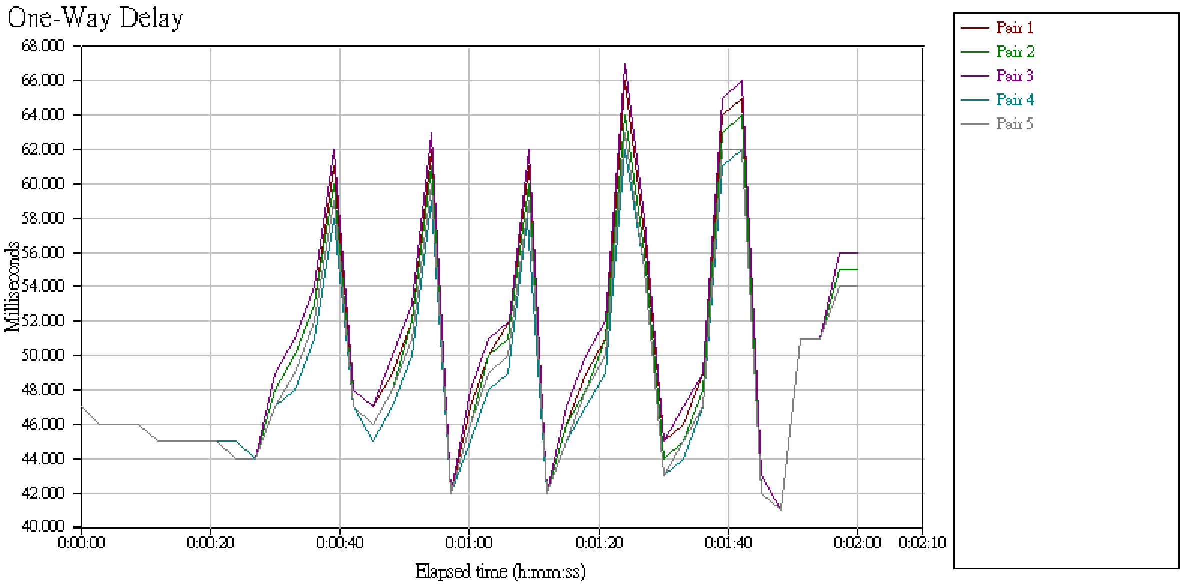


Figure 54 : One-Way Delay Result Graph for scenario-H

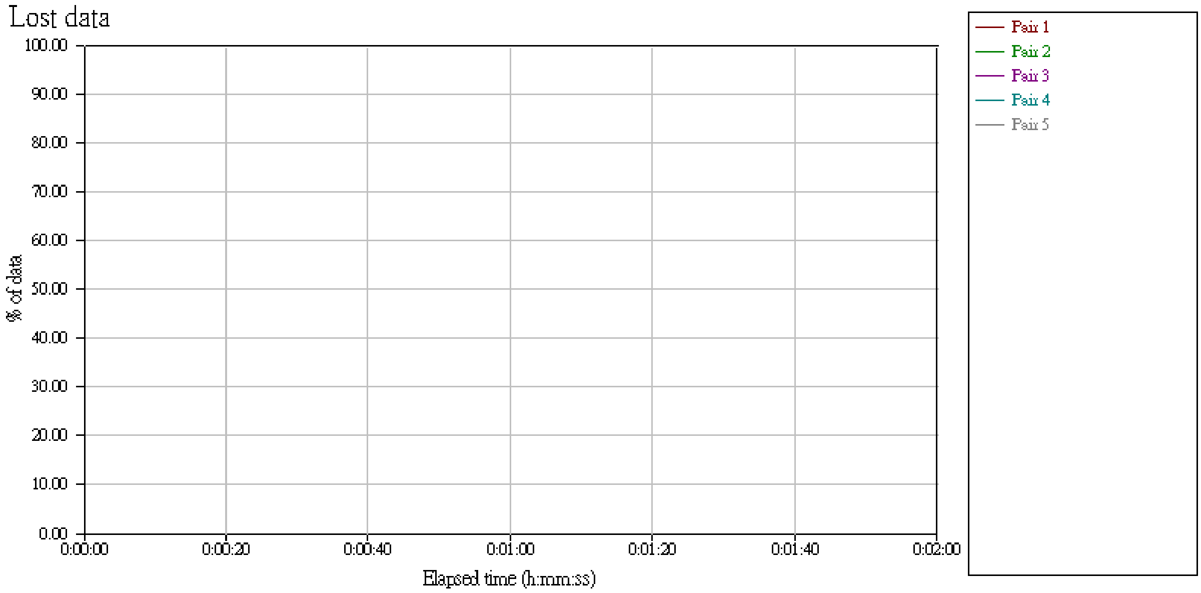


Figure 55 : Data Lost Result Graph for scenario-H

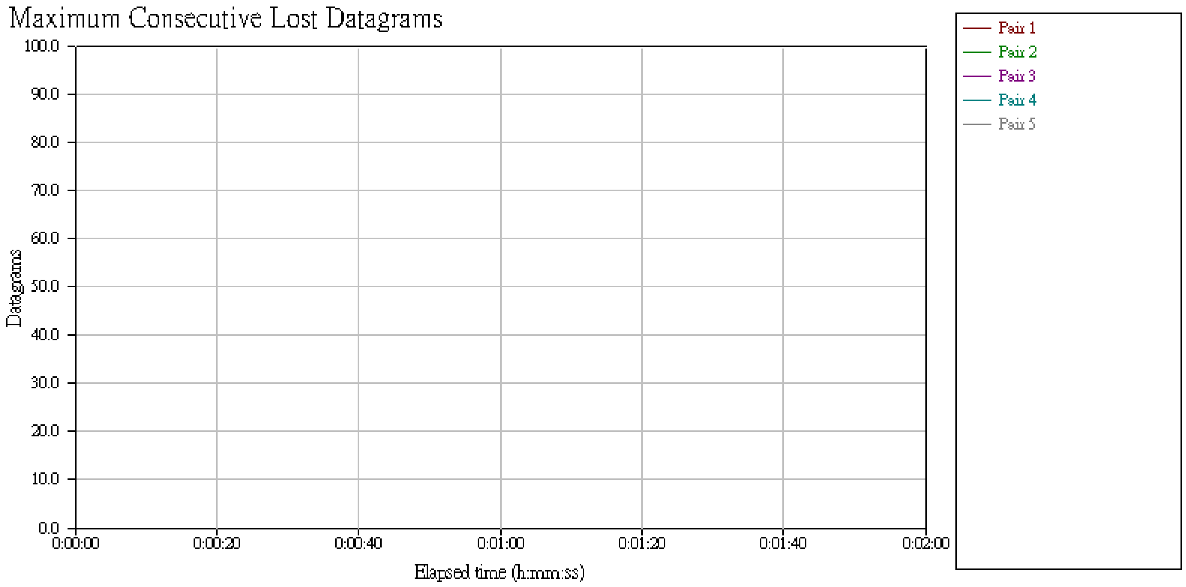


Figure 56 : Maximum Consecutive Lost Datagram Graph for scenario-H

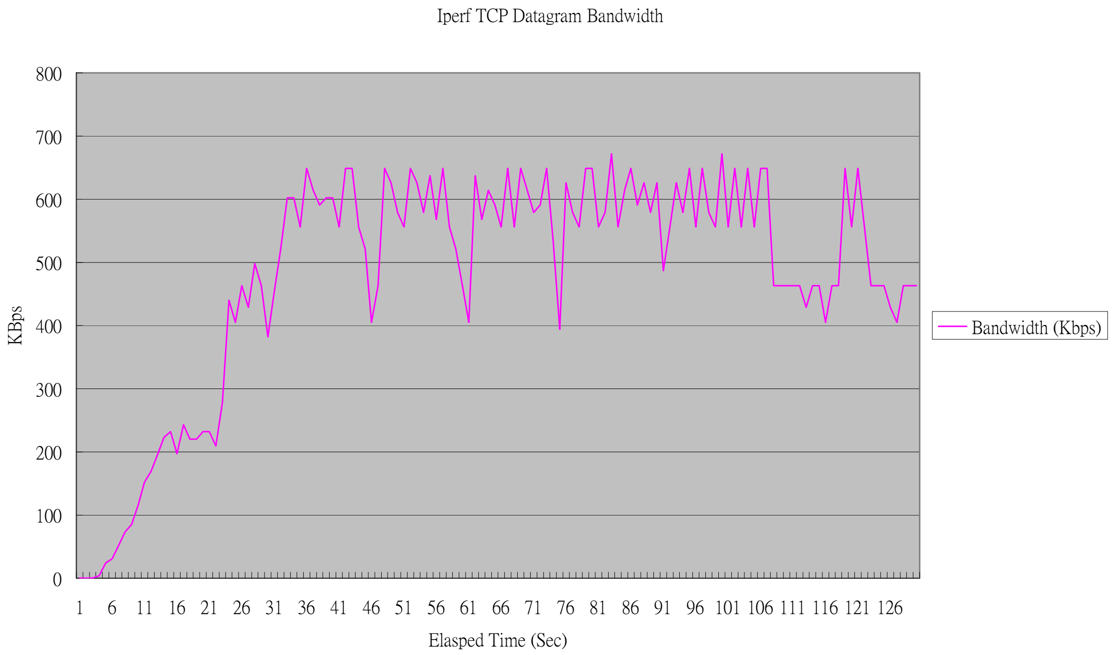


Figure 57 : Iperf TCP Datagram Bandwidth for scenario-H

5 Summary and Future Work

Quality of service is a prerequisite for real-time applications like voice over IP. The residential gateway plays as the first gate keeper at the residential endpoints to make sure the priority datagram can be given precedence in transmission and receiving over the other non-priority datagram. Usually the limited upstream and downstream bandwidth are given by most of the present internet service providers and the common incremental requirement of the network bandwidth for the residential users, it’s getting more and more critical to ensure the quality of service for the priority datagram like voice over IP.

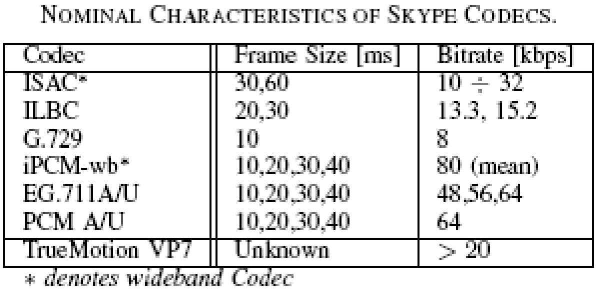
Three algorithms (E-Meter, RPDTB and I-Police) were proposed and implemented to address three key portions by measuring the egress available bandwidth, shaping the egress traffic and lowering the ingress TCP traffic. The objective of the implementation of SQPV is to prove and show the output result of E-Meter, RPDTB and I-Police in the test setup with the real network deployment emulated. The results demonstrated the egress bandwidth measurement reached a good level of accuracy, the outgoing non-priority traffic was well shaped and the incoming TCP traffic was also congestion avoided. The good service of quality for 5 SIP/RTP sessions were able to be assured concurrently with the other non-priority TCP and UDP sessions.

The proposed algorithms for E-Meter, RPDTB and I-Police can be applied and reused for the enhanced quality of service provision system for various kinds of real-time streaming applications. SIP signaling protocol along with the 64 Kbps G.711 CODEC were chosen and implemented in SQPV to simply demonstrate the result of E-Meter, RPDTB and I-Police. As long as some configuration on the bandwidth requirement of the additional real-time streaming applications and modification on the Traffic Parser to support and detect the additional required signaling protocol of the real-time streaming applications, the different quality of service provision system can be provided.

Besides SIP, Skype [21] is the most amazing example of this new phenomenon. It recently reached over 170 millions of users and accounts for more 4.4% of total VoIP traffic [18]. It’s worth making a proposal to support Skype here as an example.

Skype relies on a P2P infrastructure to exchange signaling information in a distributed fashion which can be making the system highly scalable and robust [17]. Except for the user’s authentication which is performed under a classical client and server architecture by means of public key mechanisms, all further signaling is performed in P2P network. So the Skype user’s information are entirely decentralized and distributed among nodes.

This allows the service to scale very easily to large sizes and avoid a costly centralized infrastructure. However, Skype uses a proprietary solution which is difficult to reverse engineer due to extensive use of both cryptography and obfuscation techniques [19]. It makes the traffic parser difficult to interpret the Skype signaling protocols and figure out how to add or remove the Skype session to make the run-time bandwidth allocation and admission control possible. Skype is able to select different CODEC according to the unknown algorithm. According to the nominal characteristics of Skype CODECS showed in Figure 58 [20], the bit rate of the different CODECS is at least 8 Kbps (G.729) and 80 Kbps (iPCM-wb) at most.



NOMINAL CHARACTERISTICS OF SKYPE CODECS.

Codec	Frame Size [ms]	Bitrate [kbps]
ISAC*	30,60	10 ÷ 32
ILBC	20,30	13.3, 15.2
G.729	10	8
iPCM-wb*	10,20,30,40	80 (mean)
EG.711A/U	10,20,30,40	48,56,64
PCM A/U	10,20,30,40	64
TrueMotion VP7	Unknown	> 20

∗ *denotes wideband Codec*

Figure 58 : Nominal Characteristics of Skype CODECS

The QoS provisioning mechanisms (E-Meter, RPDTB and I-Police) are applicable to Skype calls as well since the egress bandwidth measurement, egress traffic shaping and ingress congestion avoidance are all required regardless of different kinds of real-time streaming applications. Limitation is the admission control and bandwidth allocations at running time are not possible due to lack of transparency of Skype signaling protocols. The alternative approach is to have the preconfigured amount of bandwidth for priority traffic (like Skype calls) and modify the congestion detection scheme of I-Police accordingly.

6 Bibliography

[1] L. Peterson, B. Davie, Computer Networks, A Systems Approach, Edition 3, Morgan Kaufmann Publishers, San Francisco, 2003

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services, RFC 2475, December 1998

[3] M. Handley, V. Jacobson, C. Perkins, SDP: Session Description Protocol, RFC 4566, July 2006

[4] A. Johnston, SIP: Understanding the Session Initiation Protocol, 2nd edition, Artech House, 2004

[5] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RFC 3550, RTP: A Transport Protocol for Real-Time Applications, July 2003

[6] Official PESQ website, http://www.pesq.org/, 2001, 2002, 2003, 2004, 2005, 2006, 2007 OPTICOM GmbH, Erlangen, Germany

[7] A. Chadda, Quality Of Service Testing Methodology, Dec, 2004

[8] H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, RFC 3261 - SIP: Session Initiation Protocol, June 2002

[9] National Institute of Standards and Technology Group, NIST Net Home page, http://snad.ncsl.nist.gov/nistnet/, Jan, 1998

[10] SIPp, http://sipp.sourceforge.net/index.html, 2004

[11] The Board of Trustees, University of Illinois, http://www.dast.nlanr.net/projects/Iperf/, 1999-2006

[12] P. Brodsky, TeleGeography Research, http://www.apritel.org/fotos/editor2/Paul_Brodsky.pdf

[13] W.S. Hwang and P.C. Tseng, A QoS-aware Residential Gateway with Bandwidth Management, IEEE Transactions on Consumer Electronics, Vol. 51, No. 3, page 840-848, August, 2005, http://credit.csie.ncku.edu.tw/opensource/93report/CMMI_RR/38.pdf

[14] S. Floyd, Lawrence Berkeley Laboratory, Note on CBQ and guaranteed service, July, 1995

[15] D. Bansal, J. Bao, and W. Lee, Motorola Labs, QoS-Enabled Residential Gateway Architecture, IEEE Communications Magazine, Vol. 41, issue 4, page 83-89, April 2003

[16] R. Yavatkar, Intel, D. Hoffman, Teledesic, Y. Bernet, Microsoft, F. Baker, Cisco, M. Speer, Sun Microsystems, RFC 2814 - SBM (Subnet Bandwidth Manager): A Protocol for RSVP-based Admission Control over IEEE 802-style networks, May 2000

[17] D. Rossi, M. Mellia, M. Meo, ENST Telecom Paris, France, dario.rossi@enst.fr,

Politecnico di Torino, Italy, Following Skype Signaling Footsteps, page 248-253, IT-NEWS, 2008 IEEE

[18] International carriers’ traffic grows despite Skype popularity, Tele-Geography Report and Database, available on line http://www.telegeography.com/, Dec. 2006

[19] P. Biondi, F. Desclaux, Silver Needle in the Skype, Black Hat Europe 06, Amsterdam, the Netherlands, Mar 2006

[20] D. Bonfiglio, M. Mellia, M. Meo, Nicol`o Ritacca, Politecnico di Torino – Dipartimento di Elettronica, D. Rossi, ENST ParisTech – INFRES Department, Tracking Down Skype Traffic, page 843-851, 2008 IEEE

[21] Skype web site, http://www.skype.com