A novel multimedia adaptation architecture and congestion control mechanism designed for real-time interactive applications

Chaudhery, Touseef Javed

— PhD Thesis —

A novel Multimedia Adaptation Architecture and Congestion Control Mechanism designed for Real-time Interactive Applications

Author: Touseef Javed Chaudhery

Email: t.j.chaudhery@gmail.com

Date: Dec 2010

THESIS SUBMITTED TO THE UNIVERSITY OF LONDON FOR THE DEGREE OF 'DOCTOR OF PHILOSOPHY'

Networks Research Group,
School of Electronic Engineering and Computer Science,
Queen Mary University of London,
Mile End Road, London, E1 4NS, England.

Abstract

The increasing use of interactive multimedia applications over the Internet has created a problem of congestion. This is because a majority of these applications do not respond to congestion indicators. This leads to resource starvation for responsive flows, and ultimately excessive delay and losses for all flows therefore loss of quality. This results in unfair sharing of network resources and increasing the risk of network ‘congestion collapse’.

Current Congestion Control Mechanisms such as ‘TCP-Friendly Rate Control’ (TFRC) have been able to achieve ‘fair-share’ of network resource when competing with responsive flows such as TCP, but TFRC’s method of congestion response (i.e. to reduce Packet Rate) is not ideally matched for interactive multimedia applications which maintain a fixed Frame Rate. This mismatch of the two rates (Packet Rate and Frame Rate) leads to buffering of frames at the Sender Buffer resulting in delay and loss, and an unacceptable reduction of quality or complete loss of service for the end-user.

To address this issue, this thesis proposes a novel Congestion Control Mechanism which is referred to as ‘TCP-friendly rate control – Fine Grain Scalable’ (TFGS) for interactive multimedia applications.

This new approach allows multimedia frames (data) to be sent as soon as they are generated, so that the multimedia frames can reach the destination as quickly as possible, in order to provide an isochronous interactive service. This is done by maintaining the Packet Rate of the Congestion Control Mechanism (CCM) at a level equivalent to the Frame Rate of the Multimedia Encoder.

The response to congestion is to truncate the Packet Size, hence reducing the overall bitrate of the multimedia stream. This functionality of the Congestion Control Mechanism is referred to as Packet Size Truncation (PST), and takes advantage of adaptive multimedia encoding, such as Fine Grain Scalable (FGS), where the multimedia frame is encoded in order of significance, Most to Least Significant Bits. The Multimedia Adaptation Manager (MAM) truncates the multimedia frame to the size indicated by the Packet Size Truncation function of the CCM, accurately mapping user demand to available network resource. Additionally Fine Grain Scalable encoding can offer scalability at byte level granularity, providing a true match to available network resources.

This approach has the benefits of achieving a 'fair-share' of network resource when competing with responsive flows (as similar to TFRC CCM), but it also provides an isochronous service which is of crucial benefit to real-time interactive services. Furthermore, results illustrate that an increased number of interactive multimedia flows (such as voice) can be carried over congested networks whilst maintaining a quality level equivalent to that of a standard landline telephone. This is because the loss and delay arising from the buffering of frames at the Sender Buffer is completely removed. Packets sent maintain a fixed inter-packet-gap-spacing (IPGS). This results in a majority of packets arriving at the receiving end at tight time intervals. Hence, this avoids the need of using large Playout (de-jitter) Buffer sizes and adaptive Playout Buffer configurations. As a result this reduces delay, improves interactivity and Quality of Experience (QoE) of the multimedia application.

Acknowledgment

I would like to thank everyone who supported me morally and those who I enjoyed having long discussions. I will list them alphabetically so I don't hurt any feelings ☺. If I have forgotten any one please do forgive me.

Ali Raza, Ali Hasan, Maheen Hasib, M. Hashim. Moneeza Saeed Javed, Nishat Raja, Omar El Aziz (late), Parveen Akhtar, Samana Syed Naqvi, Sardar Kashif Ashraf Khan, Shahda Bashir, Sheraz Malik, Syed Rafi Ahmed, Tony (Qiang Yang), Vindya Amaradasa

I pray for the success of each and every one mentioned above. Most importantly I want to thank the one and only God, Allah (swt) for giving me the strength to complete what I have completed, and I want to pay my sallam to the Prophet Muhammed (saw).

Regards,

Dr. Touseef J. C.

Table of Contents

List of Figures

List of Tables

Abbreviations of Text

AIMD	Additive Increase Multiplicative Decrease
AQM	Active Queue Management
bps	bits per second
BR	Bitrate
CCM	Congestion Control Mechanism
CG	Coarse-Grain
FGS	Fine Grain Scalable
FI	Frame Interval
FR	Frame Rate
FS	Frame Size
FTP	File Transport Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPGS	Inter-Packet-Gap-Spacing
LER	Loss Event Rate
LSB	Least Significant Bytes
MAA	Multimedia Adaptation Architecture
MAM	Multimedia Adaptation Manager
MSB	Most Significant Bytes
MTU	Maximum Transmission Unit
N	Network
ns2	Network Simulator, version 2
PB	Playout Buffer (also referred to as the RB Buffer or de-jitter Buffer)
PDF	Probability Distribution Function
PR	Packet Rate
PRS	Packet Rate Scaling
PS	Packet Size

PST	Packet Size Truncation
Z	packetization
QoE	Quality of Experience
QoS	Quality of Service
RB	Receiver Buffer (also referred to as the PB Buffer or de-jitter Buffer)
RED	Random Early Discard
RTP	Real-time Transport Protocol
RTT	Round Trip Time
SB	Sender Buffer
SP	Start-up Phase
SR	Sending Rate
TCP	Transmission Control Protocol
TFGS	TCP friendly rate control - Fine Grain Scalable
TFRC	TCP Friendly Rate Control
TP	Throughput
TR	Terminate
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
wrt	with respect to

Abbreviations of Equation Variables

L_T	Total Loss
$L_{T(R)}$	Total Loss Ratio
S_L	Sender Buffer Loss
N_L	Network Loss
R_L	Receiver/Playout Buffer Loss
D_T	Total Delay
E_D	Encoding Delay
Z_D	Packetization Delay
S_D	Sender Buffer Delay
G_D	Propagation Delay
Q_D	Queuing Delay
t_D	Service Time Delay
t_{RTT}	Round Trip Time
R_D	Receiver/Playout Buffer Delay
$Z_{D(R)}$	De-Packetization Delay
$E_{D(d)}$	Decoding Delay
F_I	Frame Interval
F_R	Frame Rate
F_S	Frame Size
F_N	Number of Frames
P_S	Packet Size
P_R	Packet Rate
$P_{S(max)}$	Maximum Packet Size
$P_{R(max)}$	Maximum Packet Rate
$P_{S(min)}$	Minimum Packet Size

$P_{R(fix)}$	Fixed Packet Rate
$P_{IPGS(S)}$	Inter-Packet-Gap-Spacing at Sender
$P_{IPGS(R)}$	Inter-Packet-Gap-Spacing at Receiver
P_{NPG}	Number of Packets Generated
I_L	Loss Impairment
I_D	Delay Impairment
A	Advantage Factor
n_{frames}	Number of Frames
$n_{packets}$	Number of Packets
n_{flows}	Number of Flows
Q_S	Queue Size
Q_{max}	Maximum Queue Threshold
Q_{min}	Minimum Queue Threshold
C	Link Capacity
T_{TCP}	TCP Throughput
T_{EB}	Encoder Bitrate
T_{FS}	Fair Share Sending Rate
T_{ETR}	Effective Transmission Bitrate
$T_{ETR(min)}$	Minimum Effective Transmission Bitrate
T_{NSR}	Normalised Sending Rate

1. Introduction

1.1. Motivation and Contribution

There is an increasing demand for carrying interactive real-time applications with Quality of Service (QoS) requirements over a best-effort Internet. This is shifting the balance between aggregate traffic volumes that react to network congestion and those multimedia applications that do not react to network congestion. The current Internet was originally designed for data communication with the best-effort architecture in mind. Its underlying assumption is that all types of traffic need to have equal access to network resources and hence to share them fairly [1]. Unlike the data traffic transported over TCP (which is considered adaptive as it is able to react and adapt to network congestion) multimedia traffic is inelastic. Therefore, in a converged network environment with the presence of network congestion, the adaptive traffic will manage to reduce the data transmission rate upon congestion indication, whilst the inelastic multimedia applications will continue to transmit data regardless of network congestion. As a result, the multimedia applications will occupy more resources than a fair share would entitle them to. This leads to resource starvation for adaptive flows and ultimately excessive losses and delays for all data traffic. The inelastic flows do not respond to network congestion indicators because they use UDP as their transport protocol where no feedback of network congestion is provided. Multimedia applications avoid using TCP as their underlying transport protocol because during congestion TCP halves its window size, which typically results in halving of the sending rate. This results in drastic degradation in perceptual quality. In addition, due to the nature of the TCP protocol, those lost packets which are re-transmitted will take at least one Round Trip time (RTT) to arrive at their destination. This delay of one RTT violates the time-critical nature of multimedia applications: although the packets are delivered to the end destination, they are of no use to the application as they have arrived 'too-late' to be played out in sequence.

The network unfairness introduced by the unresponsive behaviour of inelastic multimedia traffic can result in network 'congestion collapse'. The network will be kept busy transmitting packets. However, they will simply be discarded before reaching their final destinations.

Based on the above observations, the Internet Engineering Task Force (IETF) suggested that all applications carried by the converged Internet infrastructure should integrate end-to-end Congestion Control Mechanism (CCM) in order to achieve long-term fairness of network resources utilisation and reduce the risk of 'congestion collapse' [2, 3]. Congestion Control Mechanisms aim to provide several benefits to both the underlying communication network and multimedia applications. With such an approach, a) network bandwidth is divided between flows in a fair manner, b) multi-service traffic can well coexist in the network without the need for traffic segregation and per-flow scheduling to guarantee their communication performance, c) Furthermore, such end-to-end Congestion Control Mechanisms can be deployed on a large scale and without requiring the Internet paradigm to be changed. Additionally when using adaptive multimedia applications their demands can be easily mapped to available network resources.

The main requirements for transmitting multimedia applications over the Internet are as follows:

(*1*) Interactive Multimedia Applications are delay sensitive, semi-reliable (i.e. able to tolerate packet losses) and rate-based. Thus, they require isochronous processing at the sender and receiver ends, in order to achieve acceptable Quality of Service (QoS) on the end-to-end path. In order to achieve this, multimedia applications would ideally prefer to send their data frames as soon as they are generated to minimise delay, loss and jitter. On the other hand multimedia applications can tolerate available network bitrate variations by reducing the quality of their frames, thereby reducing the overall bitrate sent.

In the literature [4, 5], developments of Coarse Grain scalable encoding have been shown to provide a number of bitrate options by providing some additional enhancement layers. However, because of the quantized steps in the bitrates offered by Coarse Grain encoders, a true match to network supply cannot be met, as shown in Figure 1.1. Recent advances in Fine Grain Scalable encoding offer bitrate scalability at byte-level granularity, as shown in Figure 1.2. This can significantly improve the quality of the stream and match available network resources to byte-level precision.

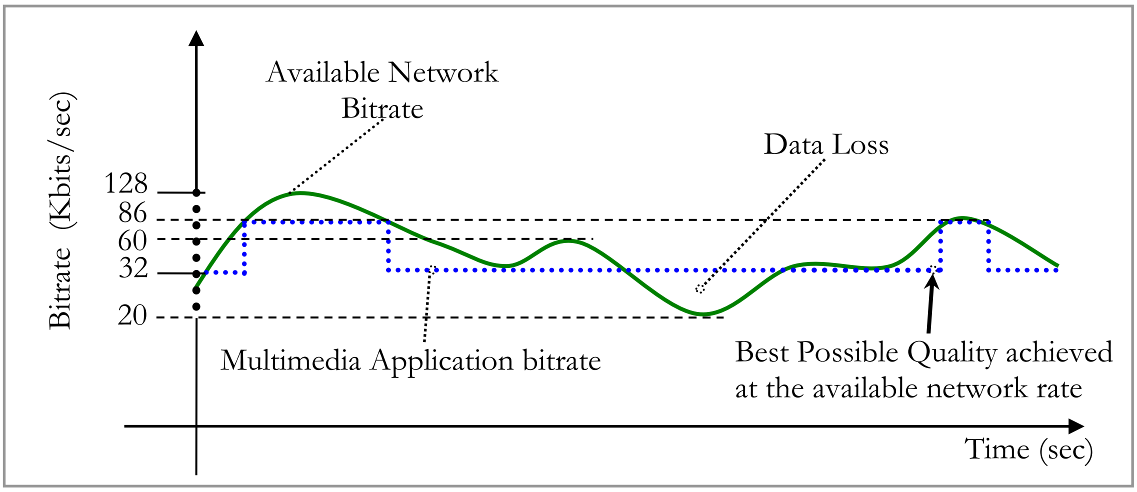


Figure 1.1, 'Varying Transmission rates' – Coarse-Grain Encoding

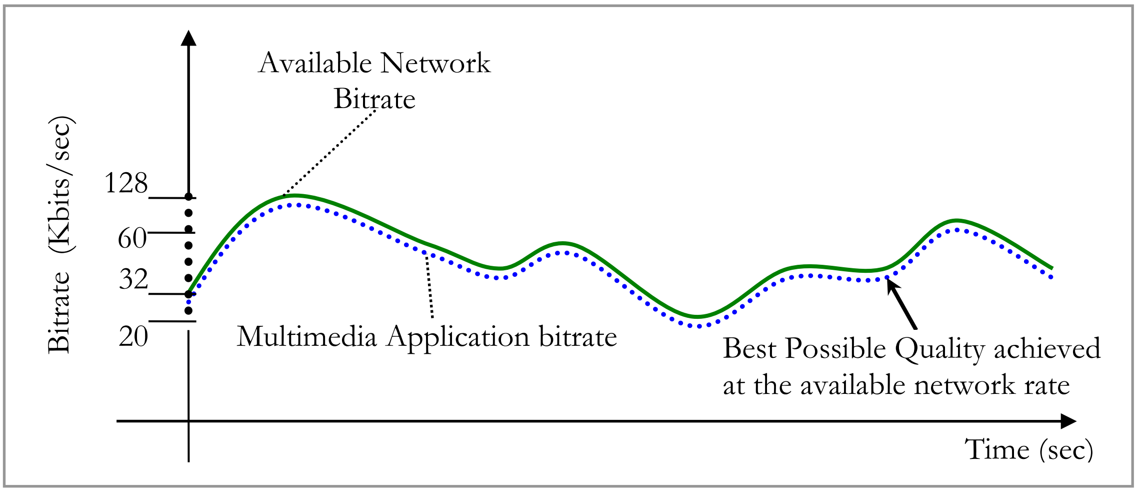


Figure 1.2, 'Adapting to Varying Transmission rates'

(*2*) Network Requirements: the Internet is a shared infrastructure. This means that applications are expected to utilise network resources fairly, i.e. by reacting to network congestion properly and promptly [3]. End-to-end Congestion Control Mechanism (CCM) achieves this. The CCM is able to estimate the available network resources (i.e. bandwidth) based on the state of the network. The available bandwidth could vary in an unpredictable and potentially wide fashion, therefore applications are required to adapt accordingly.

To reconcile the requirements of multimedia applications and network supply seamlessly, multimedia applications must be 'quality adaptive'. That is, the multimedia application should adjust the quality of the delivered stream so that the application's bitrate can match the available network bitrate as indicated by the Congestion Control Mechanism (CCM). This is illustrated in Figure 1.2. Furthermore, the CCM should adapt smoothly to varying network conditions, avoiding such abrupt variations as would be caused by, for example, halving the sending rate in response to single packet loss, as seen in TCP. Frequent changes in quality adaptation can be annoying to the end user, therefore smoother changes in quality will be more promising in terms of Quality of Experience (QoE) [6, 7].

Thus the main issue is how to provide end-to-end congestion control for multimedia applications. The updates to the Real-time Transport Protocol (RTP) specifications suggest that multimedia applications should adapt to congestion by using a congestion control mechanism [8] which confirms the recommendation from IETF, but no guidance is given as to which one to use. As suggested by [9], in order to compete with present responsive flows (e.g. TCP flows) within the same class, it will be ideal for all competing flows to exhibit the same response behaviour, so that all flows have an equal share of network resources. This is expressed as being 'TCP-friendly'. Otherwise, diverse response behaviours between competing flows can still result in unfairness and degrade the overall performance of the network and of the quality of the competing flows [10, 9, 3]. Multimedia traffic should share the network resources with TCP-based traffic in an equitable fashion. [1] states that congestion control for multimedia applications remains critical for the health of the Internet even if resource reservation or differentiated services become widely available. These schemes are likely to be provided on a per-class basis rather than per-flow basis. Thus, different users that fall into the same class of service still compete as in 'best-effort' networks. Furthermore, there will remain a significant group of users who are interested in using real-time applications over best-effort services due to lower cost or lack of access to better services. Therefore, it is vital that multimedia applications incorporate Congestion Control Mechanisms in their architecture [1].

Recent studies in TCP friendly Congestion Control Mechanism (CCM), notably TCP Friendly Rate Control (TFRC), have demonstrated the ability to provide smooth changes in terms of data sending rates in response to network congestion [11] and no retransmission upon packet loss. These CCMs are referred to as "TCP friendly" because they aim to achieve a similar bitrate to a TCP source under similar congestion conditions. Other end-to-end Congestion Control Mechanisms such as RAP [12] have been developed but they have not been able to provide smoothness in the sending rate in the same way. In the literature, TFRC is considered as the leading candidate at present.

Although a number of mechanisms have been proposed in the literature, emphasis has been placed on achieving 'fairness' and responsive behaviour among competing adaptive flows such as TCP. However, little attention has been paid to the end-to-end QoS requirements for applications using such Congestion Control Mechanisms, particularly for real-time interactive applications.

The problem of utilising the TFRC based CCM is the mismatch between the Packet Rate of the CCM and the Frame Rate of the encoder. This issue arises because the response to network congestion is carried out in different ways: the CCM reduces the Packet Rate, whereas the encoder reduces the Frame Size. Hence, the Packet Rate will be slower than the Frame Rate of the encoder. This results in packets buffered at the sender, and hence additional packet delay. Additionally, once the buffer at the sender is full, all incoming frames are discarded until buffer space becomes available again. This obviously will lead to significant quality degradation of the perceived multimedia streams.

Additionally by reducing the Packet Rate (PR) this results in an increase in the inter-packet-gap-spacing (IPGS) of packets sent[1]. The Frame Rate (FR) remains fixed and the time interval between frames, known as the Frame Interval (FI) also remains fixed. However, in the case of a reduced Packet Rate the IPGS is larger than that of the Frame Interval (FI). Hence, the difference between the two intervals (IPGS and FI) is added delay. This results in packets arriving with a larger IPGS, which demands for larger Playout (de-jitter) Buffers (PB), otherwise smaller Playout Buffers will lead to packets being discarded as they may not arrive in time sequence for playout. In addition a

[1] Note: IPGS is inversely proportional to PR

larger Playout Buffer means increased delay, which can lead to a reduced Quality of Experience (QoE) for an interactive service.

Because the TFRC CCM schedules ‘packets per unit time’, not by ‘bytes per unit time’, this adaptation architecture is not suitable for multimedia applications whose encoder adapts its frame quality (i.e. Frame Size with respect to bytes) rather than its Frame Rate.

With the recognition of the issues and challenges (a mismatch between the Packet Rate and Frame Rate), this thesis proposes a novel Congestion Control Mechanism (CCM) which integrates with a new Multimedia Adaptation Architecture (MAA). This is referred to as, ‘TCP friendly rate control – Fine Grain Scalable’ (TFGS). With TFGS the Congestion Control Mechanism responds to congestion by truncating the Packet Size while maintaining a fixed Packet Rate, i.e. reducing the effective bitrate in ‘bytes per unit time’. The outcome is that the Packet Rate (of the CCM) is equivalent to the Frame Rate (of the Encoder). Packets are scheduled as soon as they are generated, eliminating waiting delay of the packets at the sender side and of loss when the buffer becomes full. This approach provides an isochronous service which is of crucial benefit to interactive real-time services, such as voice.

The TFGS CCM responds to congestion by truncating the Packet Size. It incorporates a Multimedia Adaptation Manager (MAM) which makes use of Fine Grain Scalable encoding allowing the multimedia frame to be truncated to the packet size at byte-level granularity, as requested by the Packet Size Truncation (PST) function of the CCM. Hence, a precise match can be achieved between the application and network supply.

Using the TFGS MAA the quality of the multimedia frame may be compromised but the end-to-end interactivity is maintained. The TFGS ‘Multimedia Adaptation Architecture’ (MAA) is able to integrate the four main components of a multimedia system: (*1*)

Multimedia Application, (*2*) Multimedia Encoder, (*3*) Multimedia Adaptation Manager (MAM) and (*4*) Congestion Control Mechanism (CCM). This integration provides the capability for true on-the-fly adaptation of the multimedia stream, which enables it to meet interactive QoS requirements, along with achieving fairness amongst competing flows.

The thesis investigates this novel TFGS MAA via a quality measurement scheme and with a simulation study. This enables a quantification of the benefits in the form of a Quality of Experience (QoE) measure for the end-user.

The quality measurement scheme referred to as the E-model is used. This enables the quantification of degradation arising from Packet level Impairments such as packet loss, and delay over a scale of 0 to 100. However, to quantify byte-level impairment caused by frame/packet size truncations, which maps byte loss to an R-value, required a number of transformation processes; from Encoder bitrate to Frame Size and then its respective MOS quality to R-value. This novel formulation enabled the quantification of both Packet and Byte level Impairment into a scalar form. Further details can be found in chapter 6.

The simulation study is conducted over different traffic mixes and over a range of flows, and by using the quality measurement scheme the performance of each flow is quantified. This provides a comparison illustrating the benefits of the two types of MAA's.

1.2. <u>Thesis Organisation</u>

The thesis is organised into 8 chapters, including the introduction as chapter 1. Chapter 2, “Network Congestion”, discusses the impact of congestion on the QoE for multimedia applications and it reviews various solutions, finally concluding which one will be used to address the problem, in the remaining part of the thesis. Chapter 3, “Multimedia QoS Requirements, Adaptation & Architecture”, explores in detail the requirements imposed by multimedia applications with regard to transporting them over an IP network. It then discusses various encoding techniques that can be used to adapt to network congestion. Finally, it illustrates the multimedia architecture highlighting the various components involved from the end-to-end (i.e. from mouth-to-ear).

Chapter 4 outlines the TFRC CCM and its Multimedia Adaptation Architecture (MAA) as recommended by the IETF internet draft document “Strategies for streaming multimedia applications using the TCP-friendly rate control”. The second part of the chapter introduces the novel TFGS CCM with its multimedia adaptation architecture. Both the schemes are evaluated in chapter 6, where chapter 6 introduces a quality measurement scheme to quantify the degradation arising from packet loss, delay and byte loss on a scale of 0 to 100 with units of R-value. The method of quantifying the QoE impact of byte loss is novel and this is of significant importance to adaptive multimedia encoded schemes such as MPEG-2 and FGS.

Chapter 5, “Simulation Study” is organised in three main sections: the first section describes how the TFGS code is implemented in *ns*2, the second section goes into the detail of the simulation methodology, elaborating on how the measurement is done in *ns*2, and what parameters are used for traffic resources. It also highlights the network scenario description. The third and last section verifies that the CCM operates correctly and in the manner designed.

To give an idea of the number of packets generated over a simulation study where the total number of flows were 30, made up of 15 TFGS flows and 15 TCP flows. And this simulation repeated 25 times, resulted in a total of 1.2million packets generated, see Table 11.6 (in Appendix III, Simulation Runs).

Chapter 7, evaluates the two types of MAA using a simulation study where the simulation is conducted in a Homogenous and Heterogeneous traffic mix over a range of flows.

Finally, chapter 8 draws the final conclusions, presents some additional concluding remarks, and identifies some future research directions beyond the work presented in this thesis.

1.3. **Novelty Classification**

This thesis developed a novel Multimedia Adaptation Architecture (MAA) known as 'TCP friendly rate control – Fine Grain Scalable' (TFGS). This MAA is able to maintain an isochronous service by sending frames as soon as they are generated, i.e. by maintaining the Packet Rate of the Congestion Control Mechanism (CCM) at a level equivalent to the Frame Rate of the Multimedia Encoder. The response to congestion is to truncate the multimedia frames, which is requested by the Packet Size Truncation (PST) of the TFGS CCM.

By exploiting the flexibility of Fine Grain Scalable (FGS) encoders where the quality of a frame can be adapted (truncated) after encoding, the Multimedia Adaptation Manager (MAM) of TFGS is able to adapt the stream instantaneously 'on-the-fly', without needing to re-encode the frame. The MAM takes full advantage of this functionality by truncating the frame as requested by the PST function to ‘byte-level’ precision, achieving a true match to network supply, and better Quality of Service for the end user.

The thesis introduces a novel formulation for adaptive voice encoders in order to quantify the degradation arising from frame truncation, when responding to congestion used by the TFGS CCM. This formulation is able to integrate with the ITU-T E-model which assesses the QoE of a voice call from packet loss, delay, and other impairments. Hence, the complete quality measurement scheme is able to quantify the end-to-end QoE for the end-user when a voice flow is transmitted over the IP network regardless of which MAA is used.

2. Network Congestion

This chapter starts by illustrating how the quality of a voice call will suffer because of network congestion. It leads on to the options available to deal with this and explores in detail the specific solution that will be considered in this thesis.

What is Network Congestion?

Network congestion occurs when the required bandwidth by offered traffic demand exceeds the available network resources. In packet-switching networks, this is interpreted as the phenomenon that the packet arrival rate in a router surpasses the maximum packet service rate. As a result, the packets that cannot be immediately served will be temporarily queued in the router buffer which causes packets to experience a period of waiting time (i.e. queuing delay) before being processed. When a network is heavily loaded, the router's buffer becomes fully occupied which leads to discarding incoming packets.

Figure 2.1 illustrates the aforementioned network congestion phenomenon. The communication path comprises of two routers (R1 and R2) with the link capacity on the order of Kbps between the link 'R1-R2', which consequently makes this link the bottleneck. Once the offered load from R1 exceeds the link capacity of 'R1–R2', a packet queue will incrementally develop at router R1 and start dropping packets once the queue length reaches the buffer size. Such packet buffering and discarding behaviour results in packet queuing delay and packet loss. If this condition remains persistent for a period of time, it may lead from network congestion to network collapse (more commonly known as 'congestion collapse') [3].

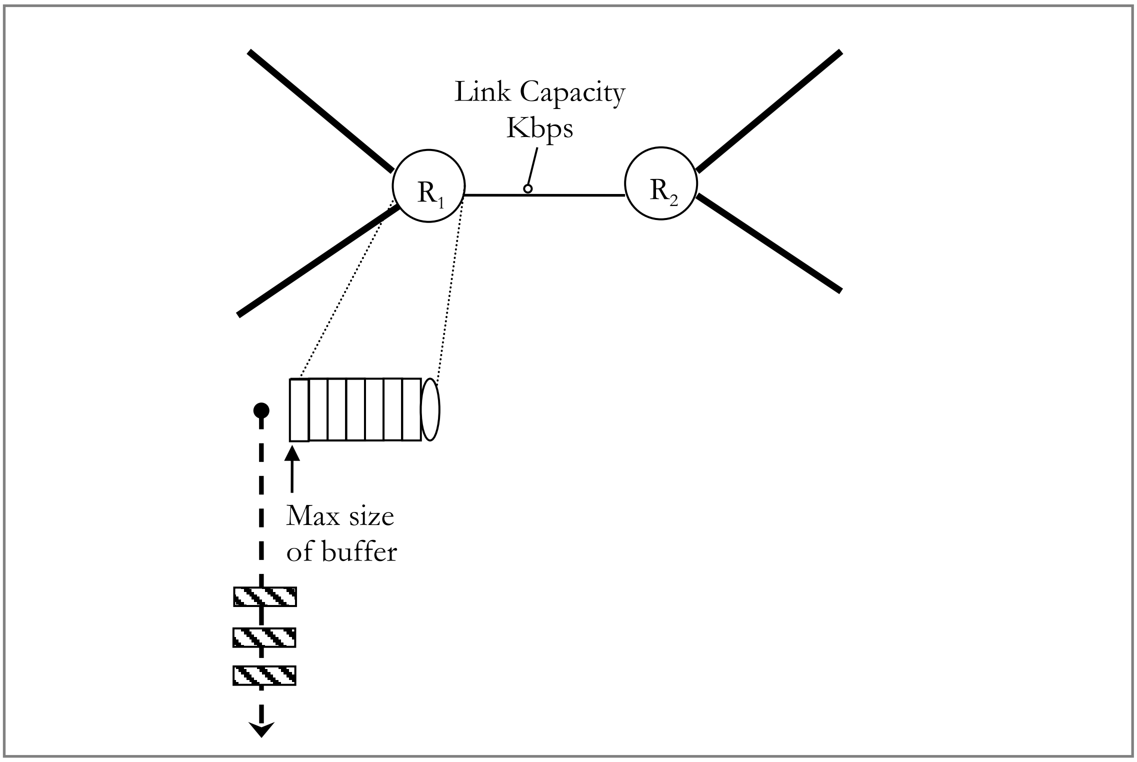


Figure 2.1, Network Congestion

2.1. Impact of Network Congestion: Loss, Delay, Jitter and Quality of Experience (QoE)

Interactive multimedia applications are considered sensitive to packet delay, delay jitter and loss. In the best-effort Internet service paradigm, packets are delivered from their source to destinations as quickly as possible without any notion of Quality of Service (QoS). It just does its best. Therefore, it lacks the control on delay, jitter and loss behaviours on the end-to-end path and guarantees of service quality. This presents tremendous challenges to the operation of real-time multimedia applications over the existing Internet infrastructure. This section defines and explains in detail how and where loss, delay, and jitter occur, and introduces an analytic tool known as the E-model to interpret these metrics into a scalar measure for quantifying user perceived voice service quality [13].

2.1.1. Loss

Loss of packets can occur both in the network and at the end-systems (Sender and Receiver). Loss is a sign of network congestion. For example, at the receiver the arrival of 'too-late' packets can occur when the packets pass through several routers. It is possible that many of those routers are very busy, and so packets have to wait in a queue for some time before they are serviced. By the time they arrive at the receiver it is 'too late' for some of them to be played out and in such a case the packet is dropped by the receiver. Packets that arrive are temporarily buffered, in what is referred to as a Playout Buffer (PB). It is normally the PB which checks the timestamp and sequence number on each packet when it arrives, and decides whether to buffer the packet for playout or discards them.

In the network, it is possible that one or more of the buffers in the route from source-to-destination are full and cannot accept any newly arriving packets. In such a case the packet is dropped by the router; i.e. the packets never arrive at the receiving application.

At the sender side, loss can occur if the application employs a Sender Buffer (SB), which is used as a temporary buffer between the two transmission rates: application (encoder) rate and transport rate.

Hence, the Total Loss Ratio, $L_{T(R)}$, (as a fraction of the total number of packets generated by the multimedia application, T_{NPG}) experienced by the multimedia stream is the addition of: Sender Buffer Loss, S_L, Network Loss, N_L, and the loss of packets that have arrived 'too late' at the Receiver, R_L.

$$\mathbf{L_T = S_L + N_L + R_L} \qquad \textit{Equation 2.1}$$

The Total Loss Ratio is:

$$\mathbf{L_{T(R)} = \frac{L_T}{P_{NPG}}}$$

$$\mathbf{L_{T(R)} = \frac{S_L + N_L + R_L}{P_{NPG}}} \qquad \textit{Equation 2.2}$$

The impact of packet loss on the Quality of Experience (QoE) can depend on at least four factors:

1. Percentage of packet loss, i.e. the number of packets lost over total packets sent
2. Packet loss distribution. Are the packets being lost in a random fashion over time, or are their 'bursts' of consecutive packets lost?
3. Packet Size. Applications that group multiple frames into a single large packet are more vulnerable to quality degradation than a packet loss than those which carries a single frame.
4. Packet loss concealment strategy, i.e. the strategy used to 'fill in' or conceal the lost packet.

Voice applications, for example, can tolerate packet losses of 10% if losses are experienced in a random fashion. However, if packet loss occurs in a 'burst' the quality degradation is more significant: worse than random losses. This is because bursty packet loss results in a larger segment of speech being lost or distorted, causing impairment that is much more noticeable to users. Furthermore, the larger the Packet Size, where one packet carries multiple voice/video frames, the harder it becomes to conceal the loss, and hence, the worse the quality degradation.

The actual pattern of lost packets seen on a real IP network is variable. It depends on the traffic load, and the moment-to-moment state of the network, resulting in both random and bursty patterns of loss.

If, for whatever reason a scheduled packet is not available for playout at its scheduled time, the result is silence in the conversation (or other forms of packet conciliation, such as replaying the same packet). When the overall packet loss percentage approaches 10%, especially where the packets are being lost in 'bursts', silence gaps in the conversation can be enough to degrade the quality significantly [14].

2.1.2. Delay

End-to-end delay, D_T, is the accumulation of sender processing, network and receiving processing. This is defined as follows:

a) The 'Sender Processing Delay' consists of: 'E_D' is encoding delay, 'Z_D' packetization delay, 'S_D' sender buffer delay.
b) The 'Network Delay' consists of: G_D' propagation delay, 'Q_D' queuing delay, 't_D' service time.
c) The Receiver Processing Delay consists of: R_D' playout buffer delay, '$Z_{D(R)}$' de-packetization delay and '$E_{D(d)}$' decoding delay.

Equation 2.3 summarises the total end-to-end delay, D_T, experienced by an individual packet in milliseconds (ms).

Total Delay = (Sender Processing Delay) + (Network Delay) + (Receiving Processing Delay)

$$D_T = (E_D + Z_D + S_D) + (G_D + Q_D + t_D) + (R_D + Z_{D(R)} + E_{D(d)}) \qquad \textit{Equation 2.3}$$

Encoding delay is the time required to digitize a raw analogue multimedia signal, by producing a stream of frames at a fixed interval. Decoding delay is the time required to convert the digital signal back to an analogue so it can be heard/seen by the receiving end. This delay is subject to processor constraints, i.e. it is dependent on the hardware specification, for example a mobile/PDA having a slower processer than that of a desktop computer will experience a larger encoding delay, but this is a fixed delay which remains constant throughout the duration of the connection. Hence, the delay on a mobile/PDA will be larger than that of a desktop.

(De-)Packetization delay is equal to the Frame Interval, F_I, between the multimedia frames, and this interval is inversely proportional to the Frame Rate, F_R, of the encoder. A higher Frame Rate will result in a shorter time interval, and hence a lower delay.

$$\mathbf{Z_{D(R)} = F_I = \frac{1}{F_R}}$$ *Equation 2.4*

The Sender Buffer or Receiver (also referred to as the Playout) Buffer, are buffers in place at the end-systems to temporarily buffer packets that will be sent into the network or played out to the user. The sender buffer is usually in place to act as a temporary buffer between the two rates: application and transport. The receiver buffer is used to accommodate varying delays (also known as delay-jitter) introduced by the network. This buffer acts to smooth out delay variations that are present in the network. This is particularly important for voice applications, where speech must be delivered (heard) at a constant rate. Such a buffer is also necessary to give the receiving application the ability to re-order any packets that have arrived out of order and discard if ‘too-late’ in time sequence.

Propagation delay is the constraint of the physical layer, which is used to transmit data across from source to destination. It can be computed as the ratio between the link length and the propagation speed over the specific medium. Propagation delay = $d/_s$ where 'd' is the distance and '*s*' is the wave propagation speed. In copper wires the speed 's' is typically ⅔ of the speed of light. This delay is considered to be a fixed delay throughout the duration of the voice/video connection (taking into account that all packets take the same route from source-to-destination).

Queuing delay is the time it takes for a packet to wait in the queue (also called the buffer) before it can be serviced. It is a key component of network delay. This term is most often used in reference to routers. When packets arrive at a router, they have to be processed and transmitted. If packets arrive faster than the router can process them (such as in a bursty transmission) the router puts them into the queue until it can manage to service them. The longer the line of packets waiting to be transmitted, the longer the average waiting time is. So, it is much preferred to have a shorter buffer, although this could result in an increase in dropped packets, which is also a sign of congestion. In response, the application should reduce its transmission rate.

Service time is the time it takes to process one packet in the queue, and this is a ratio between the Packet Size and the bitrate of the physical link (i.e. copper wire or optical fibre).

When the total delay starts to exceed 150 ms for an interactive voice conversation, this impedes the 'naturalness' of the conversation, see Table 3.1. In such a case the ability to have a phone conversation that resembles a face-to-face conversation, is increasingly lost. A delay exceeding 400 ms will result in a conversation that appears to be half-duplex, where two people are taking turns to talk [15]. This can be annoying, and is referred to as low-interactivity.

2.1.3. Jitter

As packets traverse through the network, they pass through several router buffers (queues). Some will be busier than others causing packets to experience varying network delays.

Consider two consecutive packets within a talk-spurt in an audio application. The sender sends packets at an inter-sent-time-spacing of 20 milliseconds (ms) (i.e. a packet rate of 50 packets per second). The first packet arrives at a nearly empty queue (less busy queue) at the router, but just before the second packet arrives at the queue, a large number of packets from other sources arrive at the same queue. As a result, the first packet experiences a small queuing delay whilst the second packet suffers a large queuing delay at the router. Consequently, the spacing between these two consecutive packets will be greater than 20 ms.

On the other hand, the packet spacing of these two packets can also result in an interval less than 20 ms under certain circumstances. Suppose the first packet arrives at the queue and the second packet arrives at that queue just before the entry of other packets from different sources at the same queue. In this case, the second packet is right behind the first one. If the time it takes to service a packet on the router's outbound link is less than 20 ms (which is commonly known as the service time in queuing terminology) this will result in a packet spacing of the two packets smaller than 20 ms, simply because the packets are behind each other.

The arrival of packets at varying network delays leads to a non-isochronous service which imposes a demand of the use of playout buffer at the receiving end system to remove the deteriorated effects of such variations. This means that receiving packets will be buffered for a sufficient time i.e. adding delay before playout can begin.

2.1.4. Quality of Experience (QoE) Assessment

In order to assess the quality of a voice connection in the presence of impairments it is necessary to consider the subjective rating that listeners would give. This subjective quality measure is referred to as a Mean Opinion Score (MOS) and is given on a scale of 1 to 5, as defined in [16]. Figure 2.2 shows the mapping of MOS to user satisfaction, as reported in [13] and [17].

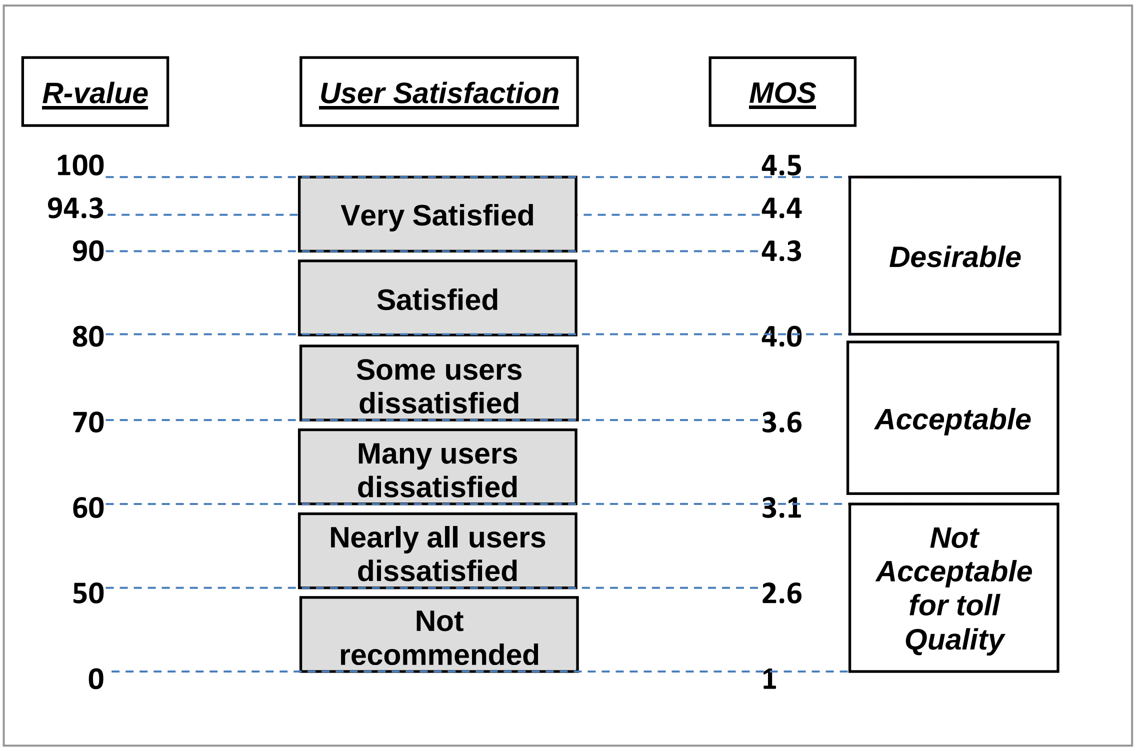


Figure 2.2, Mean Opinion Score (MOS) and its relation to user satisfaction defined by the E-model in terms of R-value (adapted from [13] and [17])

Numerous studies have been conducted to assess the effect on voice quality of various impairments under various conditions. Some of them have been compiled into reports and recommendations published in ITU-T standards.

These reports (here referred to as models) can be used to predict customer opinion when a new architecture or technology introduces impairments. The most popular model of this kind today is the E-model [18] from the number of models reviewed in [19].

The E-model can be used to interpret packet delay and loss behaviours into a scalar measure, representing the perceived quality of voice calls. The formulation of the E-model can be shown using Equation 2.6 and [18] which goes into detail of the various components involved in representing the perceived quality on a scale of 0 to 100 (known as the R-value). The parameters included in the computation of the R-factor are fairly extensive, covering such factors as echo, background noise, signal loss, codec impairments, delay and others. The relationship between MOS and R-value is expressed in the form of an equation, which can be found in Equation 2.5.

$$\begin{array}{ll} MOS = 1 & R < 0 \\ MOS = 4.5 & R > 100 \\ MOS = 1 + 0.035 \cdot R + 7x10^{-6} \cdot R(R-60)(100-R) & 0 < R < 100 \end{array}$$

Equation 2.5

A voice quality is rated over a scale of 0 to 100, which is referred as the R-value. The voice call with an R-value of 70 is considered to be of equivalent quality of a Public Switched Telephone Network (PSTN) voice call. An R-value lower than 60 is considered unacceptable.

$$R\text{–value} = (100 - I_S) - I_L - I_D + A$$

Equation 2.6

Codec Impairment, $(100 - I_S)$

'I_S' is the encoding impairment arising from different compressing techniques used (including quality degradation arising from noise, echo, and loudness), and this is different for each codec. For example a full rate fixed codec such as G.711 has a (100 –

I_s) R-value of 94.2, whereas a highly compressed codec such as G.729 generates a bitrate of 8Kbits compared to 64Kbits (of G.711) achieves an R-value of 77.3.

Packet Level Impairment, I_L, I_D

Packet Loss

'I_L' is the impairment factor arising from loss, where '$L_{T(R)}$' is the ratio of packets loss between source and destination (i.e. including sender buffer, network and receiver buffer loss). The loss (and delay) impairment is again codec dependent however, it is commonly found to follow the pattern modelled by [18] as expressed in Equation 2.7.

$$I_L \text{ (R-value)} = 30 \cdot ln(1+15 \cdot L_{T(R)}) \qquad \textit{Equation 2.7}$$

Packet Delay

'I_d' is the delay impairment factor including all the delays which occur between source and destination (i.e. including buffering at the Sender, Receiver and network). 'I_d' is modelled as [18]:

$$I_D \text{ (R-value)} = 0.024 \cdot D_T + 0.11(D_T - 177.3) \cdot H(D_T - 177.3) \qquad \text{Equation 2.8}$$

$$H(x) = \begin{cases} 0, \text{for x} < 0 \\ 1, \text{otherwise} \end{cases}$$

Here 'D_T' is the mean delay for all packets measured in milliseconds (ms) and H(x) is the Heaviside function.

Advantage Factor, A

'A' is the Advantage Factor; it accounts for users who can tolerate some degradation in quality in return for the ease of access, for example, when using a mobile or satellite phone.

Figure 2.3 illustrates how the two variables of delay and loss impairment can be represented as a quality rating value (known as the R-value) for a G.711 voice codec. For example, in order to achieve the minimum acceptable quality for a voice call, i.e. R=60, the voice call should not experience a mean delay of more than 200 ms. Its packet loss should be no more than 10 %.

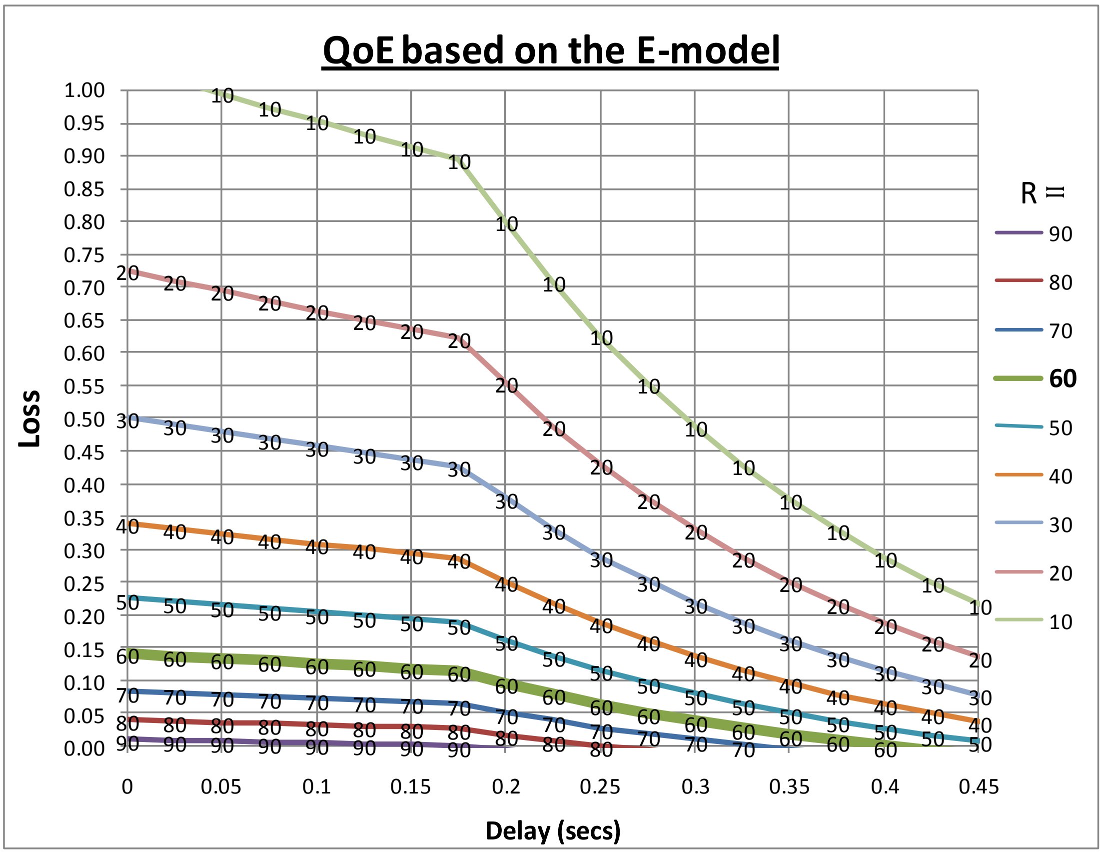


Figure 2.3, Impact of Delay and Loss on the R-value of a voice call

The larger the end-to-end delay the lower the interactivity between the two ends of the call. This reflects a poor Quality of Experience (QoE). Multimedia applications which use Forward Error Correction (FEC) to recover from packet losses in the network will no longer remain productive if a connection experiences a persistent packet loss greater than 10 percent [20, 1]. It is suggested by [1] the end-system should terminate the VoIP connection in such conditions.

2.2. Mechanisms to deal with Network Congestion

The impact that network congestion has on the quality of a voice connection resulting from loss and delay can be seen from Figure 2.3. There is a need for Quality of Service (QoS) mechanisms to deal with this.

The ideal way to avoid congestion is to ‘over provision’ the network, i.e. to increase network (bandwidth) resources. However, if extra resources are not available, then congestion must be tackled as and when it occurs. This may necessitate a number of schemes in place to deal with the problem satisfactorily.

Network congestion can be dealt at different layers of the TCP/IP stack. Application: where the multimedia application adapts its bitrate in order to reduce the load injected into the network. Transport: controls the amount of load injected into the network by buffering the application load until sufficient bandwidth is available. Network: allocate different amount of resources based on type of traffic, and discard packets when load exceeds certain thresholds.

2.2.1. Application Layer

2.2.1.1. Adaptive Multimedia Encoding

Using Adaptive Multimedia Encoding schemes enables applications to reduce their input load in the presence of network congestion. This is achieved by reducing the quality of frame being sent. A reduced quality frame means a smaller Frame Size;

hence the overall (encoder) bitrate injected into the network is lower. A lower encoder bitrate will reflect a degraded multimedia quality. However, the multimedia application will be able to maintain its interactivity and intelligibility by avoiding the loss and delay of packets during network congestion.

A number of adaptive multimedia encoding schemes are discussed in section 3.3, ranging from Layered, to Coarse Grain and Fine Grain Scalable encoding. The strength of adaptability offered by the multimedia encoding is solely dependent on scalability of the encoding technique. All this will be highlighted in section 3.3, which concludes that Fine Grain Scalable encoding is more adaptive than the other encoding schemes.

2.2.2. Transport Layer

2.2.2.1. Transmission Control Protocol (TCP)

'Adaptive Traffic', such as TCP, is able to self-control (limit) its input load by adapting its window size. This results in a reduced input load into the network. Such applications do not require segregation of traffic to achieve equal share of network resources. The built in mechanism in the TCP transport protocol referred to as 'Additive Increase Multiplicative Decrease' (AIMD) adjusts the window size based on network congestion indicators such as loss and delay.

Using TCP a single packet loss reduces the TCP window size to half, and increases its window incrementally when bandwidth becomes available.

The halving of the window size (assuming Round Trip Time 'RTT' and other variables remain constant) results in halving the sending rate, and this method of congestion response is not adequate for multimedia applications such as voice and video, because sudden changes in sending rates show noticeable effects on recipients' quality. Many multimedia applications do not run over TCP for this very reason – they do not want their transmission rate throttled back. Instead, these applications prefer to run over a User Datagram Protocol (UDP), which does not have a built-in Congestion Control Mechanism. When running over UDP, applications can send their audio and video into the network at a constant rate, occasionally losing packets, rather than reduce their bitrates to 'fair' levels at times of congestion. From the perspective of TCP, the multimedia applications running over UDP are not being fair – they do not cooperate with the other connections nor adjust their transmission rates appropriately. Because TCP congestion control will decrease its transmission rate in the presence of congestion (loss), while UDP sources need not, this leads UDP sources to crowd out TCP traffic.

2.2.2.2. Congestion Control Mechanisms (CCM): AIMD, TFRC

The essence of congestion control is to give feedback to the sender about events caused by congestion, so that the sender can adjust its sending rate accordingly. The feedback comes from the receiver, in the form of acknowledgments sent via RTCP packets for example, notifying the degree of loss and delay occurring over the network.

The control system of the CCM is illustrated in Figure 2.4.

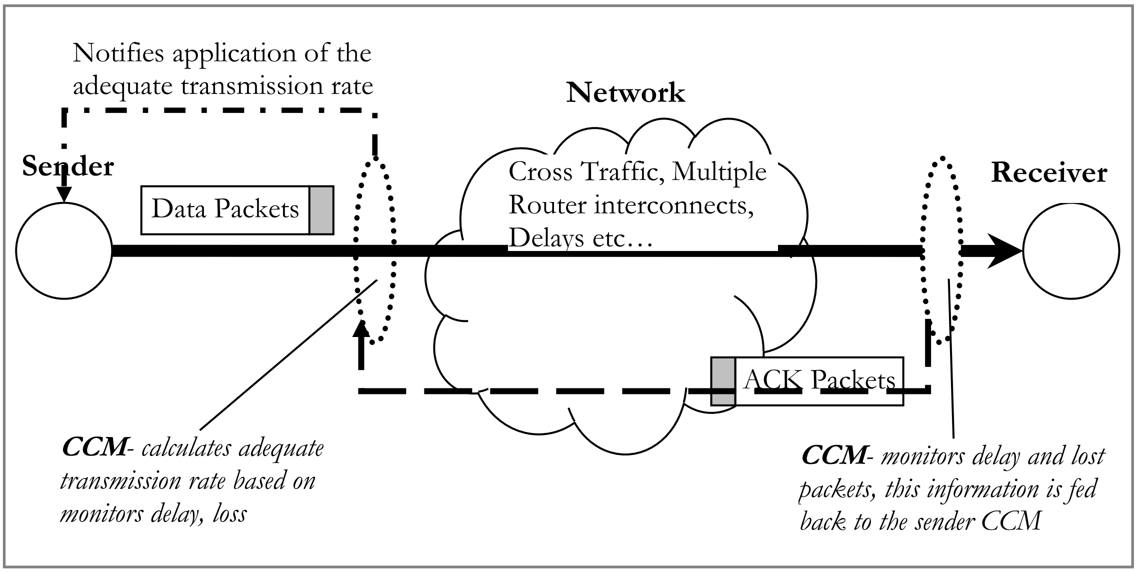


Figure 2.4, Congestion Control Mechanism (CCM) Control System

The two well-known CCM are AIMD and TFRC. AIMD is based on the congestion avoidance mechanism of TCP, however with different increase and decrease parameters. TFRC is based in a TCP-rate equation model defined by [31], which models the TCP bitrate using an equation.

1) Additive Increase Multiplicative Decrease, 'AIMD(a,b)'

AIMD-based (Additive Increase Multiplicative Decrease) congestion control mechanisms are TCP-compatible, in that they compete reasonably fairly with existing TCP, but that avoid TCP's halving of the congestion window in response to a single packet drop. TCP's congestion control mechanisms are a good choice for most current applications, as TCP is very effective at rapidly using bandwidth when it becomes available. However, for some applications the requirement for relatively smooth changes of the sending rate is more important than the ability to make opportunistic use of increases in available bandwidth. For such applications, a key reason not to use TCP's congestion control mechanisms is to avoid the abrupt halving of the sending rate in response to a single packet drop.

AIMD(*a,b*) congestion control refer to pure AIMD congestion control that uses an increase parameter '*a*' and a decrease parameter '*b*'. That is, after a loss event the congestion window (when speaking in the context of TCP) is decreased from '*W'* to '*(1-b)W'* packets, and otherwise is increased from '*W'* to '*(W+a)'* packets after each Round Trip Time. Currently, TCP uses AIMD(1, ½) congestion control along with the congestion control related mechanisms of the retransmit timer, and the exponential back-off of the retransmit timer in periods of high congestion. Given the long familiarity in the Internet with TCP, the most obvious choice for a congestion control mechanism that reduces its sending rate more smoothly than TCP would be AIMD(*a,b*) but with a decrease parameter less than ½.

[21] has shown that TCP(1/5, 1/8) and TCP(2/5, 1/8) compete fairly equally with TCP, while avoiding TCP's reduction of the sending rate in half in response to a single packet drop. This can be seen in figure 3 and 4 of [21].

Although AIMD provide TCP-friendliness and better smoothness compared to TCP. However, AIMD is not as smooth when compared to TCP-rate Equation congestion control, TFRC, this can be seen from figure 10 of [21].

2) TCP-rate Equation: TFRC CCM

The TFRC congestion control mechanism (CCM) is based on the Rate Equation model of TCP which indicates a transmission rate equivalent to a TCP source under similar congestion conditions. The basic Rate Equation model was originally developed by [31] and later further improvements were made which took into account 'retransmits timeouts' and probabilistic drops. The present Rate Equation model is of [22] and is defined as (which is commonly known as the TCP-rate equation):

$$T_{TCP} = \frac{S}{t_{RTT} \times \sqrt{\frac{(2p)}{3}} + t_{RTO}\left(3\sqrt{\frac{(3p)}{8}}\right)p\left(1+32p^2\right)} \qquad \textit{Equation 2.9}$$

Where 'S' is the fixed packet size, 't_{RTT}' is the Round Trip Time, 't_{RTO}' is the TCP retransmission timeout (set as 4 x t_{RTT}), and *'p'* is the packet loss rate. Using these measured parameters, the equation-based congestion control mechanism updates its transmission rate, T_{TCP}, every Round Trip Time.

The TFRC CCM uses the same equation but redefines the variable packet loss rate, '*p*', to loss event rate, 'l', which enables a smoother impact on the transmission rate from packets losses. In addition, the measured round-trip-time (RTT) is smoothed using an exponentially weighted moving average which is defined as 't_{RTT}'. The updated TCP rate equation is shown below.

TFRC's purpose is to provide smoother changes in the sending rate, T_{TCP}, making it more suitable for multimedia applications compared to using AIMD CCM's. Details of the loss event rate algorithm can be found in section 2.4.1.1 and [9]. The proposed algorithm offers a good trade-off between responsiveness to changes in congestion and aggressively to finding and using available bandwidth

2.2.2.3. ***<u>Datagram Congestion Control Protocol (DCCP)</u>***

A transport protocol known as Datagram Congestion Control Protocol (DCCP) [23] aims to provide a suitable transport protocol for multimedia applications by enabling them to be congestion responsive. DCCP architectural design will provide TCP functionality but without the reliability.

DCCP intends to provide a wide variety of services to real-time interactive applications at transport level. To name a few it intends to: a) provide a plugin for Congestion Control Mechanisms (currently supporting AIMD [24] and TCP Friendly Rate Control 'TFRC' [12]), b) provide connection handshake for setup and teardown of connections, similar to signalling protocols such as SIP and H.323. This will also prove to be firewall friendly. c) Provide sequence numbering to packets and acknowledgments of packets sent, similar to what RTP and RTCP currently provide.

Further developments of the protocol can be found at [25].

2.2.3. Network Layer

2.2.3.1. Weighted Fair Queuing (WFQ)

Using a simple First Come and First Out (FIFO) queue all in bound packets are placed in a single queue, regardless of the size, or type of arriving packets. All inbound packets are placed in one queue, which operates on the principle of FIFO. Weighted Fair Queuing (WFQ) on the other hand uses multiple queues to separate packets from various flows into different queues (known as classes). It can give equal (or different) amounts of bandwidth to each queue (which can also be defined as a class). This prevents one class of traffic from consuming all the available bandwidth, for example non-responsive multimedia flows saturating responsive flows such as TCP. However, this does not guarantee that flows within the same class share bandwidth fairly among themselves.

2.2.3.2. Congestion Avoidance

WFQ manages existing congestion, whereas Congestion Avoidance avoids congestion to develop. Congestion is avoided by dropping packets across different flows, this causes various applications to reduce their input load into the network. Random Early Discard (RED) is an example of such a Congestion Avoidance Mechanism [26].

RED avoids congestion in the router by maintaining an average queue size between two set thresholds, minimum and maximum [27]. To maintain an average queue size RED drops packets probabilistically. For example when the average queue size increases above the minimum threshold, RED drops each arriving packet with a certain probability. This probability is a function of the average queue size, as defined in [27]. By dropping these packets it would give indication to those particular connections to reduce its sending rate. Currently only TCP as a transport protocol responds to dropped packets by the router, other applications which use UDP require feedback from the end receiver's whether a packet was delivered or dropped. By dropping packets this should maintain an average queues size, if the queue size still continues to grow then more packets are dropped with a higher probability. If the average queue size exceeds the maximum threshold then all arriving packets are discarded, in a similar fashion to 'drop tail'.

RED also provides additional benefits and these are listed below:

1. Absorb burst traffic: Packet bursts are unavoidable in packet oriented networks [28]. If the queue space is at all times fully occupied, then the possibility to accompany bursty packets is impossible and promotes few connections to monopolize the queue space. This is known as the 'lock-out' effect. Therefore by keeping an average queue at all times will provide buffering for bursty packets and will always have space for incoming packets.

2. Global synchronization: The queue space available is shared by many different connections, therefore it is necessary to operate fairly among all connections. By operating the queue space at full capacity this results in too many incoming packets being dropped, hence creating a 'global synchronization' effect on too many connections at the same time. In the case of a TCP connection this causes multiple TCP sessions to go into slow-start. This behaviour reduces network performance, as the network can be seen to be under-utilised. In addition packet loss leads to retransmission which causes an adverse effect in reducing congestion. RED avoids this by randomly dropping packets causing some flows to slow down. RED then measures the effect on the queue, and if the reduction is not adequate, then more packets are dropped to cause more flows to reduce their rate.

3. Reducing Delay: By maintaining an average queue size, this encourages smoother flows of traffic within the router and reducing delay times between packets. This is particularly important for real-time applications, as they perform better at smoother sending rates and reduced end-to-end delay.

Furthermore, using Weighted RED (WRED) combines the capabilities of the RED algorithm with Class segregation. This combination provides for preferential traffic handling for different classes of traffic.

Due to RED's advantages for congestion aware connections such as TCP, RED has been deployed in most routers on the internet. Although RED is able to control queue lengths and reduce end-to-end delay, this is only possible in an environment where each connection responds to a dropped packet.

Presently the amount of real-time traffic is increasing over the internet. And most of these real-time applications use UDP as there transport protocol. By using RED, which indicates congestion by dropping packets this will have no effect on their sending rates. This is because UDP is a transport protocol which has no concept of congestion control. Therefore, research [2, 3, 23] suggests that applications should either invest in new transport protocols which provide both unreliable data transfer and respond to congestion, e.g. DCCP [23] or have UDP connections running with Congestion Control Mechanisms.

2.3. **Current Situation with Multimedia flows**

At present a substantial portion of the multimedia connections operating over the internet do not respond to congestion indicators. This is because they use User Datagram Protocol (UDP) as their underlying transport protocol. UDP provides no feedback on congestion indicators such as loss and delay. Real-time Transport Protocol (RTP) must be used along with UDP to facilitate such feedback.

Furthermore, multimedia applications require an isochronous service, i.e. they must send data as soon as it is generated, as such applications are time dependent. They send their data frames at fixed rates and adjust their transmission rates in terms of frame quality. Transport protocols such as TCP, or even some Congestion Control Mechanisms such as 'Rate Adaptation Protocol' (RAP) [12], are too harsh in response to packet losses, i.e. they will reduce their sending rate by half in response to a single packet loss. For that reason multimedia applications avoid using them. Additionally transport protocols such as TCP retransmit lost packets, which can take at least one additional Round Trip Time (RTT) to arrive; but multimedia applications are time critical in nature, and so delayed, lost and re-transmitted packets are of no use. In the literature the existing Congestion Control Mechanism (CCM) designers have paid little attention to multimedia application requirements and so such flows remain unresponsive to congestion indicators, (in other words they are non-adaptive or inelastic).

The increasing growth of multimedia applications could result in severe inter-protocol unfairness, e.g. UDP flows make well-behaved adaptive flows such as TCP suffer from resource starvation. This unfairness resource usage will lead to excessive packet loss and delay for both adaptive and inelastic flows, with significantly poor throughput for adaptive flows, and poor Quality of Experience (QoE) to inelastic multimedia flows. This behaviour can lead to a condition called Congestion Collapse [3, 29], where the network is ineffective because it is busy forwarding packets which are ultimately going to be dropped before they can reach their end destination.

2.4. Method chosen to address the: Congestion Control Mechanism and Adaptive Multimedia Encoding

The problem raised in the previous section highlights that flows will remain inelastic (unresponsive to congestion) until the issue of reducing the input load in congestion periods is successfully addressed. That is why it is imperative to devise a scheme which is able to provide a suitable Congestion Control Mechanism (CCM) for multimedia applications, which will make the ‘inelastic’ adaptive. This can be achieved by having a CCM which responds to congestion in a manner that is adequate for adaptive multimedia encoding.

Deployment of the other mechanisms such as WFQ and RED will not prove productive until the input load rate cannot be controlled at the source.

When using scheduling algorithms such as WFQ certain amounts of network capacity are allocated to each class of traffic. If flows within the same class are behaving unfairly among themselves because they are not responding to network congestion, this will lead to excessive losses for all flows within the same class. Therefore, it is in their own interest for multimedia flows to become adaptive.

Additionally, when using Congestion Avoidance Mechanisms such as Random Early Discard (RED), when packets of a particular flow are dropped, the congestion avoidance mechanism expects that the transport protocol or CCM to understand these notifications and responds accordingly, otherwise such mechanism will not prove productive.

Therefore, it makes sense that one should address the problem of making multimedia flows adaptive in order to control the input load into the network, before the WFQ and Congestion Avoidance Mechanism, RED, are put in place. Furthermore, the 'Internet Engineering Task Force' (IETF) recommends that all applications running over the Internet should use end-to-end congestion control so that long-term fairness of network resources can be achieved, and the stability of the Internet not be put at risk from congestion collapse [2, 3].

Congestion Control Mechanism

Hence, before considering the development of end-to-end Congestion Control Mechanisms (CCM), it is important to assess their overall design and behaviour. Both the end user (i.e. the multimedia application) and network requirements must be satisfied by the Congestion Control Mechanism; otherwise the risk of Congestion Collapse will remain and multimedia applications will continue to be unresponsive. It is in the best interest of both network and multimedia applications that the requirements of both are satisfied in a seamless manner. The essential features of a Congestion Control Mechanism (CCM) for interactive multimedia are as follows:

1. A CCM should be able to provide a fair share of bandwidth utilization for the flow, not only so that it is fair to other competing flows, but also so that it can aim to achieve high goodput.

2. A CCM should be able to take into account the congestion state of the full path from source-to-destination as various levels of congestion can be present at different intermediate nodes (routers) along the network path.

3. A CCM for interactive multimedia flows should not need to retransmit lost packets, as they are of no use to the end user. It is waste of bandwidth to retransmit them when they are not required.

4. A CCM should be able to co-exist and compete with adaptive flows such as TCP over the best-effort Internet service. Since TCP has been proven to be a successful protocol which has been fair with competing flows and has maintained the stability of the Internet along with keeping utilization high [3], it makes sense that multimedia applications should behave in a way that is 'TCP friendly'. Ideally, competing flows should adopt the same long-term response behaviour, so all flows have an equal share of the network resources and are compatible with each other. Otherwise different response behaviour may either be too aggressive or not aggressive enough, resulting in unfairness between competing flows and reducing overall network performance [3, 9, 10].

5. A CCM should provide smooth changes to transmission rates, making it easier for multimedia applications to adapt. The halving of transmission rates will not prove attractive to multimedia applications. Changes in transmission rates must

be done in a manner in which an isochronous service can still be maintained, i.e. multimedia data (frames) can be sent as soon as they are generated, i.e. a fixed Frame Rate can be maintained, so that delay and jitter can be kept to a minimum. Regardless of the challenges that loss, delay and jitter impose on multimedia applications, they are able to reduce transmission rates by adjusting the quality of the frame, whilst maintain a fixed Frame Rate.

A number of CCM have been proposed which can be found in [30], however at present 'TCP friendly rate control' (TFRC) is the leading mechanism for providing smoothness in its transmission rate. The TFRC CCM is based on the Rate Equation model of TCP which was developed by [31]. Using the same equation the packet loss variable has been redefined as 'loss event rate'. This ensures that packet losses have a smoother impact on the transmission rates. Furthermore, none of the lost packets are retransmitted. This saves bandwidth and reduces complexity on the Congestion Control Mechanism.

Additionally, using the Rate Equation model enables a response to congestion which can be referred as 'TCP friendly' because it aims to achieve a similar long term bitrate of a TCP source under similar congestion conditions. Results shown in [9] verify this claim. The loss and delay variables used in the Rate Equation model are based on the end-to-end state of the network. Therefore the transmission rate calculated by the 'Rate Equation' model takes into account the congestion levels of all nodes from source-to-destination.

Although the TFRC CCM possesses the four main traits as described above it however lacks the functionality of providing an isochronous service as described in point 5. In this thesis the main contribution and focus is to present an approach to implement this isochronous functionality. This is the one of the core novelties of this thesis. Details can be found in section 4.2.

Adaptive Multimedia Encoding

Network conditions vary over time and this reflects on the amount of network bitrate (bandwidth) available. Therefore, the ability for multimedia encoders to adapt in terms of bitrate will prove useful to them, as the application will avoid congestion to develop, hence reducing the amount of loss and delay experienced. Therefore, improving the connection's Quality of Experience (QoE).

The ability for multimedia applications to adapt their bitrate depends on the performance of the encoder's scalability. Section 3.3 highlights the three main forms of adaptive encoding techniques available: Layered, Coarse Grain and Fine Grain Scalable (FGS). In conclusion it seen that that FGS provide the best form of bitrate adaptability offering 'byte-level' scalability where the bitstream can be truncated to any length as required by the Congestion Control mechanism. This is because the multimedia data is organised in a manner or priority, from Most Significant Bytes (MSB) to Least Significant Bytes (LSB). The MSB represent the most basic but vital information (i.e. minimum quality), scaling up to the LSB which represent the enhancement of the basic information, (i.e. higher quality).

Hence, the multimedia application can provide a true match to the available network bitrate, providing the best possible quality to the end user whilst making best use of network resources available, this is illustrated in Figure 1.2. This encoding scheme is of great benefit in an environment such as the Internet, where available network bitrate is continually changing and the ability to adapt in a manner without causing step changes in quality is of significant benefit to the end user.

FGS bitstreams are encoded at their full bitrate; the encoded bitstream allows adaptation to take place after encoding, hence allowing on-the-fly adaptation of the bitstream without requiring the storage of multiple copies of the bitstream at different bitrates. This lends itself very reasonably to Congestion Control Mechanisms such as TCP-Friendly; section 4.2 illustrates in details how the use of FGS bitstreams will enable the Congestion Control Mechanism to maintain an isochronous service whilst responding to network congestion.

2.4.1. *TCP form of congestion response: TCP Friendly Rate Control (TFRC)*

2.4.1.1. Protocol Design

The TFRC protocol is characterised by three main functions: (*1*) Increase/Decrease Algorithm, (*2*) Slow-Start and (*3*) Loss event Rate calculation.

(1) Increase/Decrease Algorithm

Every time a feedback message i.e. an acknowledgment (ACK) is received the value of the sending rate, T_{TCP}, is updated using the TCP response function. If the calculated sending rate, T_{CSR} is greater than previous sending rate T_{PSR}, then the sender can increase its sending rate. On the other hand, if the calculated sending rate, T_{CSR}, is less than previous sending rate T_{PSR} the sender decreases its sending rate to T_{CSR}.

The increase or decrease in the Sending Rate, T_{TCP}, is achieved by varying the Inter-Packet-Gap-Spacing (IPGS) of packets sent, $P_{IPGS(S)}$. This is similar to how Rate Adaptive Protocol (RAP) responds to changes in the Sending Rate [12]. The IPGS is a function of Packet Size, P_S, and the sending rate, T_{TCP}.

$$P_{IPGS(S)} = \frac{P_S}{T_{TCP}} \cdot \sqrt{\frac{t_{RTT(o)}}{M}} \qquad \textit{Equation 2.10}$$

The multiplication of the '$t_{RTT(o)}$' and '*M*' parameters, are used to further reduce the oscillations in the Sending Rate making it ideal for multimedia applications and improving the network performance, further details can be found in [9]. '$t_{RTT(o)}$' is the most recent Round Trip Time (RTT) sample, and '*M*' is the square root of the average RTTs.

By taking the inverse of the IPGS it indicates the Packet Rate, P_R.

$$P_R = \frac{1}{P_{IPGS}} \qquad \textit{Equation 2.11}$$

The Packet Rate gives a measure of the number of packets sent per second (pps).

(2) Slow-Start

The TFRC slow-start is identical to the Slow-Start algorithm of TCP, where the sender roughly doubles its sending rate each Round Trip Time (RTT). The TCP's acknowledged clock mechanism provides limits the overshoot during the slow-start period. Hence, no more than two outgoing packets can be generated for each acknowledged packet, forcing a TCP connection to send no more than twice the bandwidth of the bottleneck.

A rate-based protocol does not have this natural self-limiting property. Therefore, a simple mechanism is used where the receiver feeds back the rate at which the packets arrive, T_{RSR}, (during the last measured RTT). And the sender's sending rate, T_{SSR}, is limited to the minimum of twice the receiver sending rate, T_{RSR}, or previous sending rate, T_{PSR}, whichever is smaller.

$$T_{SSR} = \mathit{min}\,(\,2\cdot T_{PSR}\;,\;2\cdot T_{RSR}\,) \qquad \textit{Equation 2.12}$$

The slow-start phase will terminate and go into the increase/decrease phase if a packet loss occurs.

(3) Loss Event Rate, l_{er}

A receiver aggregates packet losses occurring within one round trip time (RTT) into a loss event (LE). The number of packets between loss events is referred to as the loss interval (LI) and this is illustrated in Figure 2.5.

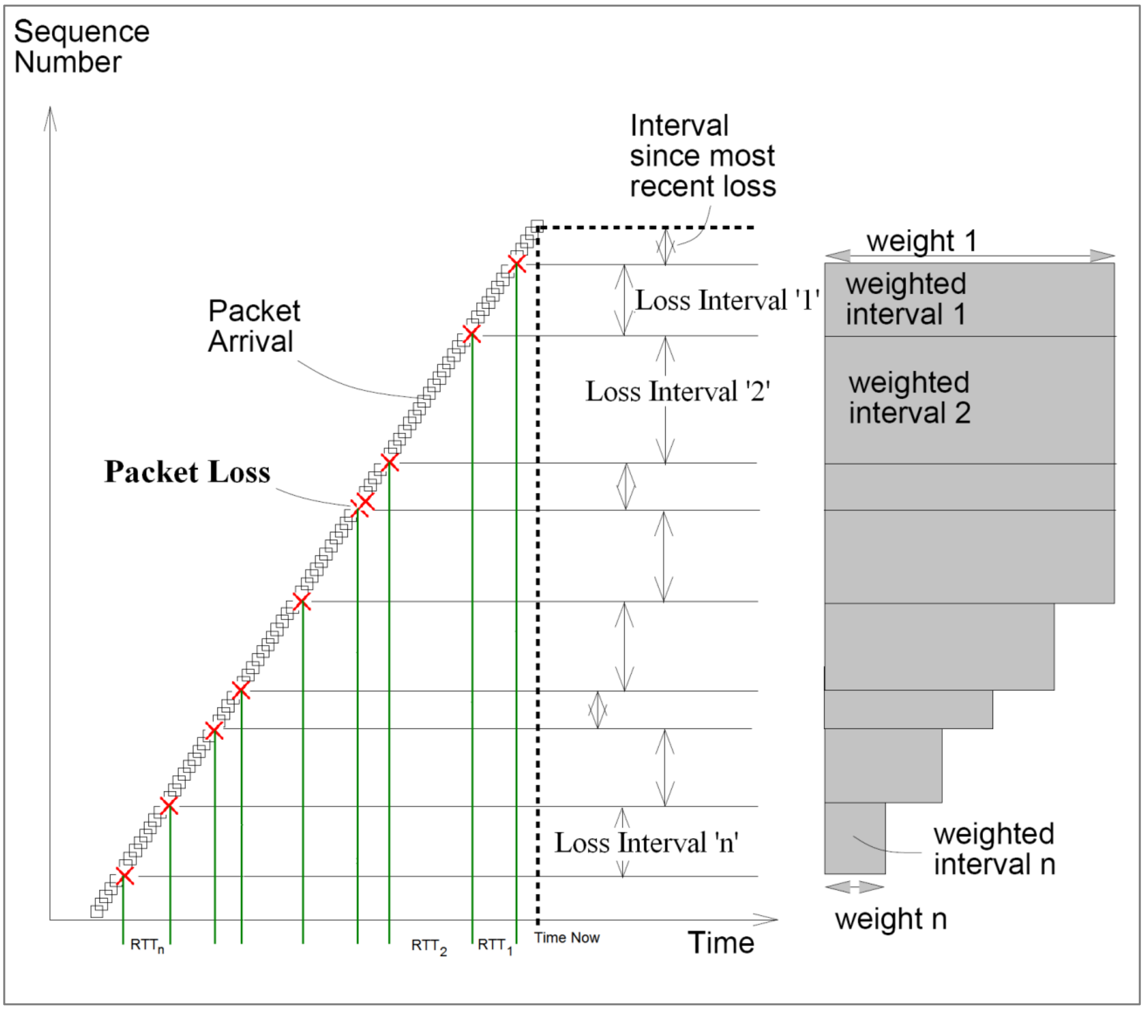


Figure 2.5, Weighted loss intervals between loss events used to calculate loss event rate, (figure adapted from [9])

The average loss interval, LI_{avg}, is calculated over a weighted average of the 'n' most recent loss interval as shown in Equation 2.13.

$$LI_{avg(1,n)} = \frac{\sum_{i=1}^{n} w_i LI_i}{\sum_{i=1}^{n} w_i} \qquad \textit{Equation 2.13}$$

The weights, 'w_i', are chosen so that the very recent loss intervals receive the same high weights, while the weights gradually decrease towards zero for older loss intervals. This allows for smooth changes in the average loss interval as loss events age. The choice of the weight values determines the trade-off between output rate responsiveness and smoothness. This problem was analysed in [9] and the values recommended are as follows:

For weights, w_i:

$$w_i = 1 \qquad \textit{for} \qquad 1 \le i \le \frac{n}{2}$$

and

$$w_i = 1 - \frac{i - \frac{n}{2}}{\frac{n}{2} + 1} \qquad \textit{for} \qquad \frac{n}{2} < i \le n \qquad \textit{Equation 2.14}$$

Where n = 8, has been demonstrated to provide a balanced trade-off, this gives weights of 1, 1, 1, 1, 0.8, 0.6, 0.4 and 0.2 for 'w_1' through 'w_8' respectively.

The calculated average loss interval, LI_{avg}, does not incorporate the most recent interval as illustrated in Figure 2.5. The full average loss interval, $LI_{full.avg}$, can be derived using Equation 2.15.

$$LI_{full.avg(0,n-1)} = \frac{\sum_{i=0}^{n-1} w_{i+1} LI_i}{\sum_{i=1}^{n} w_i}$$ *Equation 2.15*

To determine the inclusion of the most recent loss interval is to examine whether its average is greater than 'LI_{avg}' as calculated in Equation 2.13.

The loss event rate, l_{er}, is the inverse of the average loss interval as shown in Equation 2.16.

$$l_{er} = \frac{1}{\max\left(LI_{full.avg(0,n-1)}, LI_{avg(1,n)}\right)}$$ *Equation 2.16*

The smoothness factor is evaluated by [9] and it illustrates that the upper bound on the increase in the transmission rate is of a rate 0.14 packets/RTT under no congestion periods. Whereas for the lower bound, the decrease in the transmission rate takes approximately 5 persistent RTTs to half its transmission rate.

Furthermore, additional smoothness and responsiveness can be achieved in the transmission rate by using either or both of the following two schemes. The first is history discounting of the old loss intervals, if the most recent loss interval is twice the average loss interval, LI_{avg}. This condition is used to increase the transmission rate during less congested periods. The second scheme is smoothing the RTT value by averaging the RTTs using the Exponentially Weighted Moving Average (EWMA). Details of these schemes can be found in [9]. Nevertheless the core component of the TFRC CCM which is the 'loss event rate' calculation which provides the smoothness in its transmission rate.

2.5. **Summary**

This chapter defined network congestion and illustrated its impact on the quality of a multimedia connection. The Quality of Experience (QoE) Assessment model, referred to as the E-model, illustrates how serious the quality degradation can be from excessive network congestion leading to packet loss and delay. Therefore, a number of mechanisms were discussed that can be put in place at certain layers (Network, Transport, and Application) into the TCP/IP stack.

At the Application Layer the offered load into the network can be adjusted by using adaptive encoding, and the amount required to adjust the offered load can be indicated by using a Congestion Control Mechanism (CCM) which is present at the Transport Layer. The CCM monitors the state of the network based on packet loss, and end-to-end delay.

At the Transport Layer the rate at which the packets are sent into the network can be controlled and this rate is determined by the degree of packet loss, and delay that occur in the network. An example of a transport layer mechanism is Transmission Control Protocol (TCP). TCP adapts its rate in AIMD manner where it increases its rate incrementally (in non-congestion periods) and decreases its rate by half (in presence of congestion). This mode of congestion response is appropriate for applications that seek to make effective use of network capacity, but do not necessarily require timely delivery. In contrast real-time interactive applications require timely delivery, and smooth changes in the sending rate. Congestion Control Mechanisms such as TFRC can offer smooth changes in the sending rate however, lack the functionality of sending packets as soon as they are generated. This leads to a need for developing a novel

Congestion Control Mechanism which can offer an isochronous service whilst maintaining the traits of TFRC. Chapter 4 will investigate this further.

At the Network Layer packets can be segregated into different queues where the queues can be prioritized based on a class of traffic. This scheme is referred to as Weighted Fair Queuing (WFQ). A mechanism known as Random Early Discard (RED) can be placed in the queue which controls the behaviour of dropping. This is useful to avoid congestion developing at early stages, whereas WFQ manages existing congestion.

This chapter concludes with which mechanisms to be used in this thesis: a) TCP-friendly Congestion Control Mechanism (CCM), and using b) Fine Grain Scalable 'Adaptive Multimedia Encoding'. The novel Congestion Control Mechanism will indicate the available network bitrate and the Adaptive Multimedia Encoder will adjust its bitrate at byte-level granularity. This approach will result in a reduced offered load into the network during congestion, whilst offering the least level of multimedia quality degradation.

3. Mechanisms at the End system to deal with Loss, Delay-Jitter and Input Load into the network

This chapter addresses how the QoS of a multimedia application can be maintained by using several mechanisms to compensate for the impact of loss, and delay-jitter. Furthermore, it looks into adaptation techniques which can be used to adjust the bitrate of the multimedia application in order to reduce the input load into the network in the presence of network congestion. Following that, this chapter also presents the end-to-end multimedia communication architecture illustrating the components involved from end-to-end, in transporting multimedia data across an IP network.

3.1. Class of Services – Real-time Interactive Multimedia

Interactive Multimedia applications that operate over the Internet require a certain level of Quality of Service (QoS) for them to remain meaningful. Table 3.1 defines these levels of requirements for different classes with respect to delay and loss.

Class	Details	Mean Delay	Delay Jitter (De-Jitter Buffer)	Loss Percentage	Application Examples
1	Real-time, **Highly** interactive applications, sensitive jitter	150 ms	40 to 80 ms	10	Voice over IP, Video Teleconference
2	Real-time, interactive applications, sensitive jitter	400 ms	40 to 80 ms	10	Voice over IP, Video Teleconference
3	Low loss applications	1 sec	-	10	Video Streaming

Table 3.1, Class of Services (adapted from [15] table 11.1)

Delays in traditional circuit-switched environments are typically below 150 ms, and therefore highly interactive applications expect the same level of delay to achieve the same level of QoS [32]. However, with the evolution of IP networks multiple sources of delay can occur, ranging from packetization, to queuing, and de-jitter buffering. This increases the amount of delay experienced by multimedia packets. If the delay starts to exceed 400 ms the conversation appears to be half-duplex, (where two people are taking turns to talk). This can be annoying, and is referred to as low-interactivity.

Packets sent over the Internet can experience different levels of delay. In order to compensate for this delay, the receiving end usually employs a de-jitter buffer. Typical de-jitter buffers hold two to four packets and thus introduce an additional delay of 40 to 80 ms for 20 ms sized packets [15].

Voice transmissions exhibit a high tolerance for packet losses. If an occasional packet is lost, the fidelity of the voice reproduction is not severely affected. In contrast, data packets have a low tolerance for errors, and require retransmission if packets are corrupted or lost. In the event of excessive delays in the network, the packets may be discarded at the receiving end, because they are of no use if they arrive at the receiver too-late. Again, the loss does not severely affect voice fidelity if the lost packets are less than 10 percent of the total packets transmitted.

3.2. Mechanisms in the End system to maintain QoS

Section 2.1.4 in this thesis discussed the impact of loss, delay and jitter on the Quality of Experience (QoE) of a voice call. This section illustrates how the QoE can be preserved in presence of some loss, delay-jitter arising in the transport of multimedia data across the IP network.

3.2.1. Loss

When packet drops occur in the network during congestion the audio quality can greatly suffer. Furthermore, owing to the strict delay requirements of real-time applications, retransmission is not a feasible form of recovering lost packets. For this reason recovery schemes such as Forward Error Correction (FEC) have been developed to compensate for packet loss in the Internet. When packet losses occur the receiver can either fully or partially recover the media packet depending on what recovery scheme and degree of protection has been used [20].

FEC schemes are primarily targeted to recover single packet losses, as they are more frequent. However, it is also possible to recover losses of a relatively small number of consecutive packets. Recovering large bursts of packet loss is not feasible.

In principle there are two broad categories of FEC schemes: media-independent, and media-specific [33, 34], both of which will be discussed below.

3.2.1.1. Media Independent FEC

This scheme sends redundant data packets which are transmitted separately from the original packets. The redundant data is obtained by taking an exclusive OR (XOR) operation of the original packets [35] as illustrated in Figure 3.1.

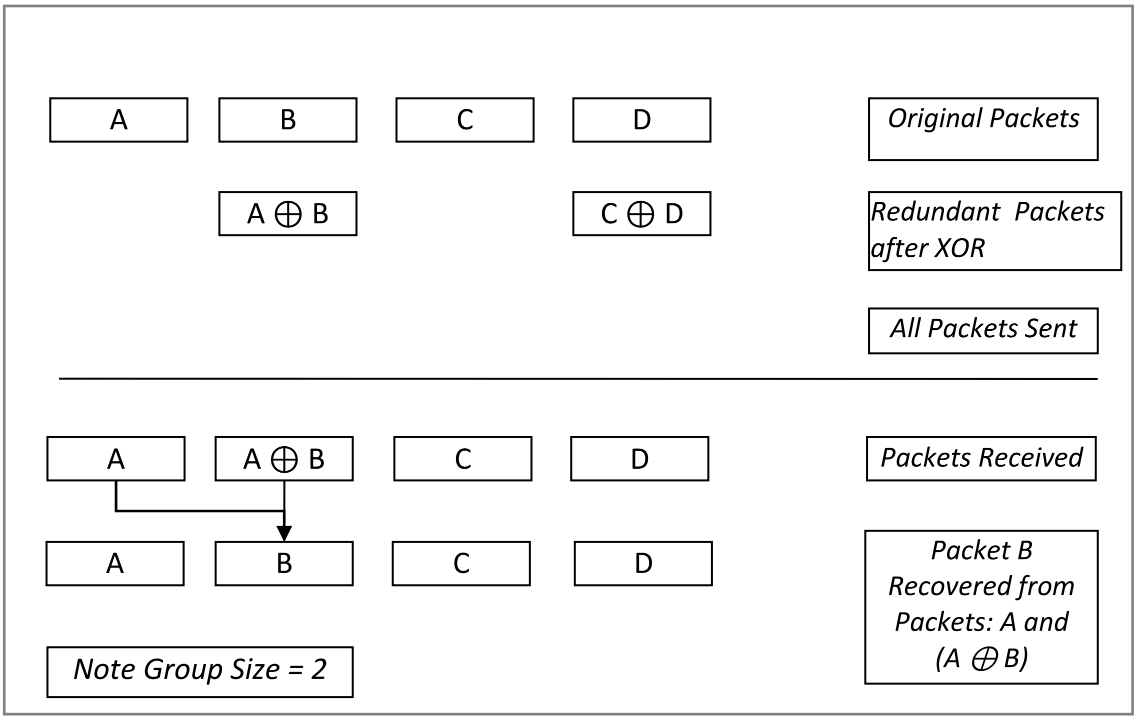


Figure 3.1, Media Independent FEC

Using this scheme if any one packet of the group is lost, the receiver can fully reconstruct the lost packet. For example, if packet B was lost in the network, it can be recovered in full using the redundant packet ($A \oplus B$) and packet A. This is illustrated in Figure 3.1. However, if two or more packets in a group are lost, the receiver cannot reconstruct the lost packets. By keeping the group size small a large fraction of the lost packets can be recovered (provided that the loss is not too excessive). However, a small group size will require a higher transmission rate. For example a group size of 2 will equate to 50% increase in packets to the original transmission bitrate of the stream.

$$\textbf{Increase factor in Transmission Rate} = \frac{1}{\text{Group Size}} \qquad \textit{Equation 3.1}$$

3.2.1.2. Media Specific FEC

The encoders of audio applications commonly encode an audio frame into two layers; a base and an enhancement, where the base layer is of low quality and enhancement layer of high quality (details can be found in section 3.3.2.1). Media Specific FEC appends the base layer of the previous packet, ('n-1'), to the current packet, n, as illustrated in Figure 3.2.

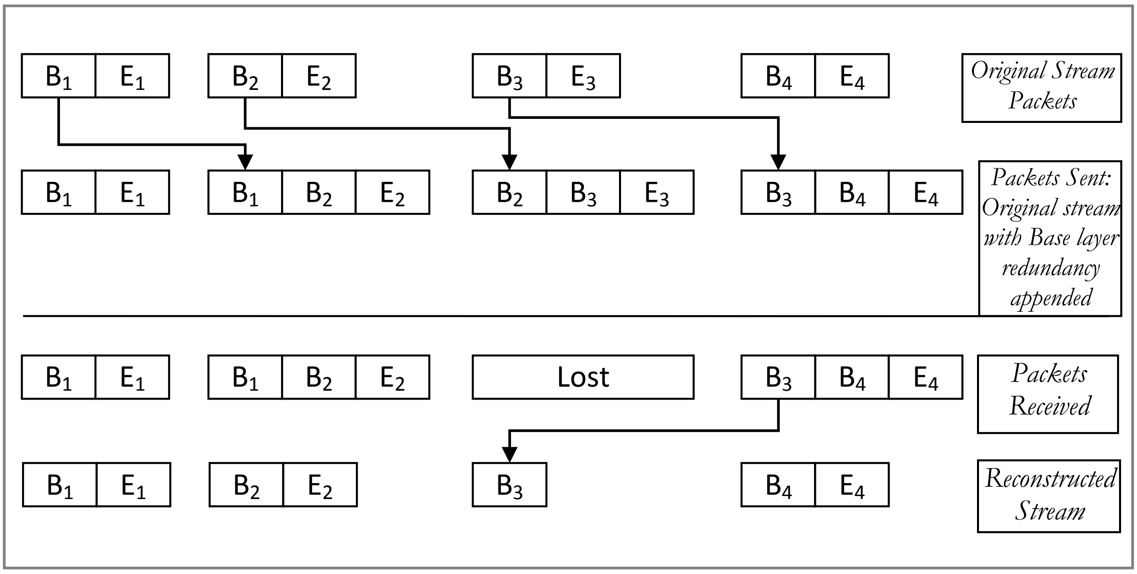


Figure 3.2, Media Specific FEC

The high quality bitrate of the stream is achieved by sending both parts of the audio frame: the base and enhancement layers. Whereas, a lower bitrate of the stream consists of only the base layer. Using the Media Specific FEC scheme the receiver can recover the lost packet by playing out the base layer quality. This maintains the intelligibility of the stream whereas a stream with missing packets might be meaningless.

The Media Specific FEC scheme can cope with some consecutive packet losses by appending more previous packets, base layers to the current packet. For example, rather than just appending one previous packet, ('n-1'), instead ('n-2'), and ('n-3') base layer redundancies frame data should be appended. With respect to data recovery the more the additions of redundancy data, the higher the degree of protection against increased numbers of packet losses. However, large amounts of redundancy will results in an increased transmission bitrate requiring more bandwidth on the network.

3.2.1.3. <u>Summary</u>

Comparatively speaking Media-independent FEC schemes have significant advantages over other schemes such as Media Specific. Protection against lost packets is relatively a simple process. Media-specific schemes require lower bandwidth as only base layer redundancy is sent, but at the cost of high complexity encodings. Additionally, when using either of two types of schemes; Media-independent and Medic Specific, the playout delay time increases as the receiver must wait to receive both the original packets and redundant packets before playout can begin.

Irrespective of the type of FEC scheme used, a common disadvantage is that they increase the bandwidth requirements. This may lead to increased network congestion, and therefore more packet losses, causing a worsening of the problem which FEC intended to solve. Therefore, multimedia application designers must take into consideration the level of FEC they intend to provide, as only a certain level of packet loss can be tolerated. [36, 20, 1] advices that a persistent loss of 10% makes FEC ineffective, and therefore, it is in the interest of multimedia applications that they respond to congestion indicators in order to avoid excessive development of loss. The suggested solution in this thesis is the adoption of an end-to-end TCP-friendly Congestion Control Mechanism.

3.2.2. Delay-Jitter

Jitter is caused by packets arriving at irregular intervals. In order to compensate for jitter the receiving application will have to buffer the arriving packets for some time before it passes them to the decoder to schedule playout (this buffer is also known as the playout, receiver or de-jitter buffer).

Delaying the packet for a sufficient time before they are decoded delivers a smooth and continuous playout of the multimedia data to the end-user. However, if the Playout Buffer (PB) delay is too large the interactivity of the multimedia stream will greatly suffer in terms of delay. On the other hand, if the PB is too small, the number of packets that arrive are 'too late' in time for playout, resulting in an increase in packet loss. These two scenarios are further illustrated as below.

Figure 3.3 shows a fixed playout buffer, of size 'Kv', where 'K' is a constant indicating the number of packets to buffer and 'v' is the time interval between sent packets (which also can be referred to as the time interval between frames). For illustration purposes, a value of 20 ms is used for 'v', and a value of 'K' is chosen between the range of 1 to 4. Table 3.2 illustrates the values of delay with respect to the size of the buffer.

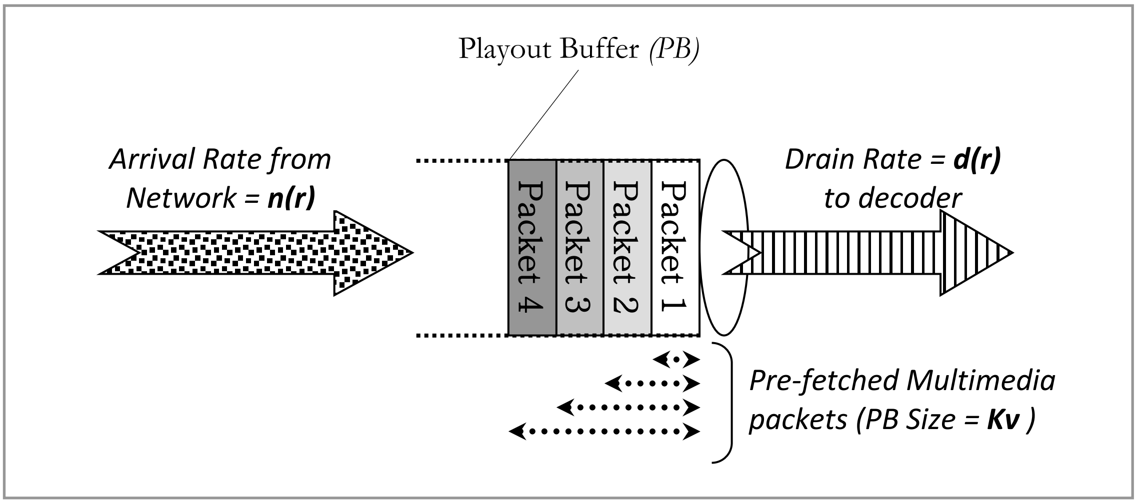
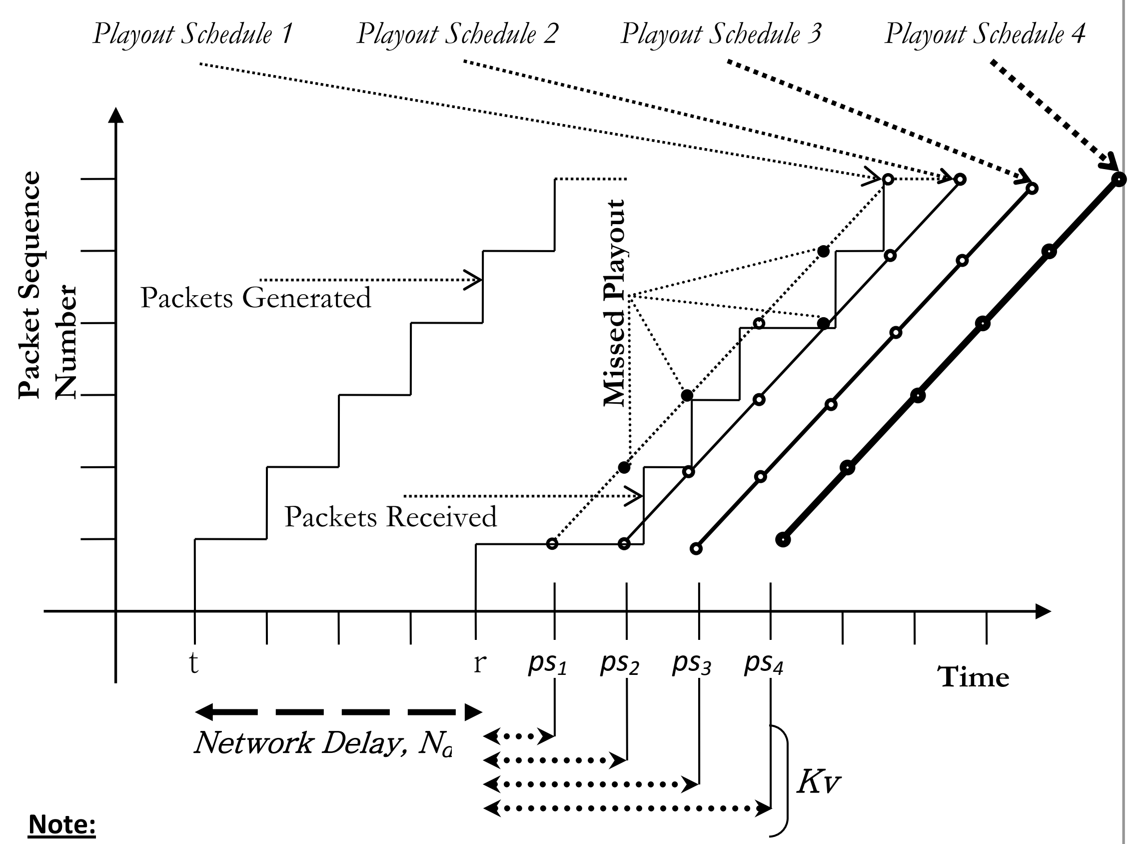


Figure 3.3, Playout Buffer being filled at Network rate 'n(t)' and drained at constant rate 'd'

Playout Scheme	K	v (ms)	PB Delay (ms)
1	1	20	20
2	2	20	40
3	3	20	60
4	4	20	80

Table 3.2, Playout Buffer Delay

Figure 3.4 illustrates the points in time when packets are generated and played out. Four Playout-Scheduled Schemes are considered based on the playout buffer size value, 'Kv'. The sender generates packets at a regular interval of 20 ms; the first packet arrives at time 'r'; the arrivals of subsequent packets are not evenly spaced because of network jitter. For Playout Schedule Scheme 1, where the playout buffer delay is of just 20 ms, packets 2, 3 and 5 do not arrive in time for playout and hence, the receiver considers them as lost packets, this is summarised in Table 3.3. For Playout Schedule Scheme 2, where the playout buffer delay is of 40 ms, only packet 4 does not arrive in time. However, for Playout Schedule Scheme 3 and 4, where the playout buffer delay is of 60 and 80 ms respectively, no packets arrive 'too-late', hence no packet loss is experienced by the receiver. Although the receiving application experiences no loss when using Playout Schedule Scheme 4 the PB imposes a longer delay before playout can begin. This could result in reduced interactivity, if the other components of delay such as in the sender buffer, or in the network, are too large.



Note:

ps = Playout Schedule

Figure 3.4, Playout delays, (adapted from [36] Figure 7.6)

Playout Scheme	K	v (msec)	PB Delay (msec)	Loss of too Late Packets
1	1	20	20	3
2	2	20	40	1
3	3	20	60	0
4	4	20	80	0

Table 3.3, Playout delays

To achieve the best trade-off between the playout buffer delay and the arrival of 'too late' packets the delay should be large enough to avoid excessive loss and small enough to maintain high interactivity. Referring to Figure 3.4 it is acceptable to say that playout schedule schemes 2 and 3 are more suitable than scheme 1, where a higher degree of loss is experienced owing to arrival of 'too late' packets, and scheme 4, results in larger delay. Furthermore, network jitter may vary over the duration of the connection. Hence using a fixed sized PB may not be adequate where a connection experiences variable network jitter. Therefore, the application designer may consider investing in adaptive playout buffers, details of which can be found in [37, 36]. The Playout Buffer can be configured to check the timestamps and sequence numbers of arrived packets before deciding to buffer packets or discarding them if they arrive 'too-late' in time sequence. Furthermore, with the help of sequence numbers on packets, packets can be ordered correctly before being placed into the buffer. (Sequencing numbering, timestamps can be provided when Real-time Transport Protocol (RTP) is used.)

3.3. <u>**Encoding Techniques available for Bitrate Adaptation**</u>

Network conditions vary over time and this reflects on the amount of bandwidth available. Therefore, the ability to adapt in terms of bitrate will be of benefit to the multimedia application so it can maintain its QoE. This will result in reducing the input load into the network. By adopting this approach the amount of congestion will reduce, hence, reflecting a lower network packet delay and loss, resulting in a better QoE for the multimedia application and better use of network resources. This section illustrates how multimedia applications adapt their bitrate in order to cope with varying network resources (bandwidth).

The capability for multimedia applications to adapt their bitrate depends on the performance of the encoder's scalability. The section below highlights three forms of adaptive encoding techniques: Layered, Coarse Grain and Fine Grain Scalable.

3.3.1. Common Audio / Voice Encoders

A number of audio and voice encoders (also referred as codecs) are currently used over the Internet, they range from: fixed rate codecs such as G.711, G729, to adaptive codecs such as MPEG-2. Table 3.4 summaries the details of the codecs mentioned, in this thesis.

	Encoding Type	**Application**	**Quality (MOS, 1 to 4.5)**	**Encoder Bitrate [2] (Kbits)**	**Voice sample Size [3] (ms)**	**Frame Rate [4] (pps)**	**Frame Size (Bytes)**
G.711	fixed	Voice	4.43	64	20	50	160
G.729	fixed low-bitrate	Voice	3.92	8	20	50	20
MPEG-2	Adaptive coarse-grain encoding	Voice & Audio	2.55 to 4.35	19.2 to 76.8 [1]	20	50	up to 168

Table 3.4, Voice Encoder Rates

[1] *can be higher if encoding at higher channels, e.g. stereo*

[2] *Encoder Bitrate, BR, is a product of Frame Size, F_S, and Frame Rate, F_R, i.e. $T_{EB} = F_R \times 8 \cdot F_S$*

[3] *Also referred to as Frame Interval, F_I*

[4] *Frame Rate =* $^1/_{F_I}$

G.711 is a widely accepted codec for voice telephony. It was standardised in 1972, and is able to encode speech at a high quality giving an MOS value of 4.3 out of 4.5. For more demanding networks G.729 codec can be used. This is able to compress speech down to 8 Kbits, however it compromises on quality, resulting in a MOS value of 3.92. Although this is acceptable toll quality, however the application using this codec is more vulnerable to loss of frames, particularly in congested networks. Losses greater than 10 percent can greatly affect perceived audio quality.

MPEG is a standard for audio and video which uses 'lossy' compression technique[2], and is able to achieve similar quality levels to 'lossless' encoding, in return producing a smaller Frame Size. Furthermore, MPEG can offer adaptive encoding allowing applications to choose different bitrate levels to operate at, depending on the quality demanded by the end user and/or the state of congestion in the network.

3.3.2. Adaptive Encoding

3.3.2.1. Layered Encoding

Layered coding found in MPEG-1 is the embedding of a multimedia signal to encode a frame into two sets of layers; base layer and enhancement layer, as shown in Figure 3.5 [38]. The base layer contains the most vital information and the enhancement layer contains the residual information to enhance base layer quality. In the presence of

[2] A technique where the resolution quality is reduced in images or higher frequency notes are removed in audio intelligently, in order to reduce the encoded Frame Size. The human interpretation by the eye, ear, and mind can easily fill in the missing blanks of the reduced quality. Therefore, the end-user is satisfied with the quality conveyed to it.

network congestion, only the base layer is sent and the enhancement layer is dropped. However, the enhancement layer is dependent on the base layer so if the base layer is lost during network transmission even though the enhancement layer is received, the frame cannot be reconstructed.

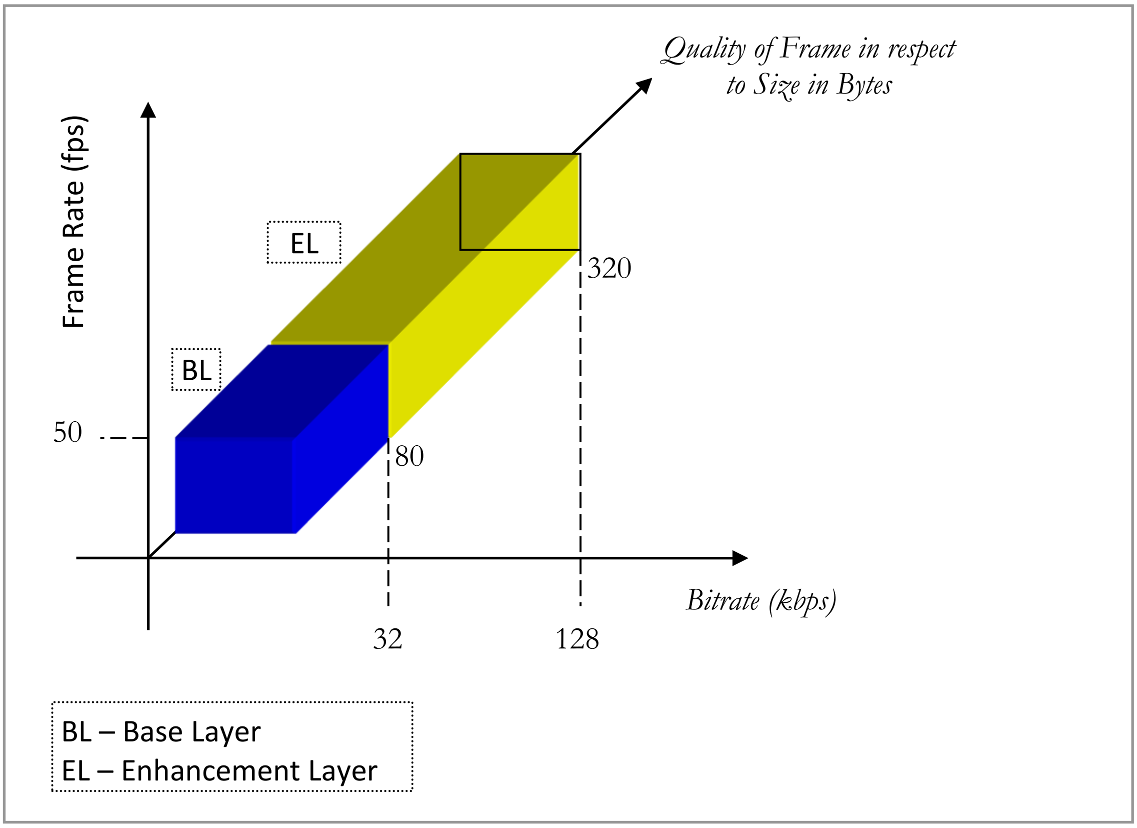


Figure 3.5, 'Scalable Layered Encoding' for Audio

Consider an audio application which is able to transmit at a bitrate of 32 Kbps representing base layer quality and at a maximum bitrate of 128 Kbps with the enhancement layer, as illustrated in Figure 3.5. During network congestion, where the network can only support a bitrate of 120 Kbps, and the multimedia stream can only adapt between two bitrates (32 and 128 Kbps), the multimedia stream will have to make do with a bitrate of 32 Kbps, because it cannot scale in between the two rates. The behaviour of the multimedia application bitrate is illustrated in Figure 3.6. By using the layered encoding approach the network resources are poorly used. This results in poor use of network resources and poor audio quality is perceived by the end user, who only receives base layer quality. Such encoding schemes prove to be very brittle in nature.

Additionally, if the available network bitrate falls below 32 Kbps the multimedia application cannot scale down its bitrate any further than the base layer bitrate. The application will experience packet losses when transmitting at a bitrate higher than the rate that the network can support.

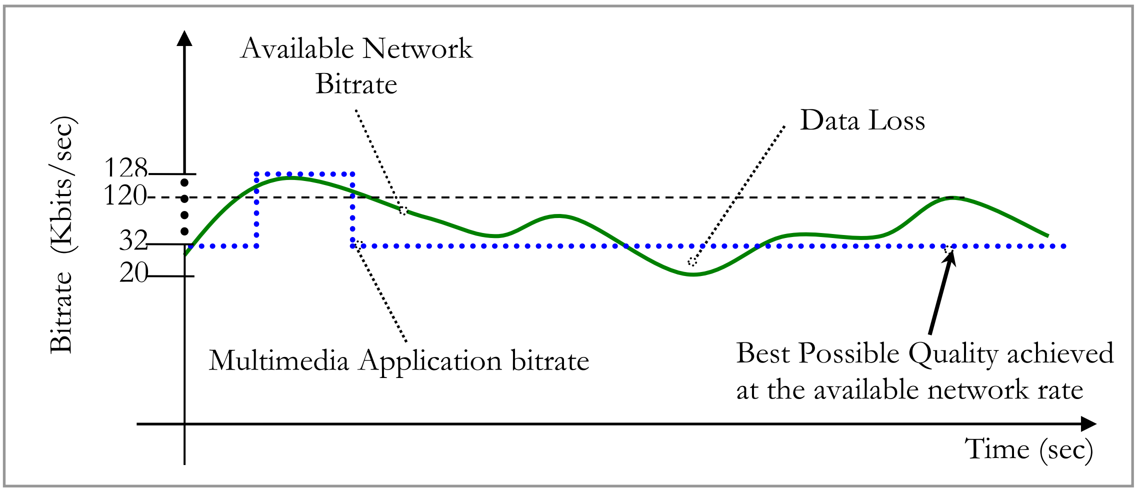


Figure 3.6, 'Varying Transmission rates' – Layered Encoding

3.3.2.2. Coarse Grain Encoding

The basic idea of Coarse Grain encoding is to encode the multimedia signal into a base layer and many enhancement layers (as found with MPEG-2 codec) [4]. However, the enhancement layers are sub-divided into a set of levels representing different quality levels, as illustrated in Figure 3.7. However, the base layer cannot be refined in this way. It is essentially the lowest level of the bitrate that can be provided). So during network congestion, the base layer is sent with as many enhancement layers as possible, to match the available network resources. Although the encoder can produce as many enhancement layers as are desired, each enhancement layer is dependent on the previous enhancement layer. For example, the second enhancement layer can only be decoded when the first enhancement layer is available, not without it. Furthermore, an enhancement layer cannot be recovered unless the base layer is available.

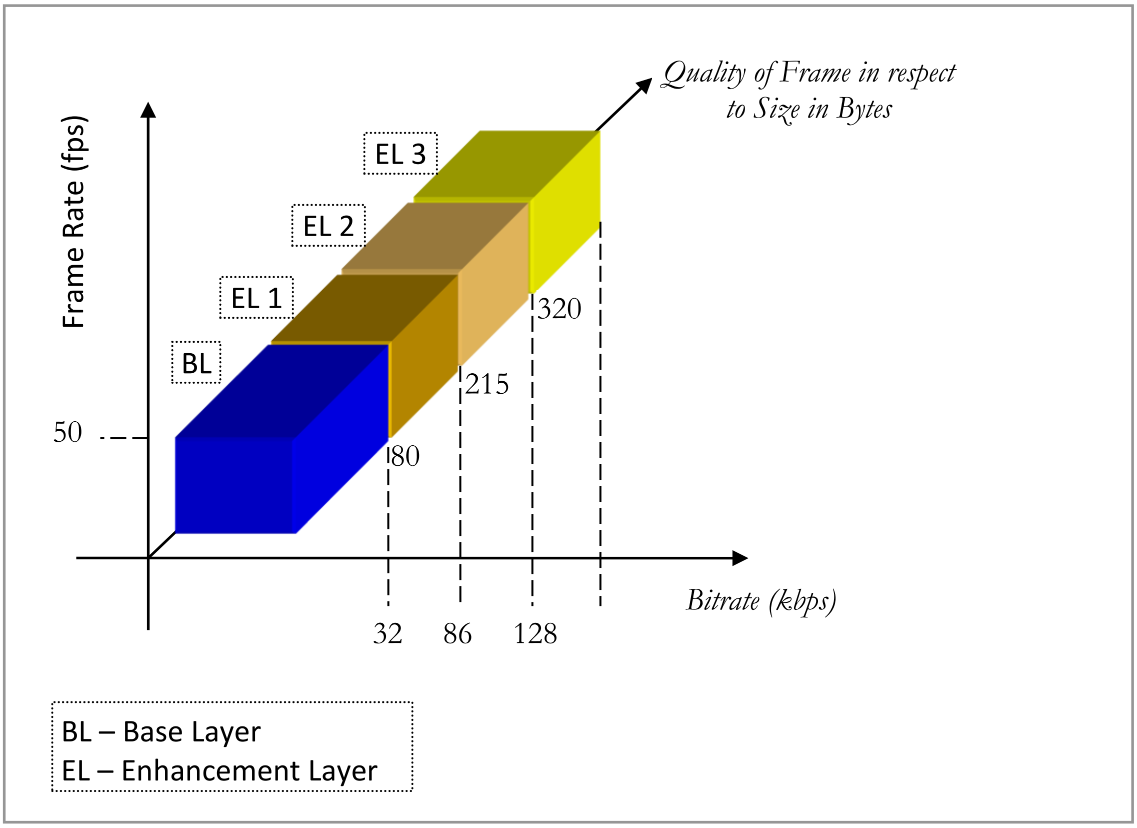


Figure 3.7, 'Coarse-Grain Scalable Encoding' for Audio

Consider an audio signal where the basic layer has been encoded at a minimum rate of 32 Kbps and the first enhancement layer at 54 Kbps, second at 42 Kbps and so on, as shown in Figure 3.7. The possible sending rate options are 32, 86 and 128 Kbps. As the available bandwidth fluctuates along the end-to-end path between the receiver and sender, the sender can potentially send at three different bitrates. Assume the network can only support a bitrate of 60 kbps. Then, the user would have to make do with the 32 Kbps signal, as the application would not be able to scale between the base and the first enhanced layer (32 and 86 Kbps respectively), this is illustrated in Figure 1.1.

3.3.2.3. Fine Grain Scalable Encoding

Fine Grain Scalable (FGS) encoders are able to achieve full scalability by organising the data in the frame in order of priority, from Most Significant Bytes (MSB) to Least Significant Bytes (LSB). The MSB represent the most basic but vital information (i.e. minimum quality), scaling up to the LSB which represent the enhancement of the basic information, (i.e. higher quality). In MPEG-4 FGS the basic idea is to code an audio signal into a base layer and an enhancement layer [39], where the base layer is non-scalable but the enhancement layer is fine-grain scalable. This makes it possible for the enhancement layer to be truncated to any size (as illustrated in Figure 3.8). The decoder is then able to reconstruct the multimedia stream from the base layer and the truncated enhancement layer without any complications. Note that the perceived quality of the multimedia will be proportional to the size of the truncated enhancement layer (i.e. the higher the truncation the lower the quality) [40].

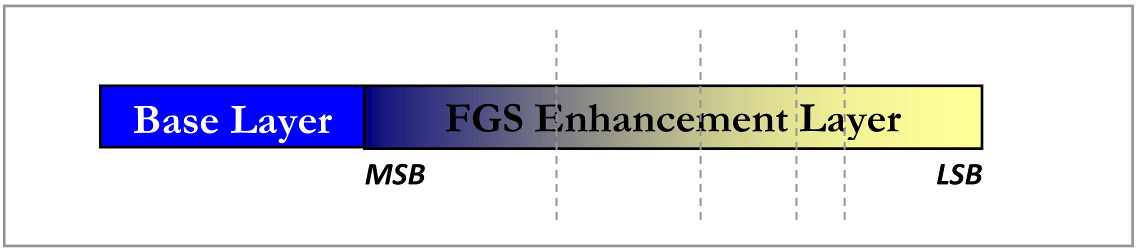


Figure 3.8, FGS coding in MPEG-4

Recent research has been carried out to make the whole multimedia bitstream fully scalable [41, 42, 43], meaning that the encoder generates frames which can be truncated almost anywhere, as shown in Figure 3.9. The concept of generating a Layered bitstream, where an enhancement layer is dependent on the base layer in the case of Coarse Grain and Layered encoding is now avoided. A full Fine Grain Scalable (FGS) bitstream is made possible to offer 'byte-level' scalability where the frame can be truncated to any length without any boundary restrictions.

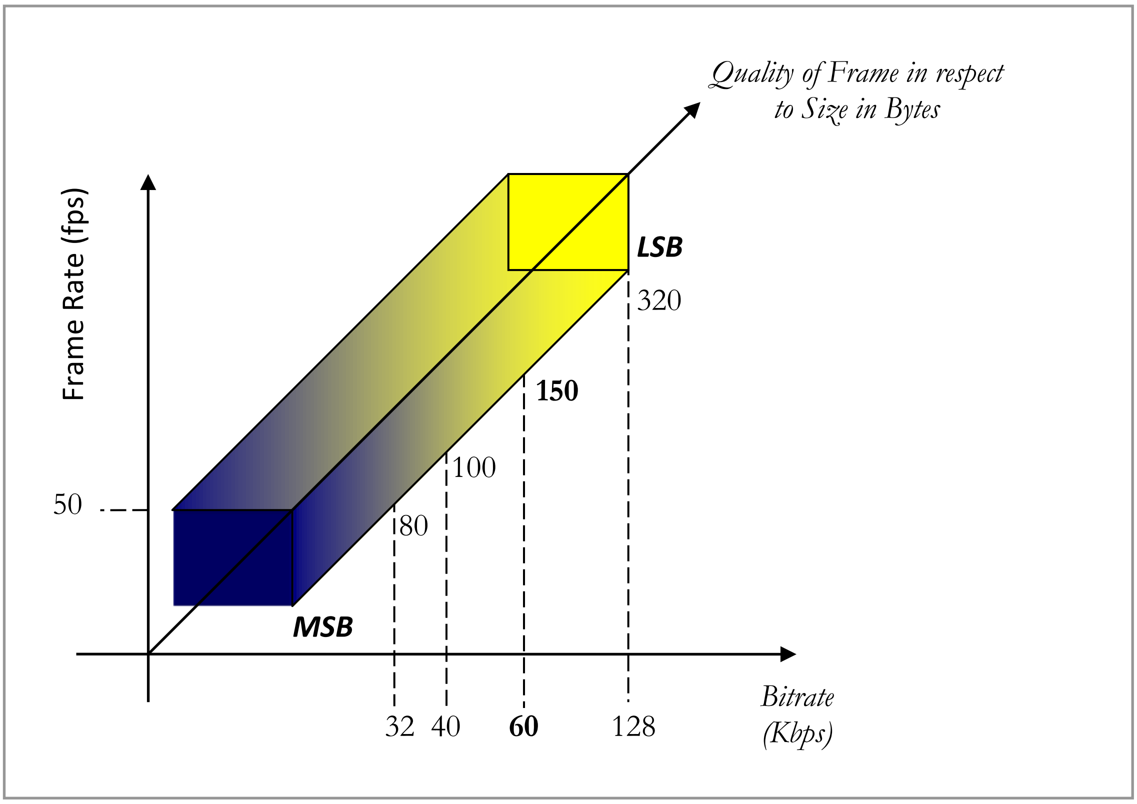


Figure 3.9, 'Scalable Encoding' – FGS

The continuous bitrate scalability offered by FGS bitstreams can be greatly beneficial to multimedia applications operating over networks with fluctuating bandwidths, such as the Internet. FGS decoders are able to fully decode the necessary information from a received truncated FGS bitstream.

Considering the same network congestion scenario used in the Coarse Grain example, where the network can only support a bitrate of 60 Kbps. The FGS bitstream will be able to truncate the frame to a size of 150 bytes to match the available network bitrate of 60 Kbps, as illustrated in Figure 1.2. Hence, the multimedia application can now provide a true match to the available network bitrate, providing the best possible quality to the end user whilst making best use of the available network resources. This encoding scheme is of great benefit in an environment such as the Internet, where available network bitrate is continually changing and the ability to adapt in a manner without causing step changes in quality is of significant benefit to the end user.

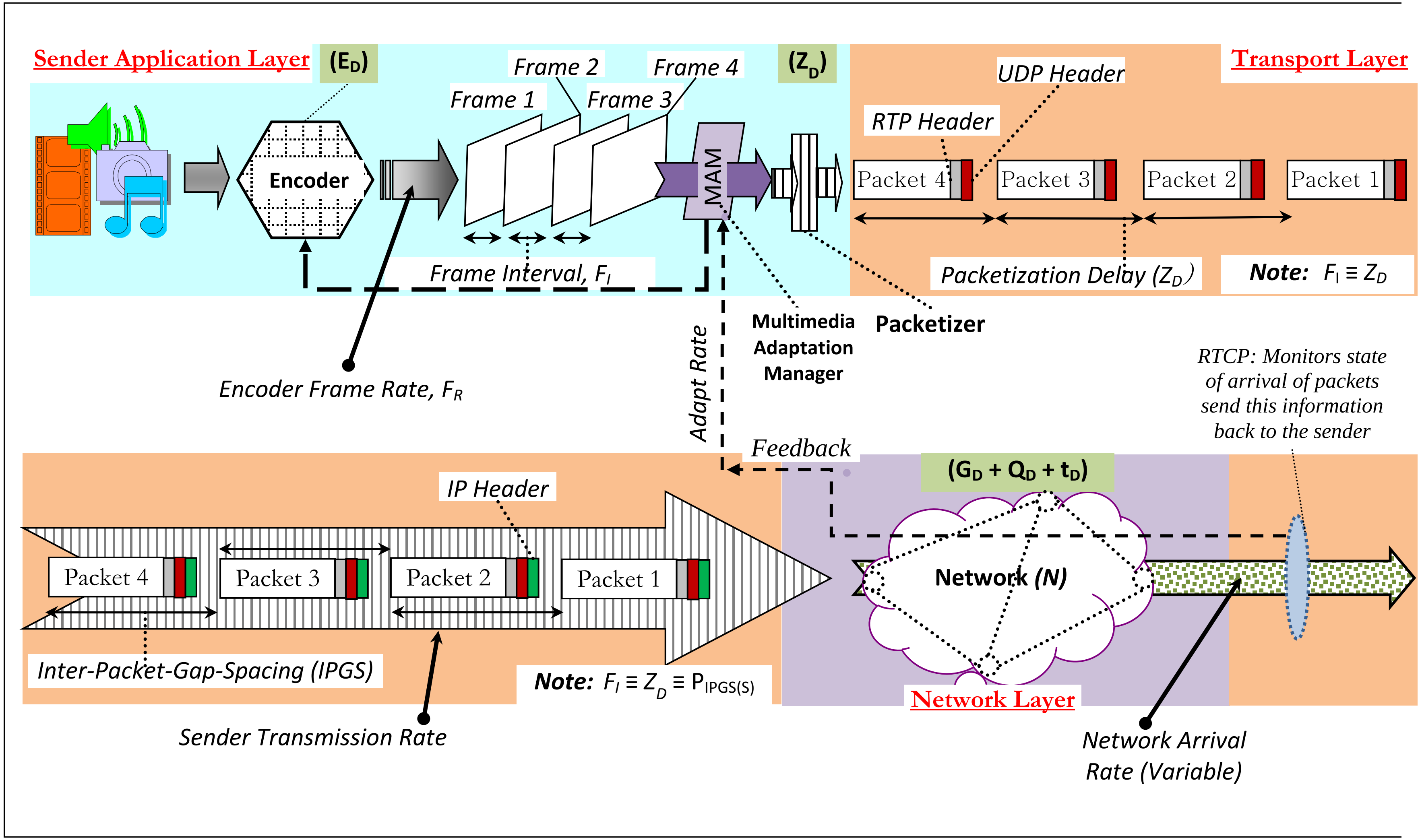
FGS bitstreams are encoded at their full bitrate; the encoded bitstream allows adaptation to take place after encoding, hence allowing on-the-fly adaptation of the bitstream without requiring the storage of multiple copies of the bitstream at different bitrates. This lends itself very reasonably to Congestion Control Mechanisms such as TCP-Friendly; section 4.2 illustrates in detail how the use of FGS bitstreams will enable the Congestion Control Mechanism to maintain an isochronous service whilst responding to network congestion.

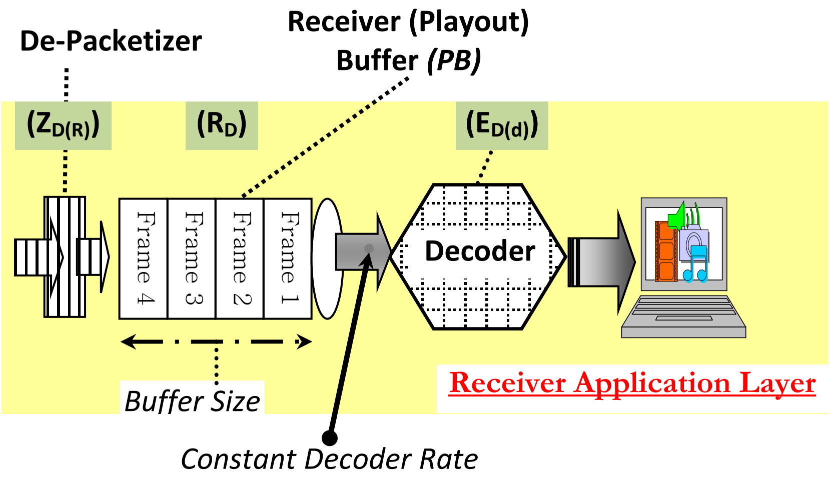
3.4. Multimedia Communication Architecture

This section will highlight the components involved in transmitting multimedia data from end-to-end. Figure 3.10 illustrates the overall picture of the multimedia communication which will be used to explain the remaining part of this section. Firstly, at the sender side of the application, the raw multimedia signal is digitized using the encoder which generates data (known as frames) at fixed Frame Intervals, F_I, of fixed Frame Size, F_S, (which can be adaptive depending on the encoder used).

The multimedia application bitrate, more specifically known as the Encoder Bitrate, T_{EB}, is a product of the Frame Rate, F_R, and Frame Size, F_S, where the Frame Rate is the inverse of the Frame Interval ($F_R = {}^{1}/_{F_I}$).

$$\underset{(bps)}{T_{EB}} = \underset{(fps)}{F_R} \times \underset{(8\cdot bytes)}{(8 \times F_S)} \qquad \textit{Equation 3.2}$$


Sender Application Layer
(ED)
Frame 2
Frame 4
(ZD)
Transport Layer
UDP Header
Frame 1
Frame 3
RTP Header
Encoder
MAM
Packet 4
Packet 3
Packet 2
Packet 1
Frame Interval, FI
Packetization Delay (ZD)
Note: FI ≡ ZD
Multimedia
Adaptation
Manager
Packetizer
Encoder Frame Rate, FR
Adapt Rate
RTCP: Monitors state
of arrival of packets
send this information
back to the sender
Feedback
IP Header
(GD + QD + tD)
Packet 4
Packet 3
Packet 2
Packet 1
Network (N)
Inter-Packet-Gap-Spacing (IPGS)
Note: FI ≡ ZD ≡ PIPGS(S)
Network Layer
Sender Transmission Rate
Network Arrival
Rate (Variable)



Key:

'E_D' *encoding delay*
'Z_D' *packetization delay*
'S_D' *sender buffer delay*
'G_D' *propagation delay*
'Q_D' *queuing delay*
't_D' *service time delay*
'R_D' *playout buffer delay*
'$Z_{D(R)}$' *de-packetization delay*
'$E_{D(D)}$' *decoding delay*

Figure 3.10, Multimedia System: for transporting multimedia streams over the Internet

3.4.1. Packetization

The frames are added with RTP headers, which add sequence numbers and timestamps to frames. This helps the receiving application to differentiate between arriving packets, i.e. to resolve out-of-order delivery of packets that arises due to network delay, jitter and loss [8, 27].

Furthermore, the frames are also added with the transport layer headers. Here UDP is used. Following this, IP headers are added (which are of the network layer), and finally the packet is sent through the Internet [44, 45, 46].

This packetization process adds 40 bytes (20 Bytes for IP, 12 bytes for UDP and 8 bytes for RTP) to the original frame. This process is illustrated in Figure 3.11.

$$\mathbf{P_S\ (Bytes) = F_S + 40} \qquad \textit{Equation 3.3}$$

And results in a transmission bitrate, T_{BR}, of:

$$\underset{\text{(bps)}}{\mathbf{T_{BR}}} = \underset{\text{(fps)}}{\mathbf{F_R}} \times \underset{\text{8 x (bytes)}}{\mathbf{8 \times (F_S + 40)}} \qquad \textit{Equation 3.4}$$

3.4.2. Fragmentation

It is important to take into account that the Packet Size is no greater than the Maximum Transmission Unit (MTU) size, a constraint of the link layer. In the case of the Ethernet link, it is a maximum Packet Size up to 1500 Bytes including all headers [47] (however this may vary across different networks). An audio packet is normally a fraction of the MTU size (160 Bytes of Frame size plus 40 Bytes for headers, totalling 200 Bytes), therefore it is safe to assume that an audio frame will not undergo fragmentation. In such a case the Frame Rate, F_R, is equivalent to the Packet Rate, P_R, hence the effective transmission rate, T_{ETR}, can be defined as:

$$\underset{(bps)}{\mathbf{T_{ETR}}} = \underset{(pps)}{\mathbf{P_R}} \times \underset{(bits)}{\mathbf{(8 \times P_S)}} \qquad \textit{Equation 3.5}$$

Where:

P_S (Bytes) = F_S + 40

On the other hand, if the multimedia packet is larger than the MTU (for example in the case of a video frame) the frame will undergo fragmentation across several packets. This will result in an increased Packet Rate and also an increased transmission rate. The way in which the fragmentation is carried out, i.e. splitting the original multimedia packet across multiple packets and adding necessary identifiers, is explained in [48]. However, the impact on the increase in packet rate and the transmission rate is shown in the equations below.

$$\boldsymbol{n}_{\boldsymbol{packets}} = \left\lceil \frac{\textbf{Multimedia Packet Size}}{\textbf{MTU}} \right\rceil \qquad \textit{Equation 3.6}$$

Maximum Sender Transmission Bitrate =

$$(\boldsymbol{n}_{\boldsymbol{packets}} \times \mathbf{F_R}) \times (\mathbf{8 \times (P_S + Fragment\ Header\ Size)}) \qquad \textit{Equation 3.7}$$

Note the last packets' payload of the fragment may not be fully occupied therefore the transmission bitrate calculated in Equation 3.7 indicates the maximum transmission bitrate; hence it is possible that the effective transmission bitrate may be smaller.

3.4.3. Frame Grouping

Some Applications may prefer to send multiple frames in the payload of a single packet, in order to conserve bandwidth by reducing the overhead required to send individual frames with 40 bytes of headers each. With multiple frames in one packet means a number of frames can be sent with only one set of 40 byte header, (bearing in mind that the total packet size does not exceed the MTU size).

The process of adding multiple frames is shown in Figure 3.11 and the impact on conserving bandwidth is illustrated in Table 3.5.

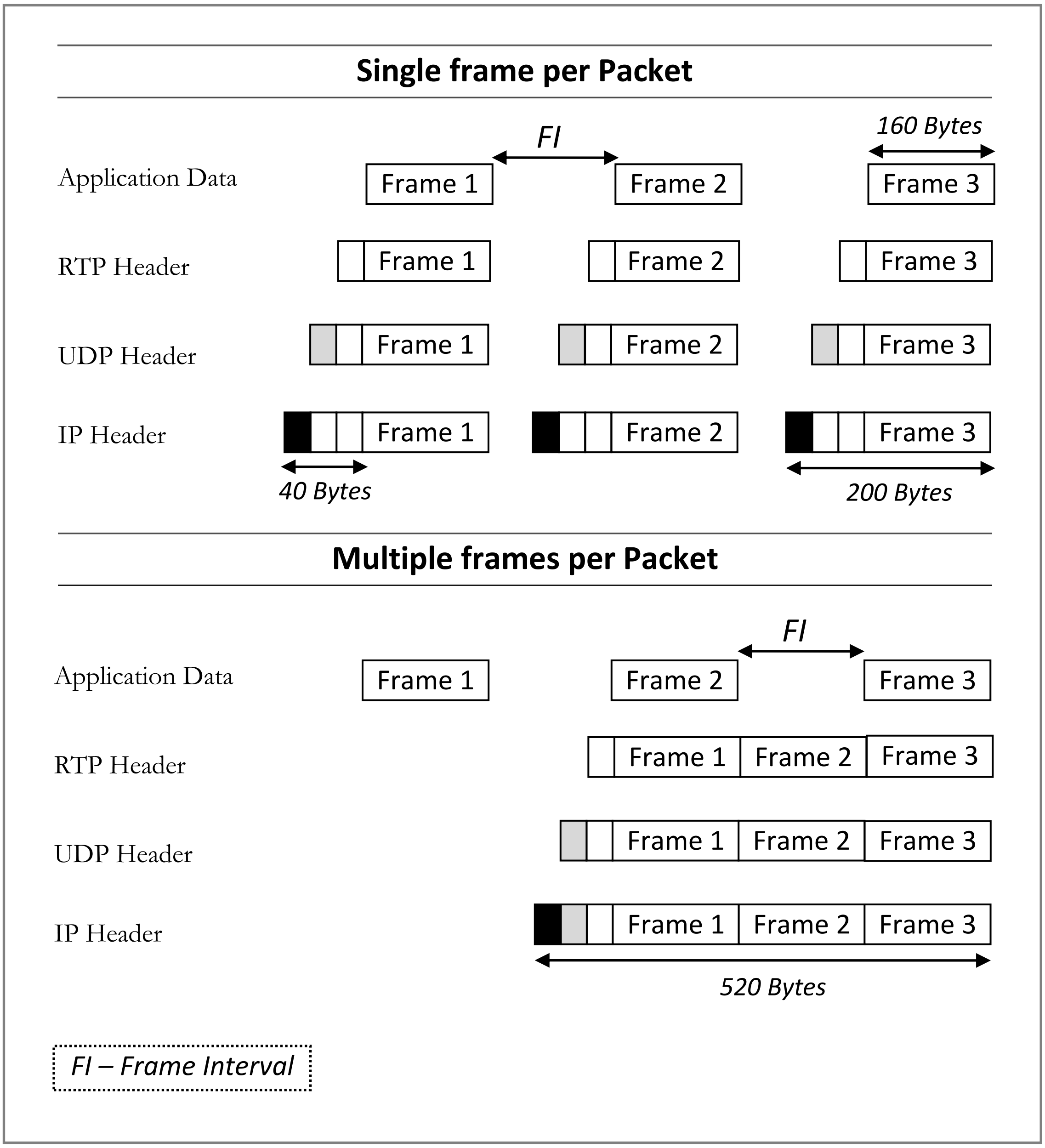


Figure 3.11, Packetization

G.711 Voice Frames per Packet	IP/UDP/RTP Headers (Bytes)	Packet Payload Size (Bytes)	Packet Size (Bytes)	Packet Rate (pps)	Bandwidth Consumed (Kbps)	Packetization Delay (ms)
1	40	160	200	50	80,000	20
2	40	320	360	25	72,000	40
3	40	480	520	16.7	69,333	60
4	40	640	680	12.5	68,000	80

Table 3.5, Packetization Delay

Although this may seem beneficial in terms of conserving bandwidth, this does add on delay which is a product of the number of frames sent, F_N, and the Frame Interval, F_I. This delay is referred to as packetization delay, Z_D.

$$\mathbf{Z_D = F_N \times F_I} \qquad \textit{Equation 3.8}$$

Furthermore, carrying too many multiple frames by a single packet increases the rate of quality degradation. One single packet loss will mean the loss of multiple frames, (referred to as a bursty loss), and this significantly degrades the received quality. Illustrating this in a quality perspective, a consecutive loss of 2 packets can lead to a quality impairment of ΔR=-38, 3 packets ΔR=-57 and 4 packets R=-66 [49]. For example a consecutive packet loss equal to 3 packets degrades the overall quality to R=43 (R=100-57=43), when the minimum acceptable quality is R=60. Therefore, the impact of consecutive loss is severe, particularly when it consists of 3 packets and more.

3.4.4. Transmission

After packetization the multimedia packets are sent in the network, and they experience various links (routers) to reach their end destination. Successive packets may experience different delay and probability of packet loss when they are transported along different routing paths. The application at the sender side can monitor this, for example, through using the information provided in the RTCP report sent by the receiver, and the relevant statistics of loss, delay and jitter can be obtained [50].

The sender application normally employs a multimedia adaptation manager (MAM) which is responsible for monitoring the network condition and will take necessary steps to respond to changing network conditions. For example, if the application perceives high loss rates, it implies that the application is sending at a higher bitrate than that which the network can support. Hence, the MAM may well decide to request the encoder to reduce its bitrate until conditions improve in the network. Furthermore, a high loss rate may well result in a higher degree of FEC, in order to compensate for packet loss.

3.4.5. De-jitter (Playout) Buffer

However, at the receiving end the packets may have experienced a variable arrival rate due to network congestion. Therefore, packets will be buffered for some time before they are scheduled for playout.

The successfully arrived packets (meaning packets that have not arrived 'too-late' in time) are ordered correctly with the help of sequence numbers and timestamps using the information found in RTP headers (otherwise 'too-late' packets are discarded'). These packets are then de-packetized and are placed into the Playout Buffer (PB). The PB drains at a constant rate which feeds the decoder. The decoded frames are played out to the end user to be heard or viewed visually.

3.5. Summary

This chapter illustrated some mechanisms that can be used at the end-system to deal with:

a) Loss – using Forward Error Correction techniques such as Media-independent or Media Specific that generate redundancy data, which can be used to recover lost packets.
b) Delay-jitter – using a Playout Buffer which temporarily stores multimedia frames so it can provide a constant rate to the decoder.
c) Input load into the network – by using adaptive encoding the multimedia application can reduce its bitrate in the presence of network congestion. However, a reduced bitrate means a reduced multimedia quality for the end-user. Nevertheless, the multimedia application will be able to maintain its interactivity and intelligibility by attempting to minimise loss and delay of packets in the network.

This chapter also goes into detail about the various components involved in the end-to-end transmission of multimedia data across the IP network:

a) Packetization process where necessary packet headers are added.
b) Fragmentation if the Packet Size is greater than the MTU size.
c) Frame grouping where a number of frames are sent together in the payload of one packet in order to reduce overhead.

4. IP Multimedia Adaptation: from Network Friendly to Media Friendly

The problem dealt with in this chapter arises from the fact that although the 'TCP Friendly Rate Control' (TFRC) Congestion Control Mechanism (CCM) was introduced to support Real-Time applications on the Internet, its main focus was placed on achieving TCP fairness. Little attention had been paid to the applications using them, and in particular, to the Quality of Experience (QoE) perceived by their users. This chapter aims to analyse the problems of transmitting multimedia data over the TFRC Congestion Control Mechanism integrated with its Multimedia Adaptation Architecture (MAA). In this thesis, this is referred to as TFRC MAA. The remaining part of this chapter proposes a novel MAA referred to as 'TCP friendly rate control – Fine Grain Scalable' (TFGS) integrated with its novel CCM.

With rate-based Congestion Control Mechanisms, such as TFRC, the response to bitrate variation (i.e. congestion) can be interpreted as a function of either the Packet Size or Packet Rate. With the TFRC Multimedia Adaptation Architecture (MAA) the Congestion Control Mechanism interprets the bitrate by adjusting the Packet Rate and maintains a fixed Packet Size. The analysis and simulations reported in chapter 6 and 7 respectively conclude that responding to congestion in such a manner has a severe effect on the end-to-end delivery of multimedia data. The prime reason for Quality of Experience (QoE) degradation is the mismatch of the Packet Rate of the CCM and the Frame Rate of the multimedia encoder. As soon as the CCM operates at a Packet Rate lower than the Frame Rate, the problem arises: buffering of frames at the sender and loss if the buffer becomes full. Otherwise when the Packet Rate is equal to, or higher, than the Frame Rate the Multimedia Adaptation Architecture performs at its desired level.

Due to the mismatch between the Packet Rate and Frame Rate, a novel Congestion Control Mechanism (CCM) is developed which integrates with a new Multimedia Adaptation Architecture (MAA). This is referred to here as, 'TCP friendly rate control – Fine Grain Scalable' (TFGS). With TFGS the Congestion Control Mechanism responds to congestion by adjusting the Packet Size while maintaining a fixed Packet Rate. The outcome is that the Packet Rate (of the CCM) is equivalent to the Frame Rate (of the multimedia encoder), i.e. packets are scheduled as soon as they are generated; eliminating waiting delay of the packets at the sender side and of loss when the buffer becomes full. This approach provides an isochronous service which is of crucial benefit to Interactive Real-Time services.

TFGS CCM responds to congestion by adapting the Packet Size and this functionality is referred to as Packet Size Truncation (PST). It incorporates a Multimedia Adaptation Manager (MAM) which truncates the encoded multimedia frame to the size indicated by the Packet Size Truncation function of the CCM. The instant on the fly truncation is possible because FGS encoded scheme is used where data is organised in terms of significance of importance, Most to Least Significant Bytes (MSB to LSB). Additionally, the truncation can be carried out at byte-level granularity. Hence, this provides a true match of application demand to network supply.

Using the TFGS MAA the quality of the multimedia frame may be compromised but the end-to-end interactivity is maintained. The 'Multimedia Adaptation Architecture' (MAA) of TFGS is able to integrate the four main components of a multimedia system: (*1*) Application, (*2*) Multimedia Encoder, (*3*) Multimedia Adaptation Manager (MAM) and (*4*) Congestion Control Mechanism (CCM). This integration provides the capability for true on-the-fly adaptation of the multimedia stream, which enables it to meet the interactive QoS requirements, along with achieving fairness amongst competing flows.

This on-the-fly adaptation is achieved by the multimedia encoder passing the Frame Size and Frame Rate parameters to the CCM. The CCM monitors the available network bitrate, and interprets the response to congestion by requesting the Multimedia Adaptation Manager (MAM) to reduce the frame size. Taking benefit of FGS encoded frames, the frame can be truncated to the size calculated by the PST function of the CCM. Once the frame is truncated the frame (along with its headers, together referred to as a packet) is sent into the network.

This behaviour of TFGS is in contrast to the TFRC MAA which treats each component as a separate entity, i.e. the Congestion Control Mechanism has no idea how the encoder responds to congestion, whether by Frame Rate or Frame Size. The equivalent Multimedia Adaptation Manager (MAM) in the TFRC adaptation architecture is the Sender Buffer (SB) which acts as a temporary buffer between the encoder rate, known as the Frame Rate, and the Packet Rate of the CCM. The application has no knowledge of the components involved in responding to congestion, it operates under the framework of ‘send and forget’.

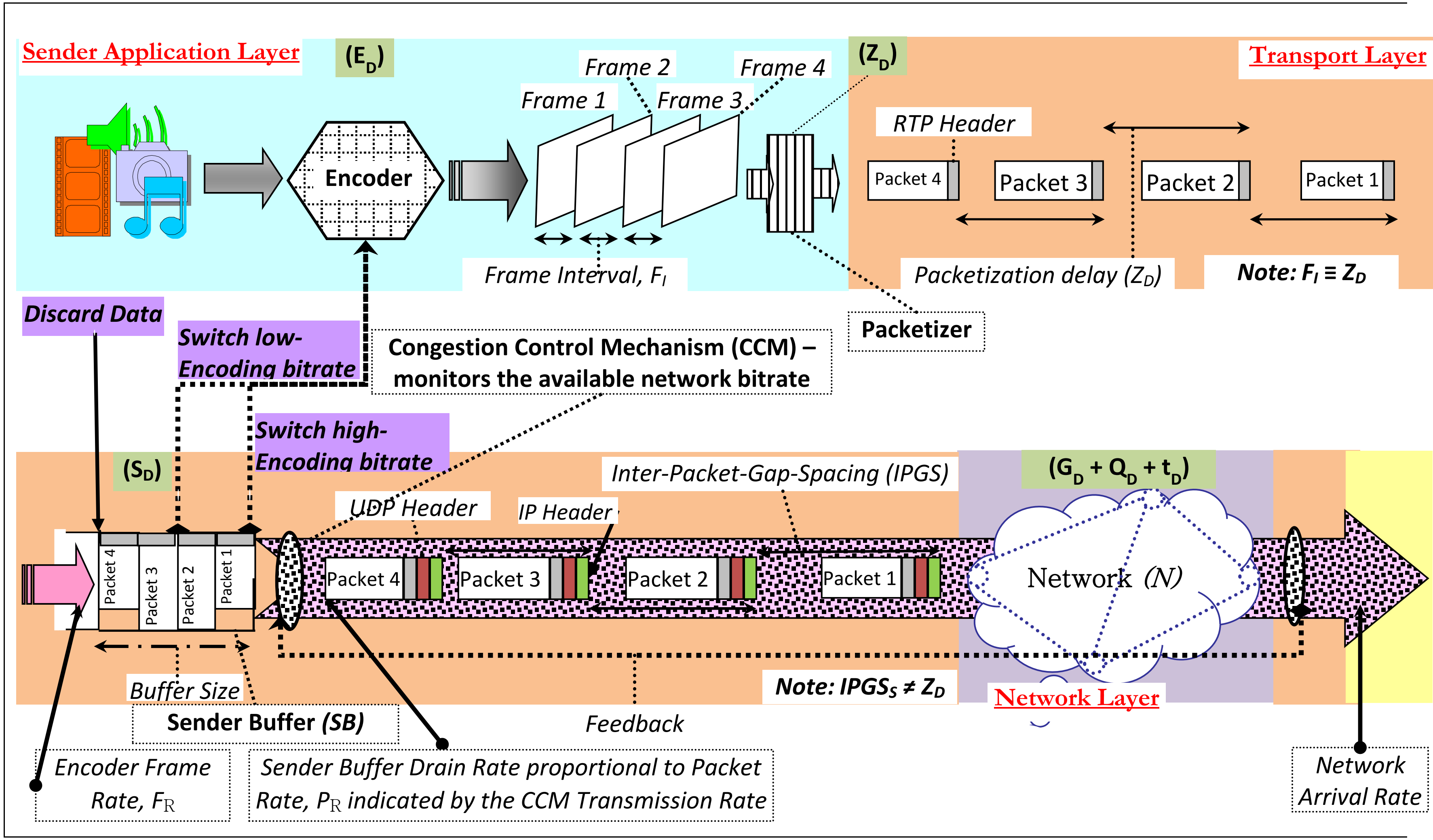
The significance of sending multimedia frames as soon as they are generated will be quantified in chapter 6 and 7. The chapters will represent the method of congestion response in the form of a Quality of Experience (QoE) measure. For example a) how much of a Packet Rate reduction can still maintain a minimum quality level of a voice call, b) how many more voice flows using either TFRC or TFGS MAA can operate under the same network conditions, whilst sustaining their minimum quality level[3].

[3] The measure of minimum quality is defined in chapter 6 using an E-model, which expresses packet loss, delay and byte loss, on a measure of scale 0 to 100 in units of R-value.

4.1. TCP Friendly Rate Control (TFRC) Multimedia Adaptation Architecture

The Multimedia Adaptation Architecture recommended for applications using 'TCP Friendly Rate Control' (TFRC) is based on an IETF document known as "Strategies for Streaming Media Applications using TCP-Friendly Rate Control" [51]. The details of this architecture are highlighted in the section "two-way live media" in [51].

This adaptation architecture integrates three components of a multimedia system (*1*) Encoder, (*2*) Sender Buffer and (*3*) Congestion Control Mechanism (CCM), whilst ignoring the Multimedia Application. As the application operates under the framework of 'send and forget' it is the CCM's responsibility to satisfy the application's demands as best it can. The above three components reside at the sender side of the multimedia connection. The positions of each of these components are shown in Figure 4.1 and the interactions between each of them are shown in Figure 4.2.


Sender Application Layer
(E_D)
Encoder
Frame 1
Frame 2
Frame 3
Frame 4
(Z_D)
Transport Layer
RTP Header
Packet 4
Packet 3
Packet 2
Packet 1
Frame Interval, F_I
Packetization delay (Z_D)
Note: F_I ≡ Z_D
Packetizer
Discard Data
Switch low-Encoding bitrate
Congestion Control Mechanism (CCM) – monitors the available network bitrate
Switch high-Encoding bitrate
(S_D)
Inter-Packet-Gap-Spacing (IPGS)
(G_D + Q_D + t_D)
UDP Header
IP Header
Packet 4
Packet 3
Packet 2
Packet 1
Network (N)
Buffer Size
Note: IPGS_S ≠ Z_D
Network Layer
Sender Buffer (SB)
Feedback
Encoder Frame Rate, F_R
Sender Buffer Drain Rate proportional to Packet Rate, P_R indicated by the CCM Transmission Rate
Network Arrival Rate

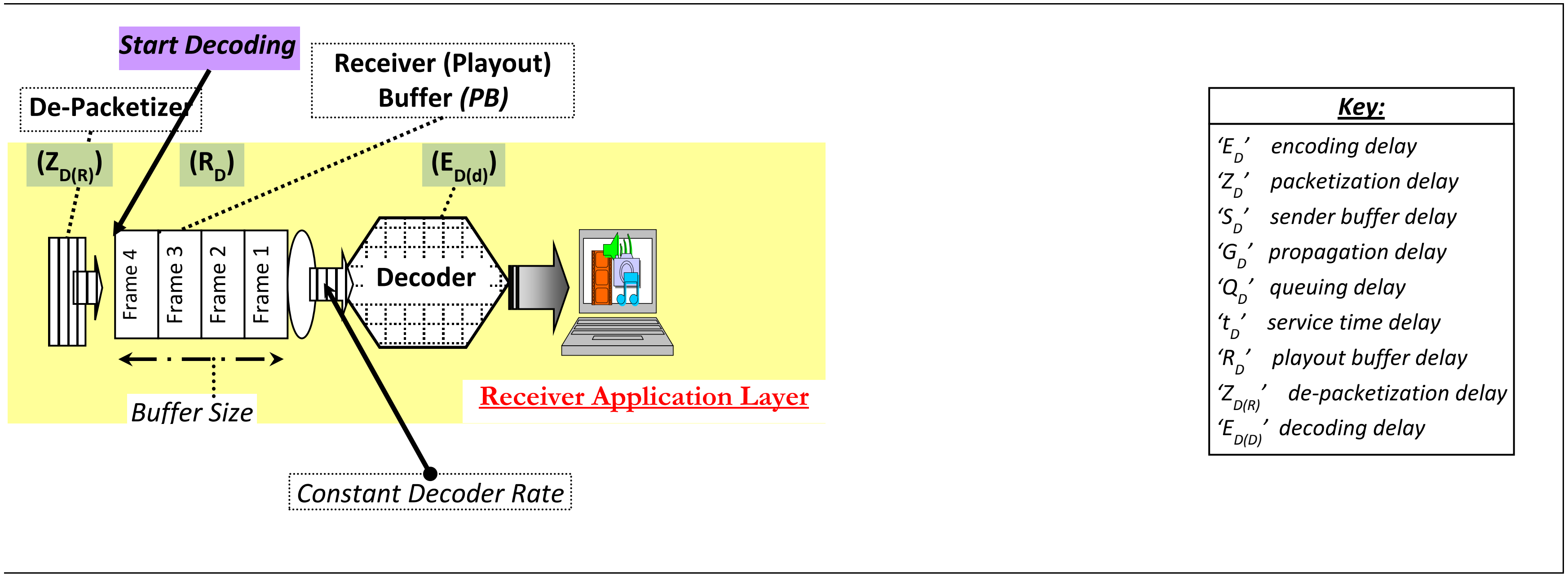


Figure 4.1, Multimedia System: for transporting multimedia over TFRC Congestion Control Mechanism

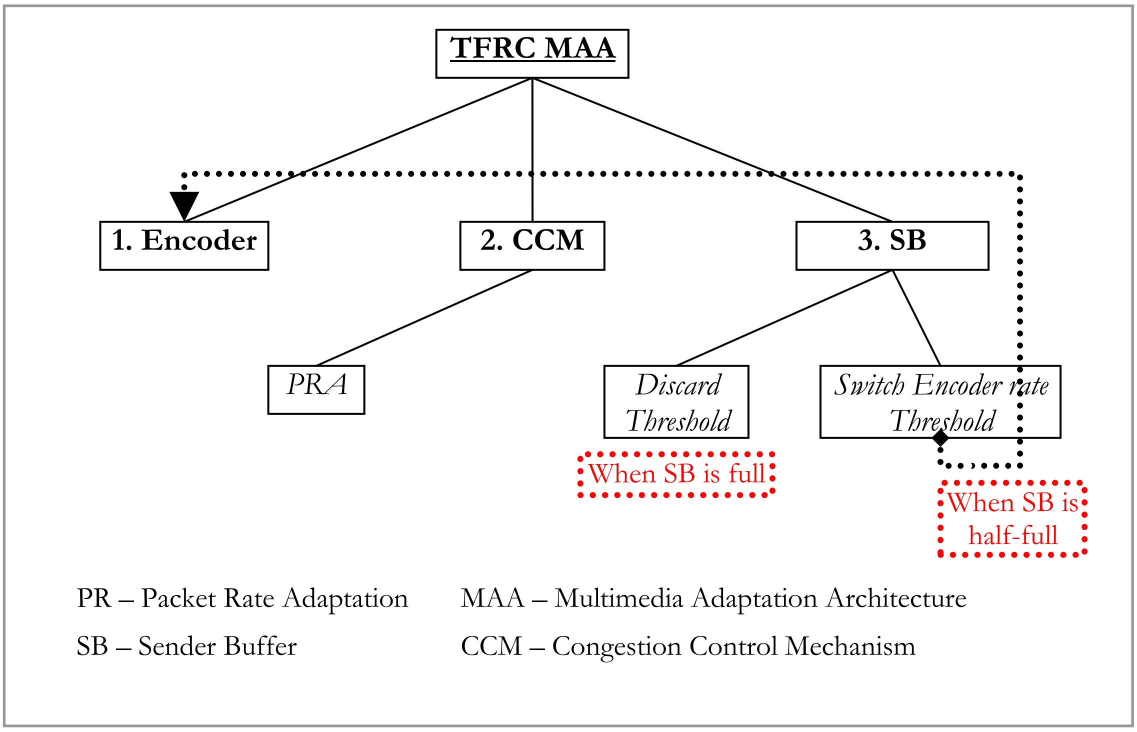


Figure 4.2, 'Components of the TFRC Multimedia Adaptation Architecture (MAA)'

4.1.1. Core functioning of the TFRC MAA in terms of Encoder Rate and Sender Buffer (SB)

Once the connection is initiated between the end-users, the encoder starts encoding the raw multimedia signal. The encoder produces frames at a fixed rate, which are placed in the Sender Buffer. The Sender Buffer is drained at a rate controlled by the Congestion Control Mechanism (CCM). At the receiving end the incoming data is placed into the Playout Buffer (PB). Once the PB is sufficiently full the receiver will start decoding the data at a constant rate which is ready to be heard or seen by the end-user.

The core component of the Multimedia Adaptation Architecture (MAA) is the Sender Buffer (SB). Its main purpose is to act as a temporary buffer for the encoded frames when the CCM's bitrate is lower than the encoder bitrate. The encoder bitrate is a function of the Frame Rate and Frame Size (measured in 'frames per second' and bytes respectively), as soon in Equation 3.2.

Encoder Bitrate:

An encoder can reduce its bitrate by either reducing its Frame Rate or Frame Size. However, with voice Encoders the bitrate can only be reduced in terms of frame quality whilst maintaining a fixed frame rate (the reasoning for this can be found in section 4.1.3.1). The frame quality reflects the Frame Size in terms of bytes.

Congestion Control Mechanism (CCM) Bitrate:

The CCM indicates the available network bitrate using a TCP response function, which is updated every Round Trip Time. This bitrate, T_{TCP}, is interpreted as a function of Packet Rate, P_R , and Packet Size, P_S, as shown in Equation 2.9.

Where 'S' is the fixed maximum Packet Size (bits), 't_{RTT}' is the Round Trip Time, 't_{RTO}' is the TCP Retransmission Timeout and 'l' is the 'loss event rate' experienced during the previous time interval of packets sent.

$$\text{Here: } S = P_S \qquad \textit{Equation 4.1}$$

$$\text{Available Network Bitrate, } T_{TCP} \text{ (bps)} = 8 \cdot P_S \times P_R \qquad \textit{Equation 4.2}$$

Here the CCM (TFRC) interprets the bitrate by adjusting the Packet Rate and maintaining a fixed Packet Size, this is referred to as '*Packet Rate Adaptation*' (PRA).

$$P_R = \frac{T_{TCP}}{8 \cdot P_S} \qquad \textit{Equation 4.3}$$

The drain rate of the Sender Buffer (SB) is equivalent to the Packet Rate, P_R, of the CCM as illustrated in Figure 4.4.

$$\text{Sender Buffer drain rate} = P_R \qquad \textit{Equation 4.4}$$

When the Packet Rate is lower than the Frame Rate the Sender Buffer will start to fill up. Otherwise, an equal or higher packet rate will drain the Sender Buffer and keep it empty[4].

When there is no network congestion, i.e. when the CCM indicates a Packet Rate equal to or higher than the Frame Rate, the SB will be empty. The multimedia Encoder will be operating at its maximum bitrate.

However, during congestion if the Packet Rate falls below the Frame Rate the SB will start to fill up. If the buffer occupancy reaches threshold of the 'low-encoder bitrate' see Figure 4.4), the SB will request the encoder to reduce its bitrate. For the majority of Voice encoders (including layered, coarse-grain, fine grain encoding: MPEG-2, FGS), they are only able to reduce their bitrate in terms of Frame Size and keep the Frame Rate fixed, hence they will not be able to reduce the occupancy of the SB, but will rather fill up the Sender Buffer over time until the Packet Rate increases. In such circumstances the SB will continue to fill and the Playout Buffer (PB) will continue to drain (and may well empty out if sufficient packets are not in the PB). When the SB reaches the discard threshold, the SB will discard all incoming frames generated by the encoder until the SB has room to accommodate at least one frame. The positions of the thresholds of the SB are illustrated in Figure 4.4. If this discarding behaviour continues the Playout Buffer (PB) may eventually empty out, because the lack of packets sent. Furthermore, the reduced Packet Rate, PR_C, causes packets to be sent at

[4] Assuming the Frame Size is no larger than the Maximum Transmission Unit (MTU) size, which is a constraint imposed by the Ethernet layer (1500 bytes), it can be safely assumed that a voice frame of 160 bytes [103] will fit into one packet without requiring fragmentation.

longer time intervals increasing the overall delay of arriving packets. The lower Packet Rate also causes packets to be temporarily buffered, introducing waiting delay. This increases the probability of packets arriving 'too-late'. Furthermore, as the Packet Rate (PR_C) reduces, this increases the Inter-Packet-Gap-Spacing (IPGS) between the packets sent ($P_{IPGS(S)} = {}^{1}/_{P_R}$), as illustrated in Figure 4.3. The increase in $IPGS_S$ of packets sent has a proportional increase in the $IPGS_R$ of received packets. A larger $IPGS_R$ of received packets demands for a larger de-jitter buffer (Receiver Buffer) to avoid packets being discarded as they have arrived 'too-late' in time sequence. Such a scheme can reduce the interactivity of the multimedia stream if the end-to-end delay becomes too large.

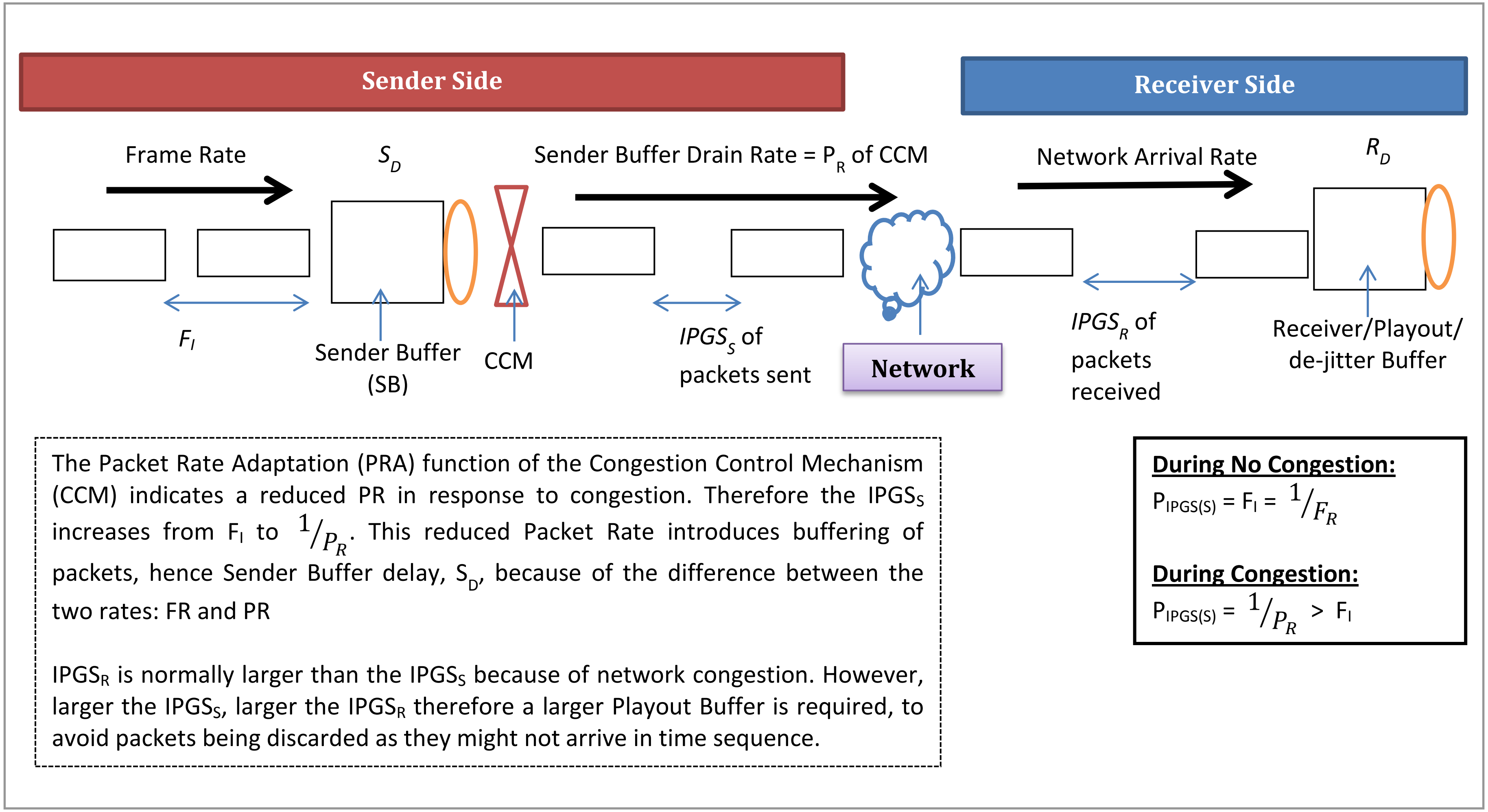
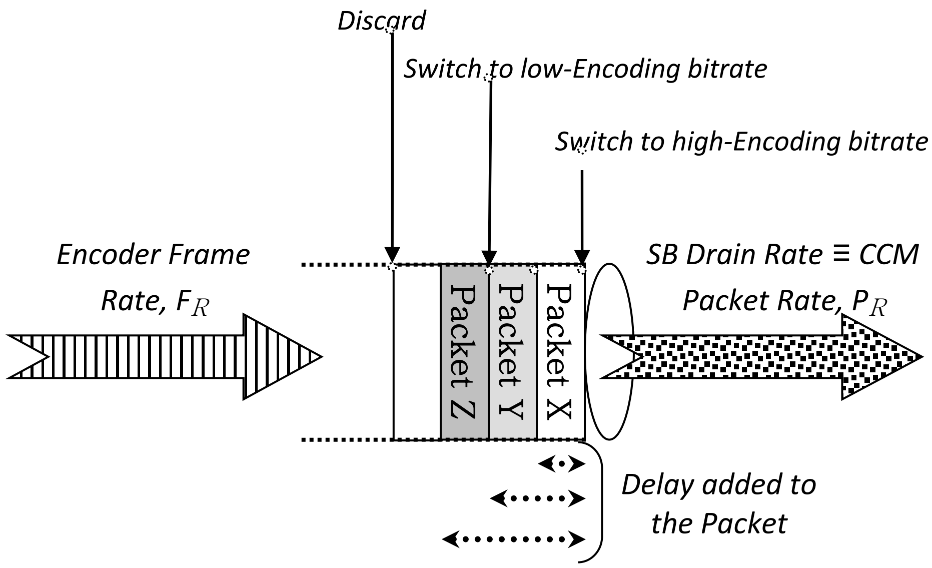


Figure 4.3, TFRC, Inter-Packet-Gap-Spacing (IPGS)

On the other hand, when congestion levels improve, the Packet Rate, PR_C, increases, the SB starts to drain, reducing the SB occupancy and discarding of packets. The PB occupancy increases improving the playout rate, hence improving the recipients' quality.



$F_R < P_R$ → Sender Buffer will Drain and eventually become Empty

$F_R = P_R$ → Sender Buffer may be occupied depending on the previous P_R however, it will not grow in size

$F_R > P_R$ → Sender Buffer will eventually fill up and start to discard frames

Figure 4.4, Occupancy of Sender Buffer

The next section will highlight the limitations of such an adaptation architecture, particularly when it comes to satisfying the QoS requirements (jitter, end-to-end delay and loss) of an Interactive Real-Time multimedia application. This will be illustrated by a simple voice stream. The CCM interpretation of the available network bitrate in terms of Packet Rate is discussed in detail. Furthermore, the occupancy of the SB is illustrated with reference to Packet Rate.

4.1.2. Congestion Control Mechanism's Operation

The Congestion Control Mechanism (CCM) indicates the available network resources in terms of bitrate (measured in bits per second). This is updated every Round Trip Time (RTT). The CCM measures the available bitrate, T_{TCP}, using the TCP response function (also known as the TCP rate-equation). The TCP rate-equation is a function of the Round Trip Time, retransmission timeout, and packet loss rate, as shown in Equation 2.9. (Further details of the evaluation of this rate-equation can be found in chapter 2 of this thesis and [52, 53].)

The CCM's response to the available network bitrate is to adjust the Packet Rate, PR_C, and maintain a fixed Packet Size, PS_C, and Equation 4.3 illustrates this. This is referred to as "Packet Rate Adaptation" (PRA).

4.1.3. Sender Buffer (SB): a temporary Buffer between Encoder Frame Rate (FR) and Packet Rate (PR) of the Congestion Control Mechanism's (CCM)

4.1.3.1. Delay

Sender Buffer Delay, S_D

This section illustrates how the Sender Buffer (SB) occupancy changes in response to available network bitrate. This is done with reference to a simple interactive voice application.

The voice encoder generates frames at a constant rate and reduces its bitrate in terms of Frame Quality resulting in a reduced Frame Size in bytes. This approach is offered by encoders such as MPEG-2, MPEG-4, FGS [7, 38, 39].

The Encoder Bitrate, (measured in bits per second) of a voice encoder is a product of the Frame Size measured in bytes and Frame Rate in frames per second (fps), as shown in Equation 3.2.

In the case of a voice encoder, the request to reduce the encoder bitrate will result in a reduction of Frame Size, instead of Frame Rate.

This method of bitrate adaptation is used for voice streams because during the talk-spurt the encoder generates frames at a constant rate, so they arrive at the receiver at a constant rate (ignoring network jitter). However, if the Encoder changes its Frame Rate during the talk-spurt, this adds jitter (in the form of increasing inter-packet-gap-spacing, 'IPGS') which introduces delay. For example a reduction of the Frame Rate during the talk-spurt will increase the time interval between the frames sent (referred as IPGS) in order to accommodate this change a larger de-jitter (or Playout) Buffer will be required at the receiver. This behaviour can result in a reduced interactivity if the total delay becomes too large. Therefore, voice encoders avoid changing Frame Rates and hence, offer 'Frame Quality adaptation'.

Once the connection is initiated between the two voice users, the encoder starts generating frames at a fixed Frame Rate of 50 fps of an adaptive frame size of 168 bytes (in the case of MPEG-2 encoders). These frames are placed in the Sender Buffer (SB). Ignoring the slow-start phase of the CCM, i.e. the CCM operating at a Packet Rate equal to or higher than 50 packets per second, the frames are sent into the network leaving the SB empty.

`

Now assume that congestion in the network forces the CCM to indicate a reduced available network bitrate, thus causing the Packet Rate (PR) to drop. A reduction in the Packet Rate will reduce the drain rate of the SB, and therefore the SB will start to fill up.

The SB will be occupied with frames; the rate at which they are sent into the network is determined by the Packet Rate indicated by the CCM.

For example if the PR drops to 47 pps and the Frame Rate remains fixed at 50 fps, the SB will be occupied with three frames at the end of the first second. The time the frames will temporarily wait in the SB before they are sent in the network is shown in Equation 4.5.

$$\text{SB Waiting Time of Frame}_{\mathrm{i}} = \left(\frac{1}{P_R} - \frac{1}{F_R} \right) \times \text{Position of Frame in SB}_{\mathrm{i}} \qquad \textit{Equation 4.5}$$

Consider using a finite SB of size 4, as defined in section 5.2.2.2 the maximum number of frames the SB can store is 4, before it starts to discard frames. The ‘switch to low-encoding’ threshold is set to half of the ‘discard’ threshold, as recommended by [51], i.e. 2 here. Figure 4.5 illustrates this SB configuration. Once the SB starts to fill up and reaches the ‘switch to low encoding’ threshold, the SB will signal the encoder to reduce its bitrate. The encoder will reduce its bitrate by reducing the quality of the frame. This is reflected in a reduced Frame Size (FS).

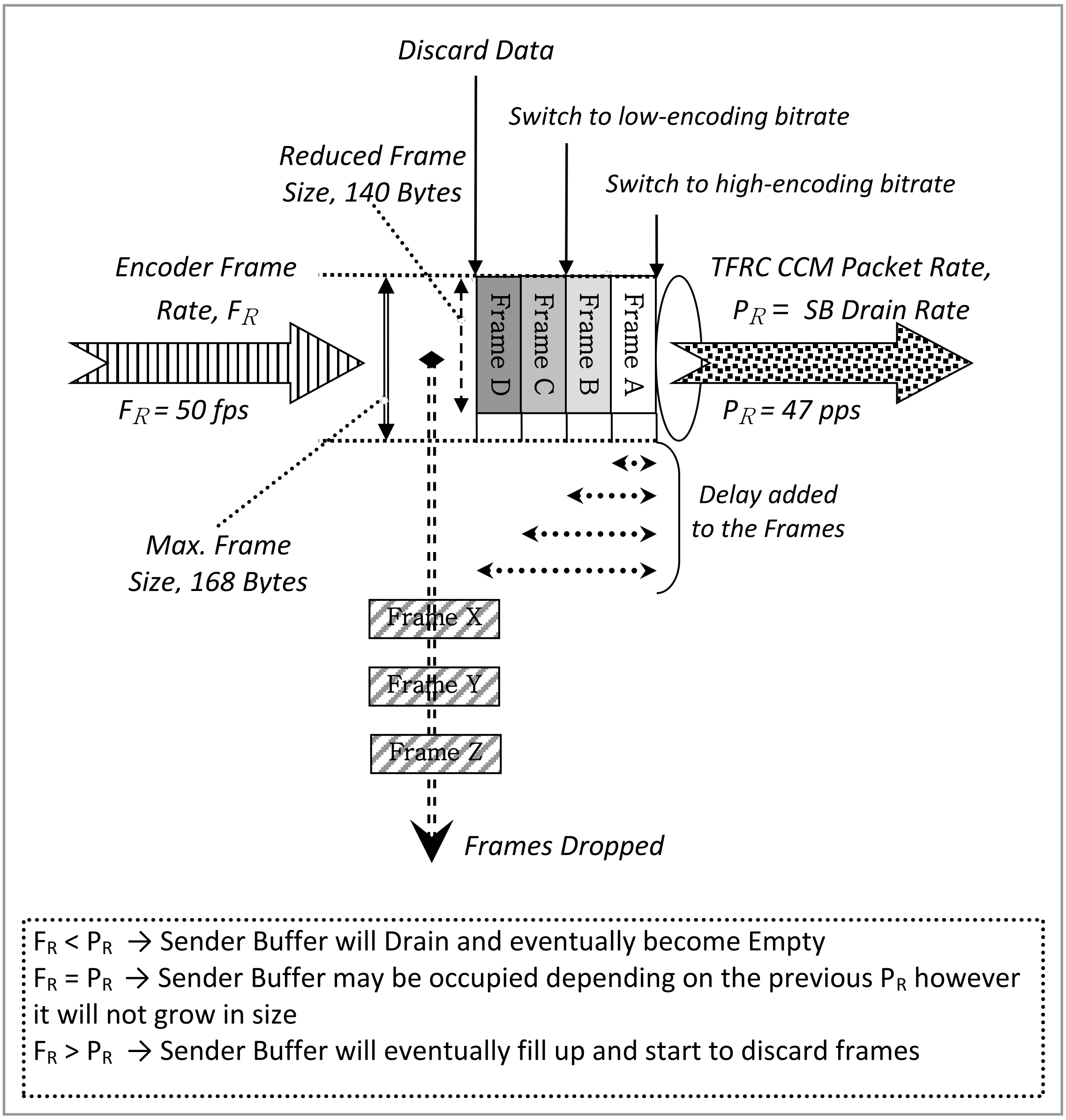


Figure 4.5, Occupancy of the Sender Buffer (SB), Audio Stream over TFRC MAA

The encoder will however, maintain the same fixed Frame Rate, 50 fps but at a reduced Frame Size. If the CCM consistently maintains a packet rate of 47 pps, the SB will be fully occupied and will therefore, discard incoming frames until space becomes available. This will happen even though the voice encoder has reduced its bitrate, because the reduction has been in terms of Frame Size rather than Frame Rate. The Frame Rate will remain higher than the Packet Rate (or the drain rate of the SB) resulting in the SB remaining full, and so the reduction of the encoder bitrate is of no benefit. The prime reason for this is that there is a mismatch between the two rates: Packet Rate and Frame Rate.

When the SB is full the SB will discard all incoming frames generated by the encoder until space becomes available in the SB. In addition to loss of frames, each frame that is in the SB would experience a waiting delay, S_D of:

$$S_D \text{ for each Frame} = \frac{1}{P_R} \text{ x Discard Threshold size in Frames} \qquad \textit{Equation 4.6}$$

Taking the same example where the Packet Rate is of 47 pps, causing the SB to fill up, each frame that enters the SB would experience a waiting delay of 85 ms.

Figure 4.6 illustrates the increases of waiting delay on packets in the Sender Buffer when the Packet Rate is reduced from its desired value 50 pps (equivalent to the Frame Rate of the encoder, 50 fps). Using the quality measurement scheme (found in section 2.1.4) which enables to quantify the impairment arising from delay and loss for a voice connection, Figure 4.7 shows the impact of that delay on quality.

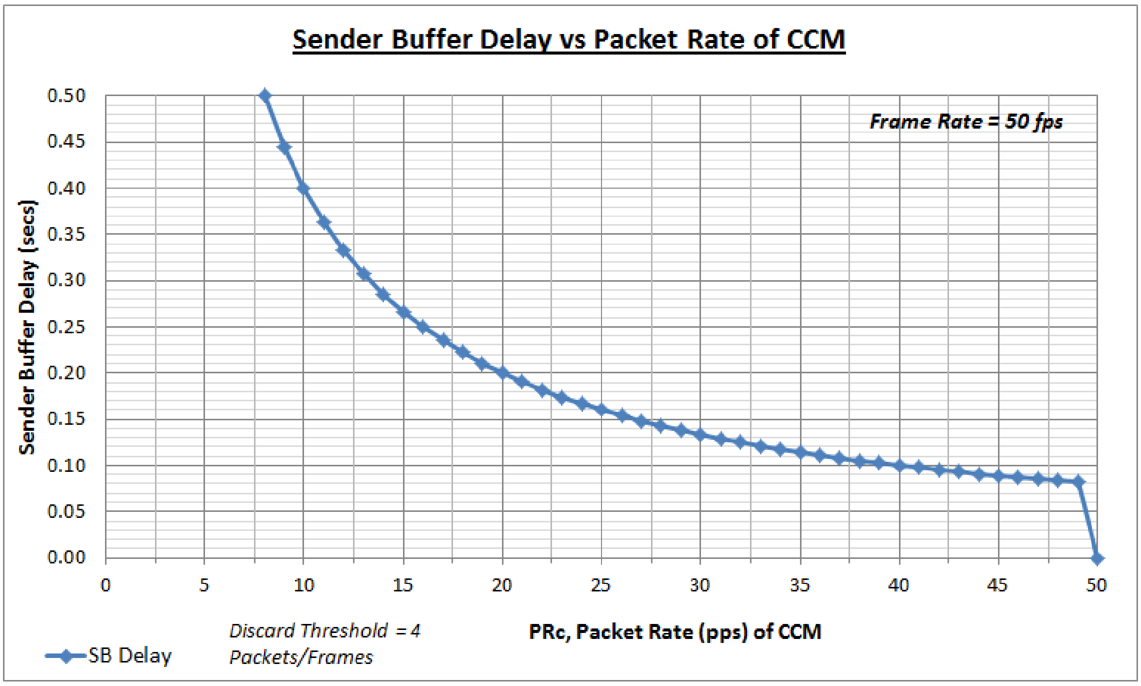


Figure 4.6, Sender Buffer Delay vs. Packet Rate of CCM

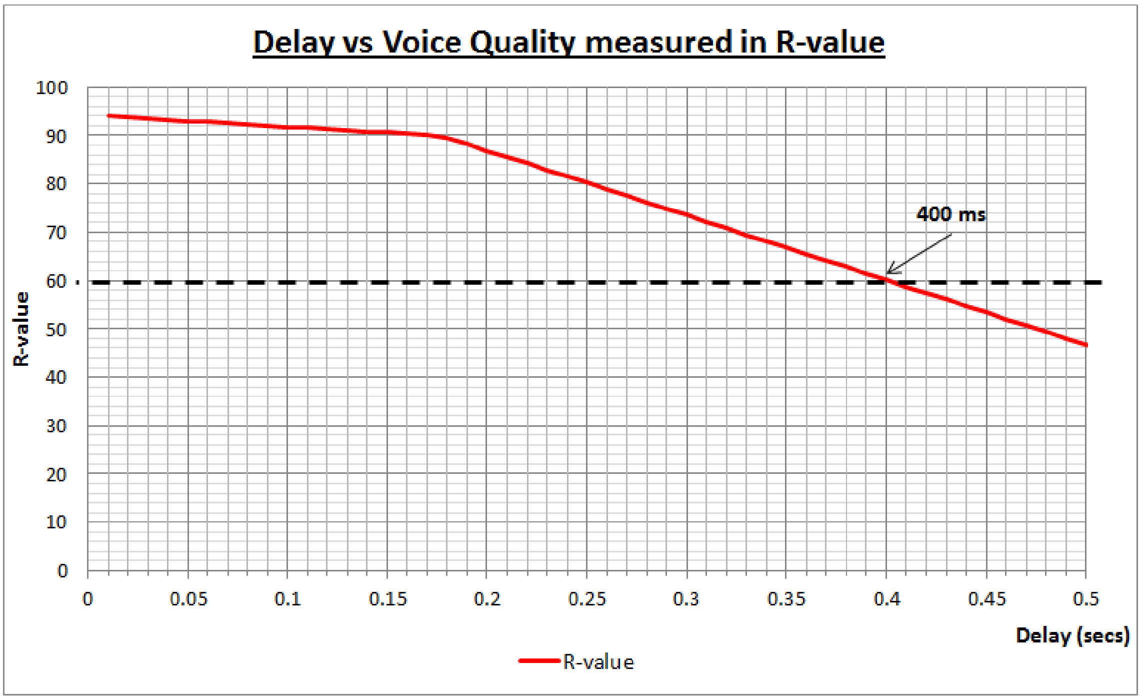


Figure 4.7, Delay vs. Voice Quality measured in R-value (graph formulated using Equation 2.8)

Packetization Delay, Z_D = $IPGS_{Sender}$

The packetization delay is normally equal to the Frame Interval (F_I) as defined in section 2.1.2. The Frame Interval is inversely proportional to the Frame Rate of the encoder, and the $P_{IPGS(S)}$ is equal to the F_I, when the P_R is equal to the Frame Rate.

$$\mathbf{Z_{D(R)}} = \mathbf{F_I} = \frac{\mathbf{1}}{\mathbf{F_R}}$$ *from* Equation 2.4

$F_I = P_{IPGS(S)}$ when $P_R = F_R$

else

$\mathbf{Z_D} = P_{IPGS(s)} = \frac{1}{P_R}$ *for all conditions*

Equation 4.7

However, if the P_R reduces in the case of the TFRC CCM due to network congestion, the $P_{IPGS(S)}$ increases and hence the Z_D increases proportionally, as shown in Equation 4.7.

The condition of $P_R > F_R$ does not exist as the maximum P_R is bound to the F_R, when the $P_R \geq F_R$. when the Packet Rate, P_R, reduces this increase the packetization delay, Z_D, which introduces delay impairment.

4.1.3.2. Loss

Considering the same voice example, a Frame Rate of 50 fps and a Packet Rate of 47 pps will result in a packet loss of 3 packets every second. If the same PR is maintained over time causing the SB to remain full will result in a SB loss, S_L, of 6 percent as calculated using the equation below.

$$S_L = \frac{F_R - P_R}{F_R}$$ *Equation 4.8*

The impact that loss will have on the quality of the voice call can be seen in Figure 4.8. A Sender Buffer loss of 14%, arising from a Packet Rate reduction to 43 pps, causes the quality to fall below R=60, which is unacceptable to the end-user.

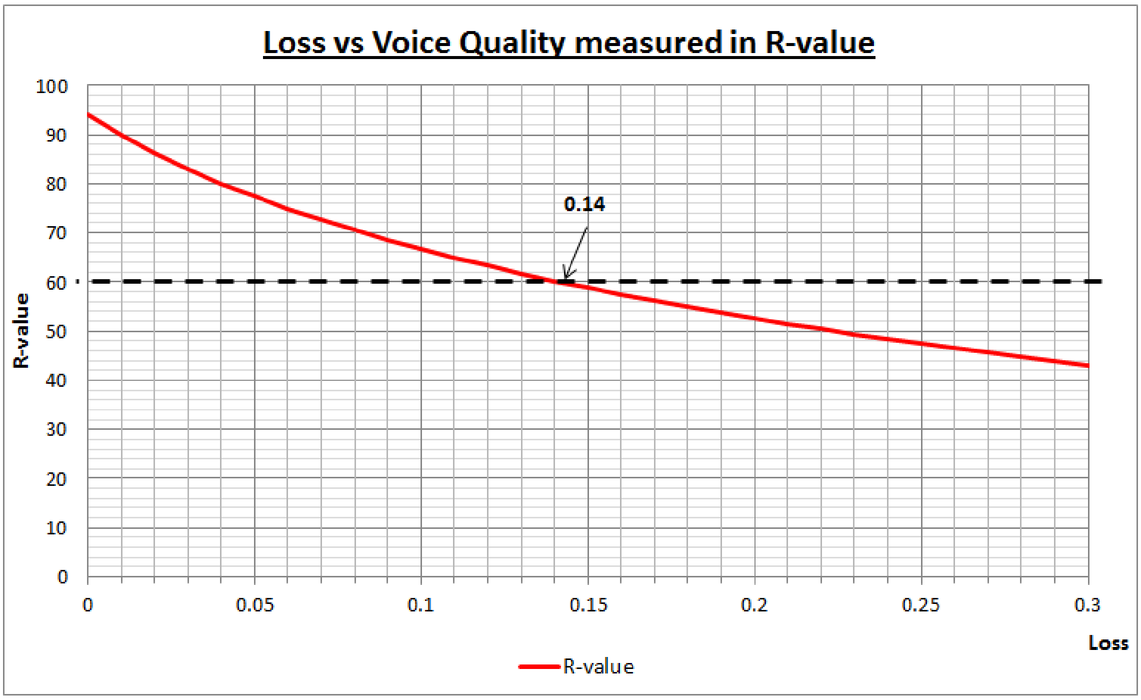
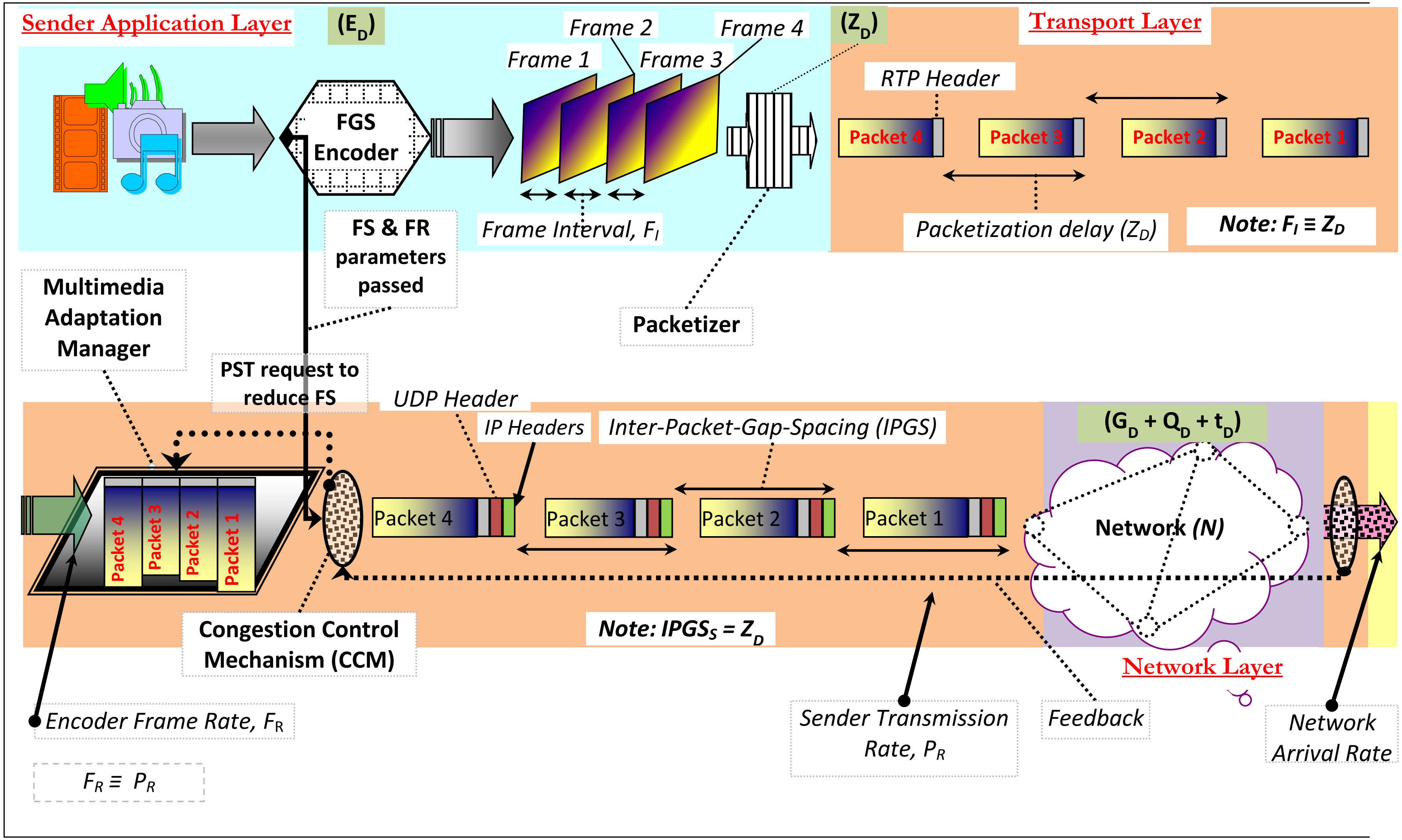


Figure 4.8, Loss vs. Voice Quality measured in R-value (graph formulated using Equation 2.7)

4.2. TCP friendly – Fine Grain Scalable (TFGS) Multimedia Adaptation Architecture

The aim of the novel TFGS Multimedia Adaptation Architecture (MAA) is to match application demand to network supply in such a way as to enable multimedia applications to maintain their interactivity whilst adapting their quality depending on the degree of network congestion. This is achieved by the Congestion Control Mechanism (CCM) operating at a fixed Packet Rate, equivalent to the Frame Rate of the Encoder, and responding to network congestion by reducing the Frame Quality reflected in the form of Frame Size as indicated by the Packet Size Truncation (PST) function of the CCM. This results in an isochronous service, i.e. there is no mismatch of the two rates; Frame Rate and Packet Rate. The CCM schedules packets at the same rate at which the frames are generated. This approach results in no loss or delay of packets, unlike the TFRC MAA, which adopts the Sender Buffer (SB), hence causing the buffering of packets at the sender side, in order to reconcile the two rates (Frame Rate and Packet Rate).

The novel Multimedia Adaptation Architecture integrates four components of a multimedia system together, (*1*) Application, (*2*) Multimedia Encoder, (*3*) Multimedia Adaptation Manager (MAM) and (*4*) Congestion Control Mechanism (CCM). These components reside at the sender side of the multimedia connection. The positions of each of these components are shown in Figure 4.11 and the interactions between each of the component are shown in Figure 4.10.


Sender Application Layer
(E_D)
FGS Encoder
Frame 1
Frame 2
Frame 3
Frame 4
(Z_D)
Transport Layer
RTP Header
Packet 4
Packet 3
Packet 2
Packet 1
FS & FR parameters passed
Frame Interval, F_I
Packetization delay (Z_D)
Note: F_I ≡ Z_D
Multimedia Adaptation Manager
Packetizer
PST request to reduce FS
UDP Header
IP Headers
Inter-Packet-Gap-Spacing (IPGS)
(G_D + Q_D + t_D)
Network (N)
Congestion Control Mechanism (CCM)
Note: IPGS_S = Z_D
Network Layer
Encoder Frame Rate, F_R
Sender Transmission Rate, P_R
Feedback
Network Arrival Rate
F_R ≡ P_R

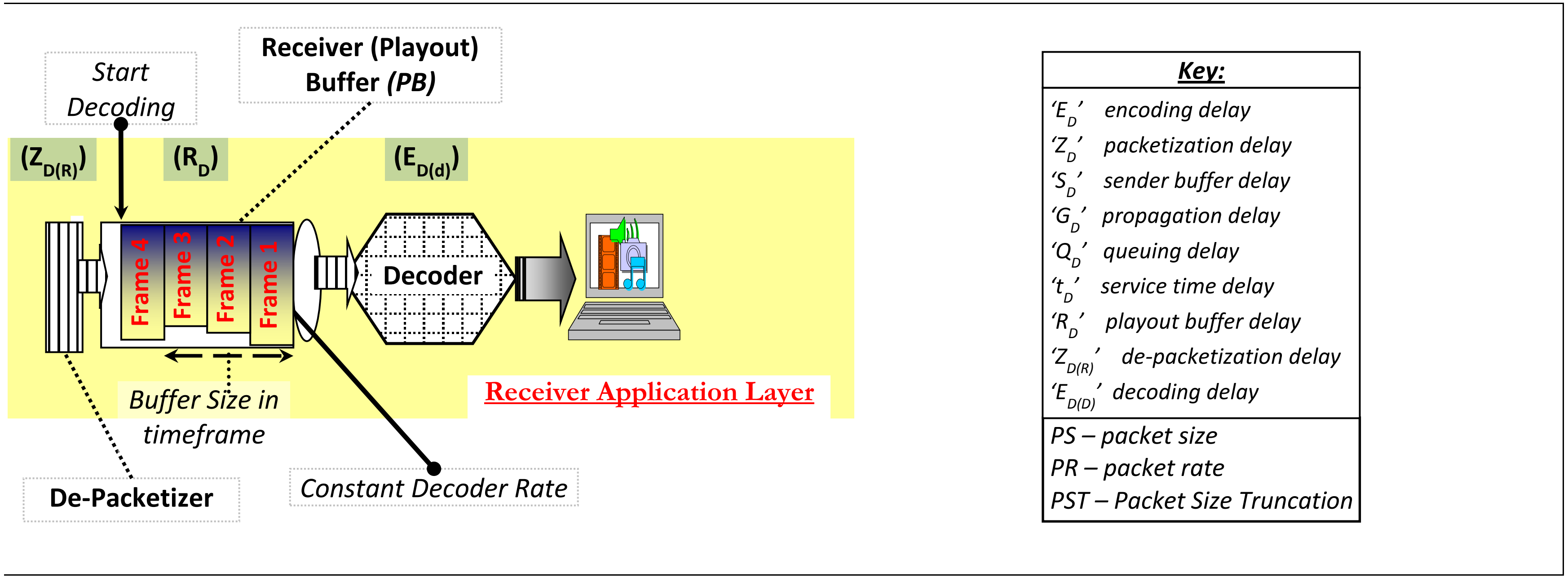


Figure 4.9, Multimedia System: for transporting multimedia streams over TFGS Congestion Control Mechanism

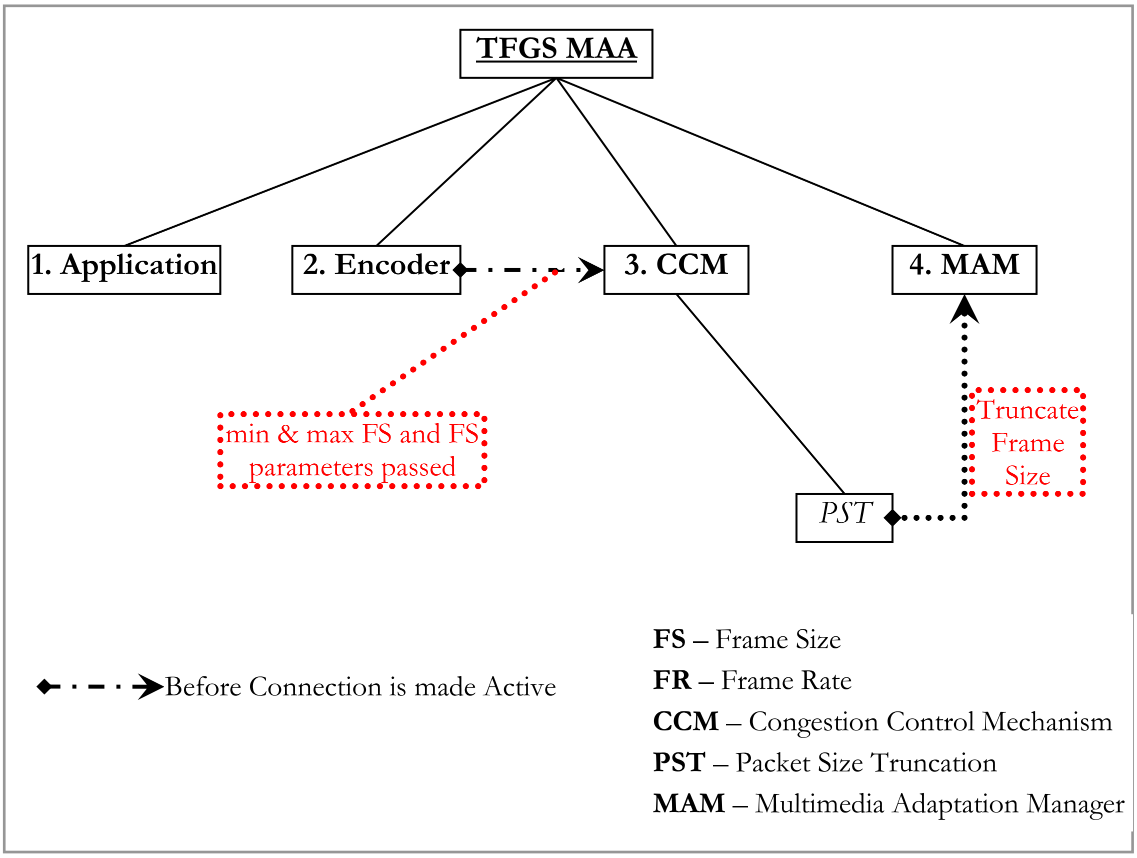


Figure 4.10, 'Components of the TFGS Multimedia Adaptation Architecture (MAA)

4.2.1. TFGS Congestion Control Mechanism

The core component of the TFGS MAA is the CCM, which is responsible for all decisions once the connection is active. The CCM indicates the available network resources using the TCP rate-equation shown in Equation 2.9, (in terms of a bitrate). This is updated every RTT.

S = *maximum Packet Size as defined by the multimedia application* *Equation 4.9*

The size, S , remains fixed throughout the calculation when using the TCP rate equation. This size can be referred to as the TCP Packet Size, S , and this may well be different to the Packet Size sent. This is because the Packet Size Truncation (PST) function truncates the Packet Size. Although the truncation is conducted by the Multimedia Adaptation Manager (MAM) of the packet to a size indicated by the PST function, however in this text it is referred to as the PST function which does both: i.e. it calculates the size and truncates the packet.

The variation in the Packet Size sent reflects the congestion in the network. The measurement of the t_{RTT}, t_{RTO} are exactly the same as of the TFRC CCM. The loss event rate, l, calculation is different and this is because the size of the packet which is sent varies over time depending on the condition of the network. The difference seen in the variation of the Sending Rate when comparing the two CCMs is because the 'loss event rate' calculated is slightly different. The TFGS CCM loss measurement mechanism is based on Virtual Packets (VP), this is scheme which combines small packets of size, s, to packet of size, S. When a Receiver receives a sum of S or more bytes from N number of small packets s, it records the arrival of a Virtual Packet of size S. Similarly a VP is marked lost when the amount of bytes lost exceeds S bytes, see Figure 4.11. Further details of this calculation can be found in section 5.1.1. By contrast, in the case of TFRC CCM, all packets are of fixed size and therefore a 'loss event' is based on the number of packets lost in the 'loss interval'.

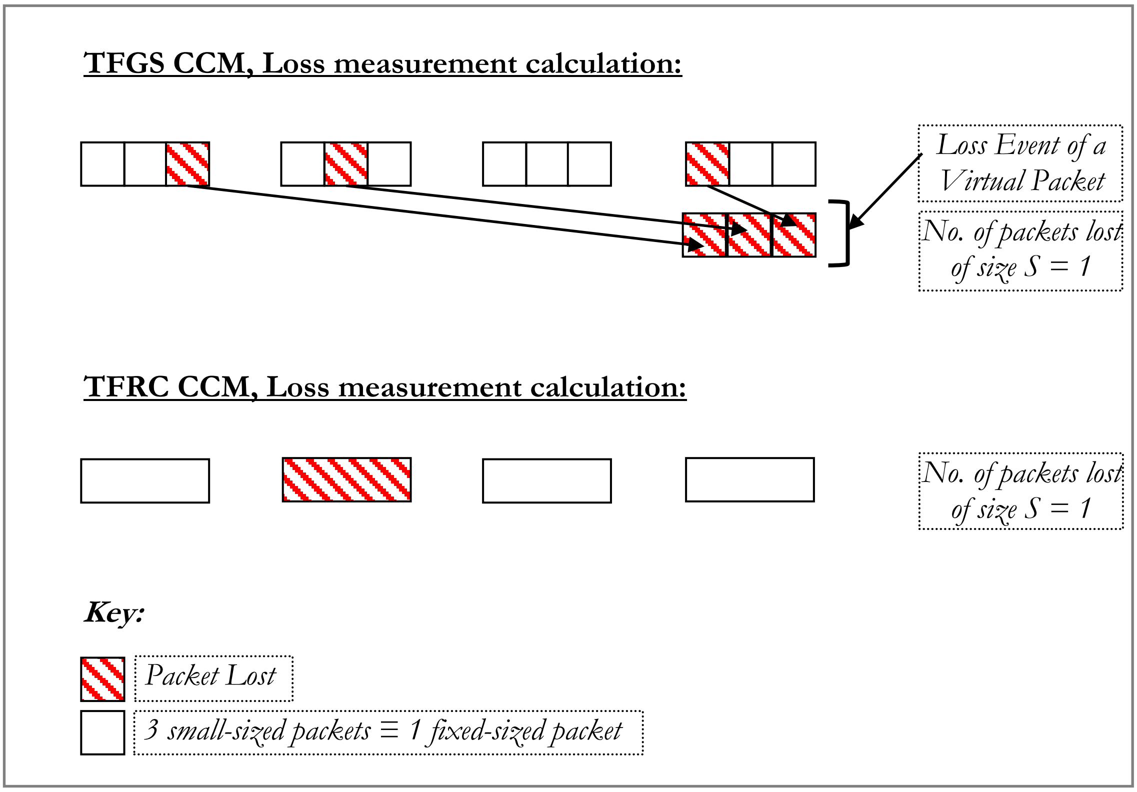


Figure 4.11, Loss measurement calculation

Packet Size Truncation (PST) function of the Congestion Control Mechanism (CCM):

The TCP Packet Size, 'S' in the TCP rate-equation is always set to a fixed maximum size, as requested by the multimedia application. Once the TCP rate-equation has indicated the available network bitrate, T_{TCP}, the 'Packet Size Truncation' function varies the Packet Size, P_S, of the packet sent, using the formulation shown in Equation 4.10.

$$P_S = \frac{T_{TCP}}{8 \cdot P_R} \qquad \textit{Equation 4.10}$$

Note: This $P_S = s$, as used in the Virtual Packet calculation for loss

The aim of the CCM is to maintain a fixed Packet Rate and adjust the Packet Size. This functionality is referred to as '*Packet Size Truncation*' (PST). For example, when the available network bitrate is of 128 kbps, the packet size is 320 bytes, and when the available sending rate is of 32 kbps, the packet size is 80 bytes and so on (considering the packet rate is fixed at 50 pps). This is illustrated in Figure 4.12.

Comparatively in the case of TFRC CCM when the available network bitrate is at 76.8 kbps, the packet rate is 30 packets per second (pps), and a bitrate of 102.4 kbps will result in a packet rate of 40 pps and so on; assuming the packet size is fixed at 320 bytes.

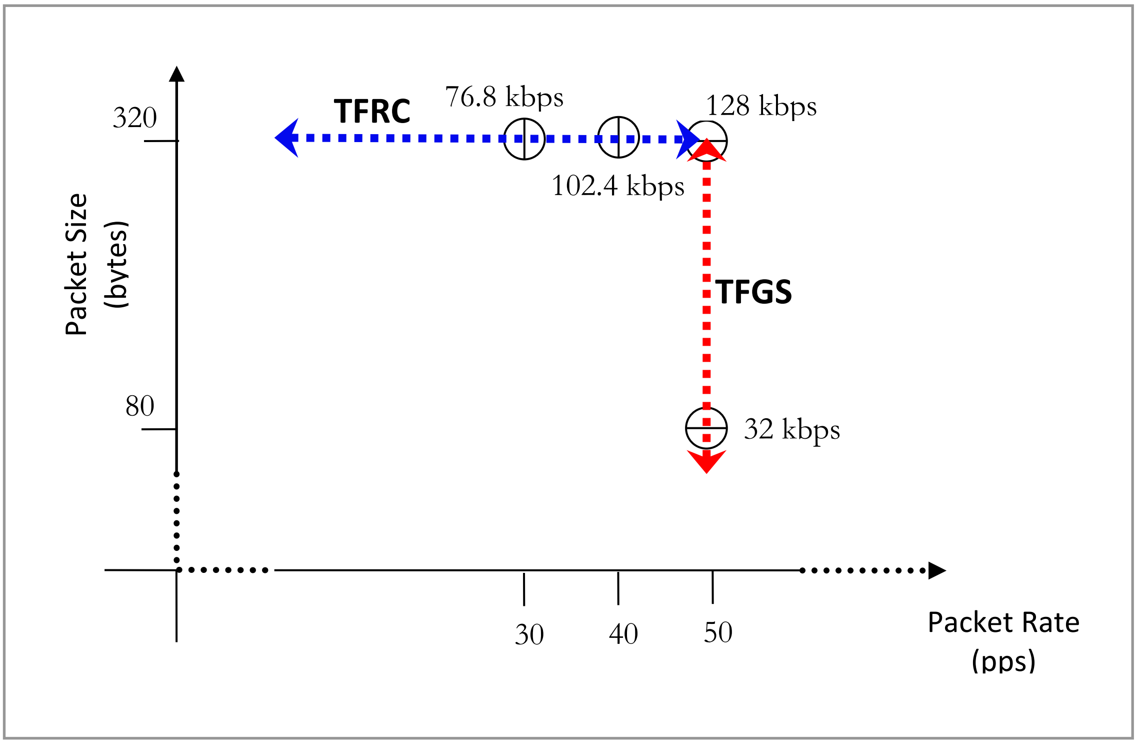


Figure 4.12, TFGS sending rate in respect to Packet Size

This approach allows voice encoders to send their frames as soon as possible, and so to avoid as much delay as possible. By maintaining a fixed Packet Rate, the CCM schedules (i.e. sends) the packets as soon as they are generated and responds to congestion by reducing the Packet Size. A reduced Packet Size induces a truncated Frame Size resulting in a reduced quality of the frame.

The Packet Size Truncation function of the CCM indicates the desired size of the packet. This information is passed to the Multimedia Adaptation Manager (MAM) which truncates the frame accordingly. Once the frame is truncated, the packet is then sent into the network. The time taken to truncate the frame is negligible, as the encoder does not need to re-encode the frame. It is just a matter of cutting off the 'Least Significant Bytes' (LSB) at the end of the frame.

Once the connection is initiated between the two voice users the CCM will operate at a fixed packet rate. The packet size will vary as the conditions in the network change.

For example, if the CCM indicates an available network bitrate higher than the maximum bitrate of the application, the application will be allowed to send data at its maximum bitrate. Any excess network resources will not be used.

Below, a voice bitstream is used to show how the Multimedia Adaptation Manager (MAM) will respond to a PST request.

Voice

In the case of a voice application which uses a Fine Grain scalable (FGS) encoder which generates frames at a fixed rate of 50 fps of a maximum frame size, 320 bytes, the TFGS CCM will send the frames at the same rate at which they are generated, i.e. the Packet Rate, PR_C is equal to the Frame Rate, FR_e. Hence there is no waiting delay at the sender.

When there is no congestion (i.e. the CCM indicates an available network bitrate equal to or higher than the maximum bitrate of the audio streams) the frames are sent at their maximum size at their fixed Frame Rate, resulting in a maximum bitrate of 128 kbps.

As congestion conditions change in the network, the network may not be able to support a maximum bitrate of 128 kbps. In such a case, the PST function of the CCM within the MAA will calculate the required size and pass this information to the Multimedia Adaptation Manager (MAM). The MAM will truncate the encoded frames to the desired size. Once the frames are truncated, they are sent over the network. The adaptation is instantaneous (i.e. on-the-fly) as it requires no re-encoding of the frames, and therefore the truncation time is negligible [54, 43].

The truncation of the frames will result in a reduced quality of the frame, but will maintain a fixed Packet Rate, in order to keep interactivity high. The truncation of the frame is made possible by using ‘Fine Grain Scalable’ encoding, where each frame is scalable to byte-level granularity [54]. This means that if the CCM desires a size of 167 bytes, then the frame can be truncated exactly to that size. By contrast, ‘Coarse Grain’ encoding provides quantized level of granularity (for example in increments of 10 bytes). This means the frame will adapt to 160 bytes losing 7 bytes of frame quality. Encoding schemes such as Coarse Grain have low granularity i.e. large quantization levels in terms of bitrate options and this can lead to steep changes in quality which can be annoying to the end user [6, 7]. Using a Fine Grain Scalable (FGS) encoder it is possible to achieve a desired match of application demand to network supply at byte-level granularity.

Figure 4.13 illustrates how the Packet Size will vary according to changing network capacities. For example, if the available network bitrate indicated by the CCM is of 64 kbps, then the Packet Size will be truncated to a size of 160 bytes, whilst maintaining a fixed packet rate of 50 packet per second (pps). The impact that Packet Size truncation will have on the end-user quality is discussed in detail in section 6.2. A packet truncated to a size lower than 70 bytes (excluding packet headers) will result in an R-value below 60, which is of unacceptable quality for the end-user.

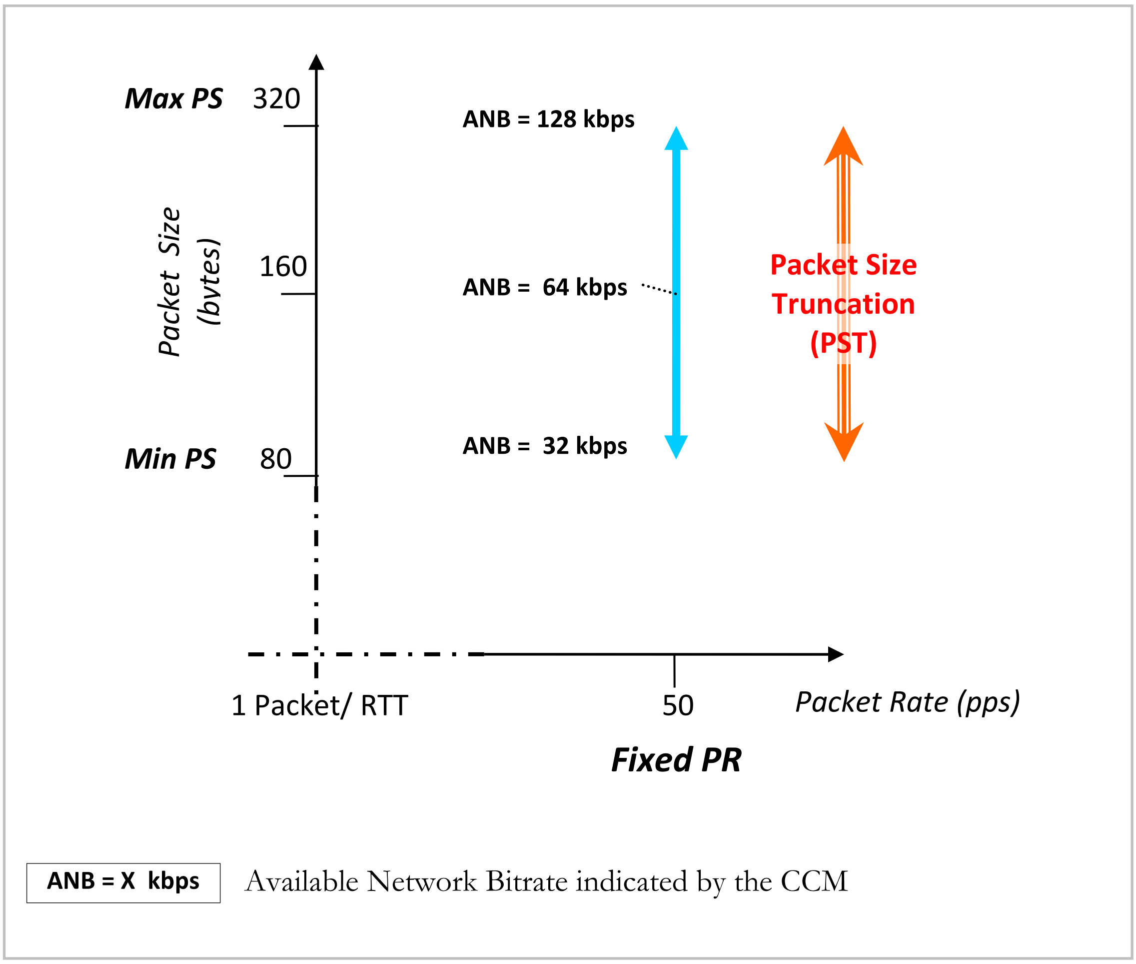


Figure 4.13, TFGS CCM Operation

4.3. Summary

This chapter gives a comparative study of how the TFRC and TFGS Multimedia Adaptation Architectures (MAA's) operate. It highlights the difference between them, and the problems associated with the TFRC MAA particularly addressing the operational issue of the Congestion Control Mechanism (CCM) (for the TFRC MAA, which responds to congestion by varying the Packet Rate when the application generates frames at a fixed rate, where this rate cannot adapt to that of the Packet Rate). This leads on to the problem of the Sender Buffer (SB) which introduces delay and loss when it is used as a temporary buffer between the two rates (Packet Rate and Frame Rate).

An alternative, novel, MAA (referred to as TFGS MAA) is described. This avoids the above problems (of SB delay and SB loss). The chapter goes into detail as to how this is achieved:

a) Changing the way the CCM adapts to congestion. TFGS CCM adjusts the Packet Size rather than the Packet Rate. This keeps the two rates (Packet Rate and Frame Rate) equal, avoiding the need for buffering. This functionality is referred to as Packet Size Truncation (PST).
b) By taking advantage of Fine Grain Scalable (FGS) adaptive encoding the multimedia frames can be truncated to the size indicated by the PST function of the CCM at byte-level granularity. The truncation of frames is carried out by the Multimedia Adaptation Manager (MAM).

The next chapter, 6, and simulation chapter 7 quantifies the operation of the two MAA's with respect to Quality of Experience (QoE). For example:

a) How much of a Packet Rate reduction can be tolerated before losing acceptable quality levels in voice calls

b) How many additional acceptable voice flows can be carried by the network whilst using either the TFRC MAA or the TFGS MAA.

5. Simulation Methodology

The factors that undermine the quality of a multimedia connection are delay, loss and jitter; these impairment factors occur whilst in transmission of multimedia packets over the network due to congestion and at the end system.

The analysis conducted in Chapter 6 focused on the impairment arising from the method of congestion response used by the CCM, i.e. how a reduced available network bitrate can have an impact on the sender side of the multimedia connection. This can result in buffering of frames which introduces 'Packet Level Impairment' or by Frame Size truncation which introduces 'Byte Level Impairment'. This analysis can be referred to as sender side impairment of the multimedia connection excluding the network and receiver impairment.

This chapter (including 7) focuses on the latter two impairments including sender side impairment, highlighting the impairment caused by the network loss and delay, and receiver loss due to packets arriving 'too-late'. This will be achieved using a simulation study, which will give a fuller picture of the QoE a multimedia application will experience when it operates over a TFRC or TFGS Multimedia Adaptation Architecture (MAA). This chapter illustrates how the simulation study is structured and chapter 7 shows the results following on with a detailed analysis.

The network simulator used here is an open source simulator known as '*ns2'*, which is widely used by academic researchers. The '*ns*2' simulator is particularly appreciated for the work on the network layer and transport layer, particularly in the Active Queue Management and TCP domain [57]. All major TCP-friendly mechanisms such as RAP, TFRC [12, 21] have been implemented and tested using '*ns2*'. This gives confidence to a new developer to use '*ns*2' and avoids unnecessary development of code.

'ns2' is an event driven packet level simulator which keeps track of each and every event over time. For example, the simulator can give details of the time when the packet was created, sent into the network, received at its destination and, if dropped, when and where. This functionality enables users to make thorough investigation and verification of the results achieved.

The key difference between the two MAA is how the Congestion Control Mechanisms (CCMs) operate, TFRC adapts its Packet Rate in response to network congestion whereas TFGS adapts its Packet Size (and maintains a fixed packet rate). It is this core difference which will be addressed in the simulation study. The CCM is in control of the Packet Size and Packet Rate response to congestion.

This chapter is organised in three main sections, the first section introduces how the TFGS code is implemented in *ns*2. The second section goes into detail of the simulation methodology elaborating on how the measurements are made in *ns*2, what parameters are used for traffic resources and highlights the network scenario description. The last (third) section verifies whether both the CCMs operate in the manner designed.

5.1. Implementation of the TFGS CCM

The implementation of the TFGS CCM code is based on the current code available for the TFRC CCM. The original code of the TFRC CCM had been developed and tested in *ns*2 [9, 57]. The available network bitrate, T_{TCP}, is calculated by the rate-equation as shown in Equation 2.9 (from chapter 2). The TFRC CCM interprets this bitrate as a function of varying the Packet Rate, P_R, and keeping the Packet Size, P_S, fixed (shown in Equation 4.3) and this is referred to as the 'Packet Rate Adaptation' (PRA) function.

$$P_R = \frac{T_{TCP}}{8 \cdot P_S} \qquad \textit{from Equation 4.3}$$

Here, $S = P_S$ of the CCM

Using the same code of the TFRC CCM, modifications are made to the way the available network bitrate is interpreted, i.e. as a function of varying the packet size, P_S, and keeping the Packet Rate, P_R, fixed (as shown Equation 4.3), this is referred to as the Packet Size Truncation (PST) function, which incorporates the TFGS CCM.

The Packet Size, 'S' in the TCP rate-equation is always set to the fixed maximum size as requested by the multimedia application. Once the TCP rate-equation indicates the available network bitrate, T_{TCP}, the 'Packet Size Truncation' function then varies the Packet Size of the packet sent, using the formulation shown in Equation 4.10.

Once the available network bitrate is calculated, the PRA or the PST function of the two CCMs are invoked. This controls the scheduling of the packets. The pseudocode for both the CCMs is shown below.

```
PRA_TFRC_Agent::NextPacket() {

  InterPacketGapSpacing = PacketSize /TransmissionRate
  double Min_ InterPacketGapSpacing = 1.0 /PacketRate_Max
  If (InterPacketGapSpacing < Min_ InterPacketGapSpacing){
       InterPacketGapSpacing = min_interval_PR
  }

  send_timer_schedule (InterPacketGapSpacing)
}
```

Code 6.1, Packet Rate Adaptation (PRA) Code used in the TFRC CCM

For the PRA function, the Packet Rate, P_R, is reflected by varying the Inter-Packet-Gap-Spacing (IPGS), as shown in line 2 in the code above.

$$P_{IPGS} = \frac{1}{P_R} \qquad \textit{Equation 5.1}$$

If the calculated Packet Rate of the CCM is greater than the required Packet Rate as defined by the multimedia application, the Packet Rate of the CCM is then restricted to the maximum limit of the multimedia application. This is illustrated in the code. That code is written in the context of Inter-Packet-Gap-Spacing (IPGS). Once the correct IPGS is calculated, the next packet is scheduled after this time spacing.

The PST function of the TFGS CCM is calculated differently as defined in Equation 4.10. However, if the Packet Size calculated is greater than the set Packet Size (as defined by the multimedia application) the Packet Size is restricted to the maximum limit, of the multimedia application as for the PRA function. Furthermore, if the Packet Size is lower than a size of 41 Byte (including 40 Bytes for headers) the size is set to a fixed minimum of 41 Byte.

Once the correct Packet Size is determined the next packet is scheduled after a fixed time spacing as defined by the Inter-Packet-Gap-Spacing ($IPGS_S$).

```
PST_TFGS_Agent::NextPacket() {

  PacketSize = int (TransmissionRate /PacketRate)
  If (PacketSize > Max_PacketSize) {
```

```
        PacketSize = Max_PacketSize
    }
    Else If (PacketSize < 41) {
        PacketSize = 41
    }

    InterPacketGapSpacing = 1.0 /PacketRate
    send_timer_schedule (InterPacketGapSpacing)
}
```

Code 6.2, Packet Size Adaptation (PSA) Code used in the TFGS CCM

5.1.1. Correction to the TCP Rate Equation

The formulated equation for TCP, shown in Equation 2.9, gives a fair estimate of the TCP transmission bitrate using a fixed packet size of '*S*' (for example 576 bytes). However, voice applications generate small packet sizes (i.e. 160 bytes) compared to a TCP connection. When using Equation 2.9 to calculate the approximate TCP transmission bitrate this voice connection will experience a lower transmission bitrate by a factor of 3.6. Therefore, the flow will not get its fair share of bandwidth when competing with a 576 byte flow [58, 59]. Work reported in [60] corrects the equation-based model so applications sending small packets would experience the equivalent transmission bitrate of an application sending a large packet of size, 'S'. This is achieved by setting the Packet Size, '*S*', in the TCP rate-equation to the same size as the competing flows such as TCP. So the available network bitrate, T_{TCP}, indicated by the TCP rate-equation is now equivalent to that of the TCP application. The Packet Size sent may be of a smaller size, however the overall bitrate will be equivalent because the packets sent are at a higher Packet Rate in packet per second (pps).

Furthermore corrections to the 'loss event rate', 'l_{er}', were also made, to take into account small packet sizes. Work reported in [60] modifies the loss measurement mechanism and this is based on Virtual Packets (VP). The TFGS CCM loss measurement mechanism is based on Virtual Packets (VP), this is scheme which combines small packets of size, s, to packet of size, S. When a Receiver receives a sum of S or more bytes from N number of small packets s, it records the arrival of a Virtual Packet of size S. Similarly a VP is marked lost when the amount of bytes lost exceeds S bytes, see Figure 5.1. By contrast, in the case of TFRC CCM, all packets are of fixed size and therefore a 'loss event' is based on the number of packets lost in the 'loss interval'. To apply this method, it is necessary to redefine the destination is 'loss event' and 'loss interval'.

Loss Event (LE): A packet loss constitutes a 'loss event' if at least 'S' bytes are lost, and this is referred to as 'loss event' of a Virtual Packet. Loss Interval (LI): is measured as the number of virtual packets between two successive loss events, including the lost packet that ends the loss interval.

Further details of this calculation can be found in [60].

By contrast, in the case of TFRC CCM, all packets are of fixed size and therefore a 'loss event' is based on the number of packets lost in the 'loss interval'. Figure 5.1 illustrates the method of how the loss measurement mechanism is interpreted when using the corrected loss event calculation for small packets, s, compared to the method of using fixed sized packets of size, S.

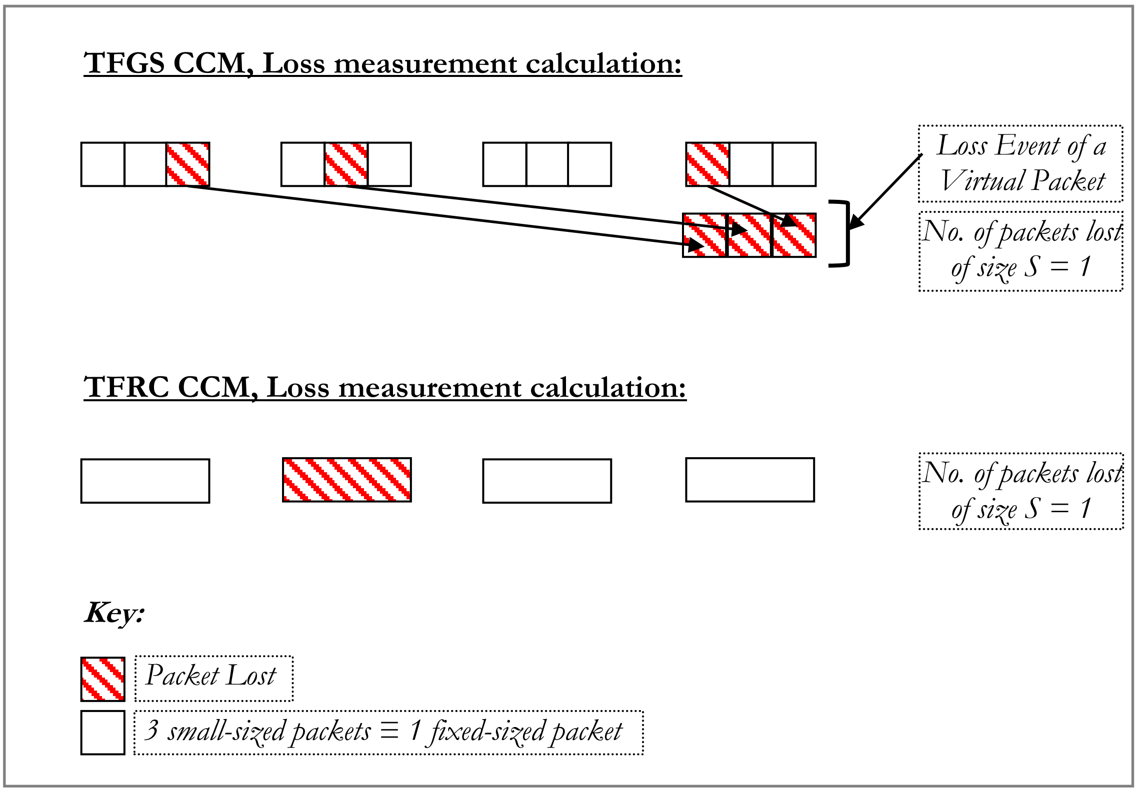


Figure 5.1, Loss measurement calculation

5.2. Methodology of Simulation

5.2.1. Measurement & Post Processing of Data from network simulator

The '*ns*2' simulator enables the user to simulate the behaviour of the traffic source and of the transport layer. The encoder component in the Multimedia Adaptation Architecture (MAA) is referred to as a traffic source agent in '*ns*2'. The CCM is referred to as the transport agent in '*ns*2' as it controls the scheduling and adaptation of packets that are sent into the network hence, the Packet Rate, PR_C, and Packet Size, PS_C. Figure 5.2a illustrates this linkage.

The remaining two components of the TFGS MAA (Multimedia Adaptation Manager (MAM) and Application) are not simulated, as they are only involved in a real implementation. The Sender Buffer (SB) delay component of the TFRC MAA is estimated, because the TFRC MAA has not been standardised for '*ns*2' or any other software package. Details can be found in section 5.2.1.2 to why the SB component is estimated.

Table 5.1 lists components of both the MAAs (TFRC and TFGS), and states if they are simulated or not. If simulated how is their behaviour traced, and whether the traced behaviour is precise or an approximation.

MAA	Components of the MAA	Simulated (Yes/No)	Variable used to trace the behaviour	Traced Behaviour: Precise or an Approximation (P/A)
TFRC	SB Delay	N	Inter-Packet-Gap-Spacing at Sender, $IPGS_S$	A
	SB Loss	N	Duration of Connection and Number of Packets Sent by the CCM	P
	Encoder	Y	Packet Rate	P
	CCM - PRA	Y	Inter-Packet-Gap-Spacing at Reciever, $IPGS_R$	P
TFGS	Application	N	n/a	n/a
	Encoder	Y	Packet Rate	P
	MAM	N	n/a	n/a
	CCM - PSS	Y	Packet Size	P

Table 5.1, Variables Traced by the Simulator

Using the simulation study the performance of the two MAA can be compared by monitoring the loss, delay, jitter, Packet Rate and Packet Size values experienced by the voice connections, which can then quantify the Quality of Experience (QoE) in terms of an R-value, on a scale of 0 to 100.

The section below will go into detail how the loss, delay, jitter, Packet Rate and Packet Size values are monitored from the simulator, and Figure 5.3 will illustrate the position from which the data is extracted, in the '*ns*2' framework.

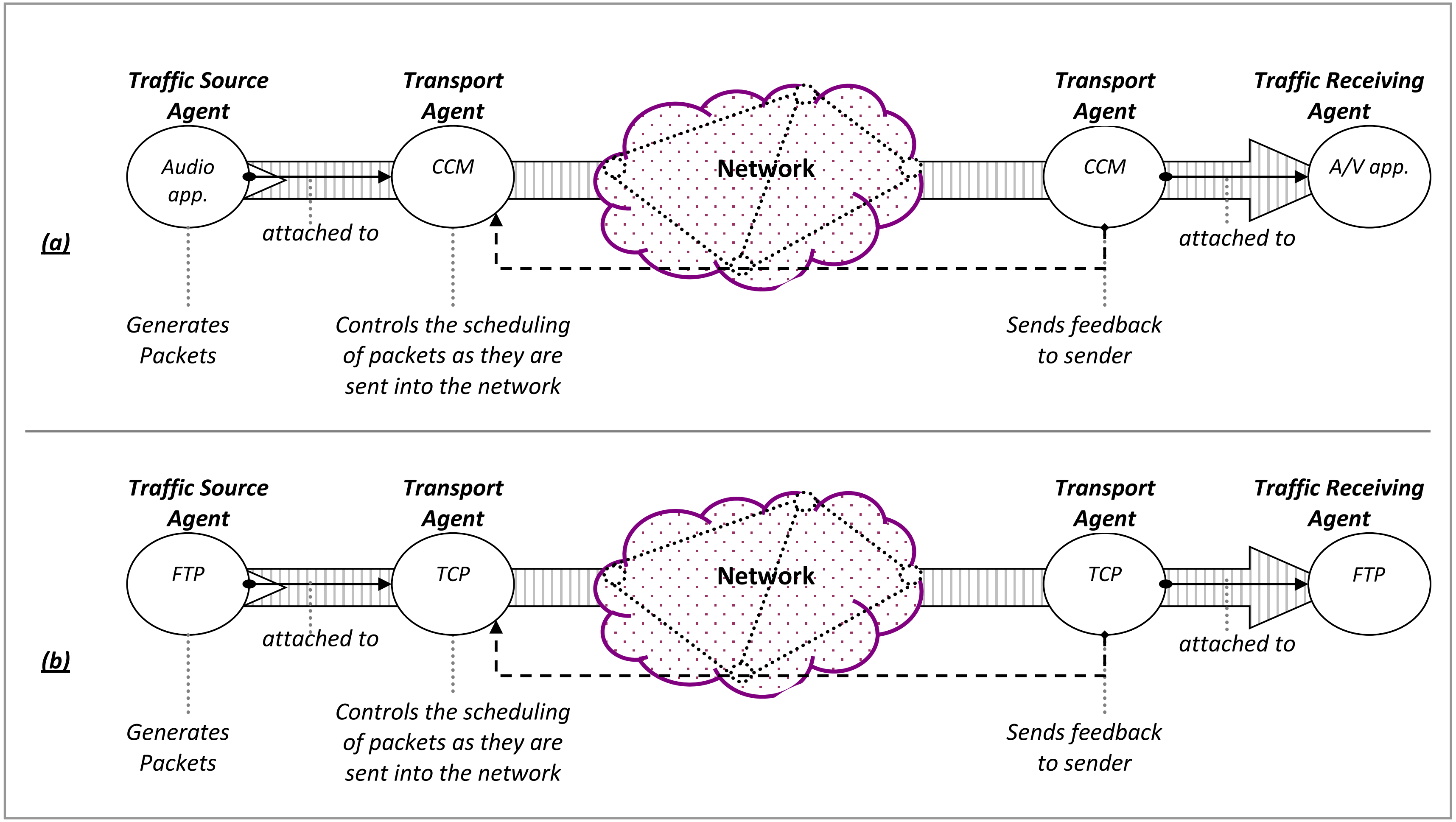


Figure 5.2, ns2 linkage between Traffic source and Transport Agent

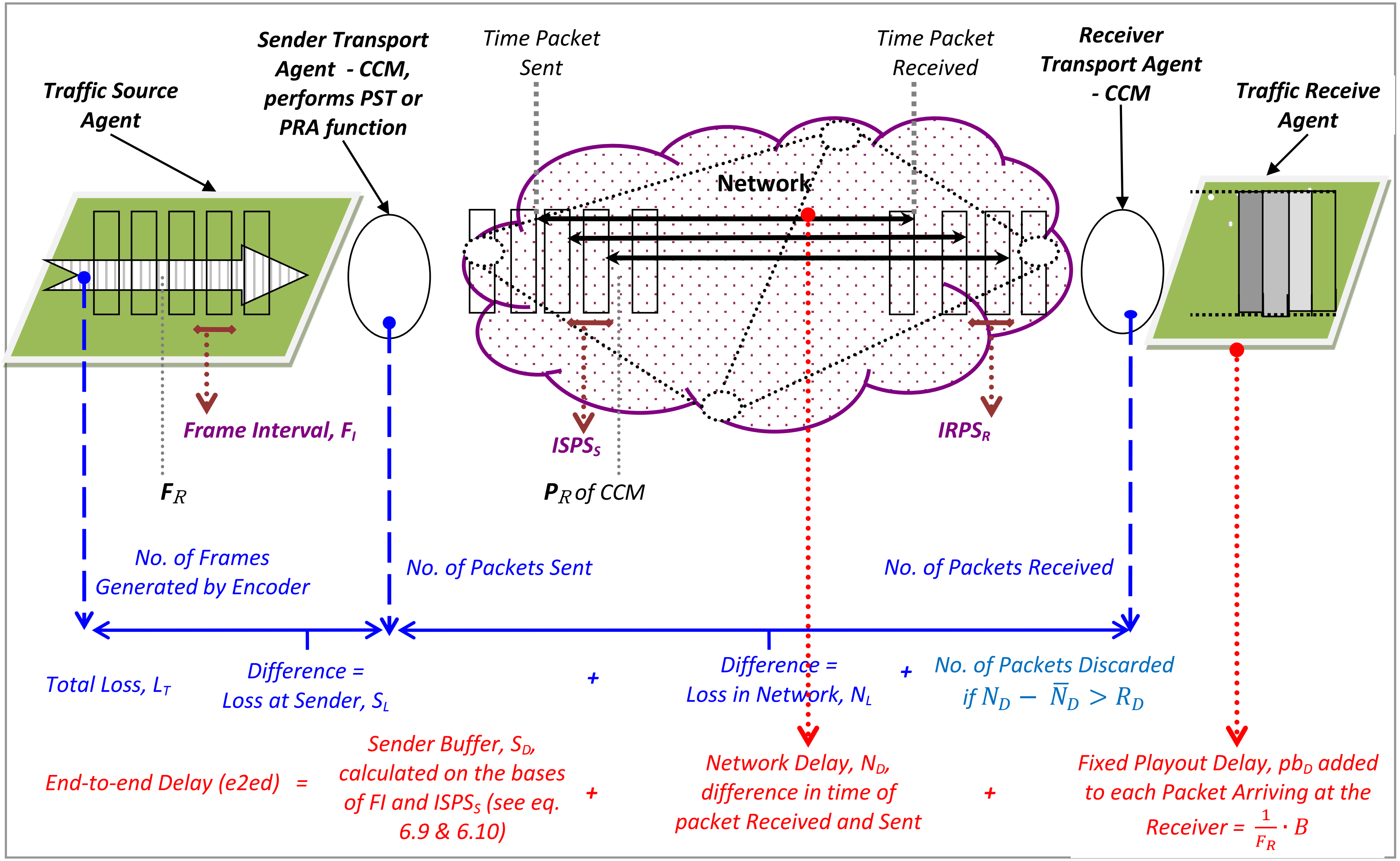


Figure 5.3, Delay and Loss Components

5.2.1.1. Loss

The total loss ratio, $T_{L(R)}$, (as a fraction of the Total Number of Packets Generated by the multimedia application, T_{NPG}) experienced by the multimedia stream is the addition of: Sender Buffer loss, SB_L, Network loss, L_N, and the loss of packets that have arrived 'too late' at the Receiver, PB_L. This is expressed in

Equation 2.2 (chapter 2), which is shown below.

$$\boldsymbol{T_L = SB_L + N_L + PB_L}$$ *from* Equation 2.1

from

$$\boldsymbol{T_{L(R)} = \frac{T_L}{T_{NPG}}}$$

Equation 2.2

The Sender Buffer, S_L, loss is calculated by taking the difference between the number of packets sent via the CCM and the number of frames generated by the encoder (in *'ns2'* this is known as the traffic source agent). Note one frame fits into the payload of one packet, i.e. the frames are not fragmented across packets therefore, making the calculation below valid. The simulator will indicate how many packets have been sent via the CCM and by recording the duration of the connection, this will indicate the number of frames that would have been generated by the encoder during this time period. This is illustrated in the equation below:

n_{frames} *generated =* $\boldsymbol{P_{NPG}}$ *= Duration of Connection × F_R of Encoder*

(no.) (sec) (Frames /sec)

Equation 5.2

Calculating the difference between the number of packets sent via the CCM and the number of frames generated gives an accurate measure of the number of packets dropped by the Sender Buffer, S_L, during this connection period.

$$\boldsymbol{S_L} = n_{frames} - n_{packets} \textit{ sent via the CCM} \qquad \text{Equation 5.3}$$

The Network Loss, N_L, is calculated by taking the difference between the number of packets received by transport agent and number of packets sent by the transport agent.

$$\begin{aligned} \mathbf{N_L} = & \text{ No. of Packets Received by the receiving Transport Agent} \\ & \mathbf{-} \text{ No. of Packets Sent by the sending Transport Agent} \end{aligned} \qquad \textit{Equation 5.4}$$

Note once the traffic source agent stops generating frames the receiver transport agent is kept live for some time, so it can receive any remaining packets in the network that are still being forwarded to the end destination, in order to give an accurate measure of the number of packets lost.

Receiver Buffer Loss, RB_L, refers to packets that have arrived 'too-late' for playout.

(***1***) It is calculated by taking the difference between the network delay of each packet, N_D, and the mean network delay of all packets, $\overline{N}_D$ [55]. This is expressed in Equation 6.4 (chapter 6).

Other methods of calculating Receiver Buffer Loss are illustrated below. However, all graphs (in chapter7) are based on the first formulation (defined by Equation 6.4) and all the probability distribution graphs are based on the third formulation (defined by Equation 5.6).

(***2***) The Receiver Buffer Loss based on a 'probability distribution function' of delay, is calculated in the manner expressed in Equation 5.5. This states the number of packets that have exceeded the mean network delay in the network.

Receiver Loss Probability Estimate = $\mathrm{Pr}\{\mathrm{N_D} - \overline{\mathrm{N}}_\mathrm{D} > \mathrm{R_D}\}$ *Equation 5.5*

This is illustrated in the Figure 5.4.

(***3***) The Receiver Buffer Loss based on a 'probability distribution function' of inter-packet-gap-spacing ($\mathrm{IPGS_R}$), is calculated in the manner expressed in Equation 5.6. This states the number of packets that experience an IPGS, a time interval greater than the Playout Buffer size in seconds relative to the packet in front.

Receiver Loss Probability Estimate = $\mathrm{Pr}\left\{\mathrm{P_{IPGS(R)}} > \mathrm{R_D}\right\}$ *Equation 5.6*

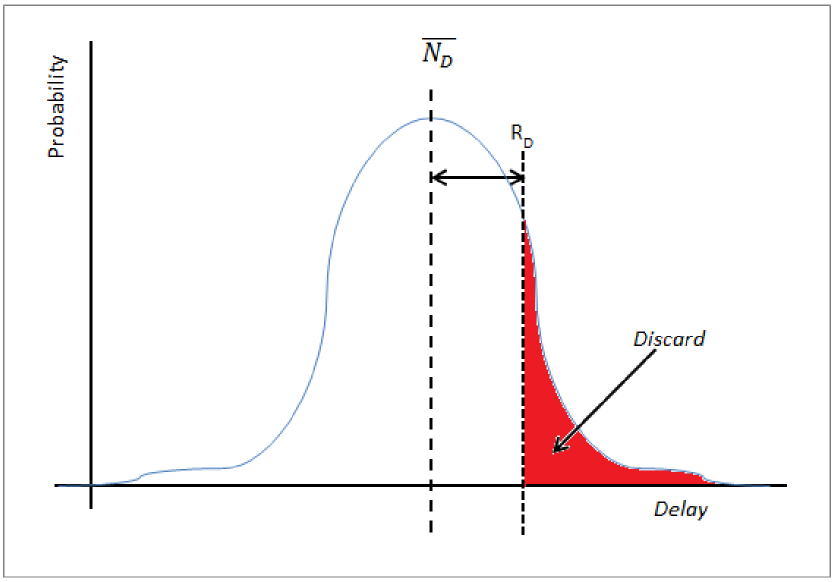


Figure 5.4, Loss Probability at Receiver/De-jitter Buffer

5.2.1.2. Delay

The total end-to-end delay experienced by an individual packet is expressed in Equation 6.10 (chapter 6), which is shown below.

Network Delay, N_D, is the sum of, G_D' propagation delay,
'Q_D' queuing delay, and 't_D' service time.

$$\mathbf{N_D = (G_D + Q_D + t_D)}$$

$$\mathbf{D_T}\text{ (ms)} = \mathbf{\left(\frac{1}{P_R} + S_D \right) + N_D + \left(R_D + \frac{1}{F_R} \right)} \qquad \textit{Equation 5.7}$$

The Sender Buffer (SB) Delay, $\mathbf{S_D}$, is estimated by monitoring the Inter-Packet-Gap-Spacing (IPGS) for each packet. If the $IPGS_S$ is equal to the Frame Interval, then the SB is considered to be empty.

If , $P_{IPGS(S)} = F_I = \frac{1}{F_R}$

Then,

$$\mathbf{S_D \text{ of Packet} = 0}$$

Equation 5.8

Otherwise, if the IPGS is larger than the Frame Interval, F_I, then the SB is considered to be fully occupied. The SB delay for each packet is calculated by monitoring the ISPS and multiplying it by the size of the buffer, B, which is predefined with a finite value.

If , $P_{IPGS(S)} > F_I$

Then,

$$S_D \text{ of Packet}_i = P_{IPGS(i)} \times B$$

Equation 5.9

This approach has been chosen for a number of reasons.

(***1***) The TFRC MAA is not standardised for *ns*2 or any other software and this is because the TFRC MAA is still currently an IETF Internet Draft.

(***2***) The Internet Draft [61] states that the user needs to decide whether to drop packets in the SB either from the tail, head or randomize the dropping. The behaviour of dropping chosen will have a different impact on the amount of delay packets will experience in the Sender Buffer. In addition, the method of dropping chosen would have a different impact on the loss impairment. A tail or head dropping method leads to consecutive losses, increasing silence periods at the receiving end, this can significantly degrade the QoE, leaving the end-user considerably dissatisfied with the service. A loss of 14% can lead to a quality impairment of ΔR=-33. Whereas, a consecutive loss of 2 packets can lead to a quality impairment of ΔR=-38, 3 packets ΔR=-57 and 4 packets R=-66 [49]. For example a consecutive packet loss equal to 3 packets degrades the overall quality to R=43 (R=100-57=43), when the minimum acceptable quality is R=60. Therefore, the impact of consecutive loss is severe, particularly when it consists of 3 packets and more. The method of loss calculation chosen underestimates the impact of loss of the Sender Buffer, as it calculates losses over the duration of the connection, ignoring consecutive losses.

(***3***) The smoothing of the IPGS using Equation 2.10 (from chapter 2) and of the 'Loss Event Rate' using Equation 2.16, reflects that the IPGS indicated will remain for some time, before it increases/decreases drastically impacting on the Packet Rate, and hence the Sender Buffer occupancy. Therefore the measured SB delay will indicate the long term behaviour of the SB occupancy.

The condition of having the IPGS lower than the Frame Interval (i.e. resulting in a Packet Rate being higher than the Frame Rate) is prohibited by the CCM. This is achieved by restricting the CCM's IPGS to the Frame Interval. Therefore, the Packet Rate cannot be faster than the Frame Rate of the encoder.

Network Delay, N_D : Taking the difference between the time the packet was received and the time it was sent at the respective transport agents gives an accurate measure of the network delay the packet experienced in the network.

$$\mathbf{N_D} = \text{Time packet Received at Receiver Transport Agent} \\ - \text{Time packet Sent by Sender Transport Agent} \qquad \textit{Equation 5.10}$$

The Playout Buffer Delay[5], R_D, is a fixed delay which each packet experiences once it has been received, in order to absorb network jitter. The PB is set to a time required to buffer 'B' packets before playout begins. The PB delay is expressed in Equation 6.5 (chapter 6), which is shown below.

5.2.1.3. Packet Size (PS) and Packet Rate (PR)

Once the 'Packet Rate Adaptation' (PRA) or the Packet Size Truncation (PST) functions are performed by either of the two CCMs (TFRC and TFGS) the behaviour is monitored at the transport agent. This records the size of the packet sent and the Inter-Sent-Packet-Spacing (ISPS). Taking the inverse of ISPS indicates the Packet Rate, P_R at which the packet are sent into the network. This is shown in Equation 2.11.

[5] Also referred to as Receiver Delay, R_D

5.2.2. Parameters Used

5.2.2.1. Traffic Source and Transport Agent

The traffic source agent parameters used in this simulation study for the voice applications are based on the MPEG-2 codec. The TFRC or TFGS CCM is attached with a traffic source agent which is configured to match MPEG-2 codec parameters: a fixed Frame Rate of 50 fps and a maximum adaptive Frame Size of 168 bytes. This is the peak voice quality, R=93.24, that can be achieved by the MPEG-2 codec.

Furthermore, adding IP, UDP and RTP headers (20, 12, 8 bytes respectively) to each packet results in a packet size of 208 bytes, reflecting a maximum bitrate of 83.2 Kbps for each traffic source agent.

The traffic flows can be considered as constant bitrate (CBR) sources when operating at their maximum bitrate, i.e. during no congestion. However, during congestion the transport agent (i.e. the CCM) will adapt the Packet Rate (for TFRC CCM) or Packet Size (for TFGS CCM). With the TFRC CCM the PRA function will increase the ISPS in order to reduce the Packet Rate whilst keeping the Packet Size fixed. This will reduce the overall bitrate. However, with the TFGS CCM the PST function will truncate the Packet Size and keep the ISPS fixed. For both the CCMs during congestion the traffic flows can be considered as Variable Bitrate (VBR) flows.

The details of the parameters of each traffic source agent and transport agent are shown in Table 5.2.

The traffic source agent (File Transfer Protocol, ‘FTP’) used for the TCP connection is a file transfer of an infinite file size. The transport agent Packet Size is equivalent to the Packet Size of the voice application, 208 bytes, in order to provide a comparative scenario with respect to the homogenous traffic mix. So both the flows can experience an equivalent fair-share of bandwidth between themselves during congestion. The transmission bitrate is controlled by the Additive Increase Multiplicative Decrease (AIMD) congestion control algorithm of the TCP transport protocol. The linkage between the transport agent, TCP, and traffic agent, FTP, is shown in Figure 5.2b.

Application Type	CCM	Voice Codec	Sender Buffer size in frames	Playout Buffer size in frames	Receiver Buffer Delay (sec)	Frame Rate of Traffic Source (fps)	Packet Rate of CCM (pps)	
						fixed	*min*	*max*
Audio	*TFRC*	*MPEG-2*	4	4	0.08	50	*unrestricted*	50
	TFGS	*MPEG-2*	0	4	0.08	50	50	50
FTP	*TCP*	*n/a*	n/a	n/a	n/a	n/a	n/a	n/a

Application Type	CCM	Frame Size[1] of Traffic Source inc. Headers[2] (bytes)	Bitrate of Traffic Source (bps)	Packet Size after Adaptation by the CCM (bytes)		Bitrate of CCM (bps)	
				min	*max*	*min*	*Max*
Audio	*TFRC*	208	83,200	208	208	*unrestricted*	83,200
	TFGS	208	83,200	41	208	16,400	83,200
FTP	*TCP*	208	*unrestricted*[3]	540	540	*Unrestricted*	

Table 5.2, Parameter values of Audio and Data sources

1 → *The Frame Size is set to 168 Bytes because this is the maximum quality (R = 93.24) that can be achieved by the MPEG-2 codec, based on the quality analysis conducted in chapter* 6.

2 → *Audio Packet Headers = IP(20) + UDP(12) + RTP(8) = 40 Bytes*

TCP Packet Headers = IP(20) + TCP(20) = 40 Bytes

3 → *An infinite file size is used for transfer*

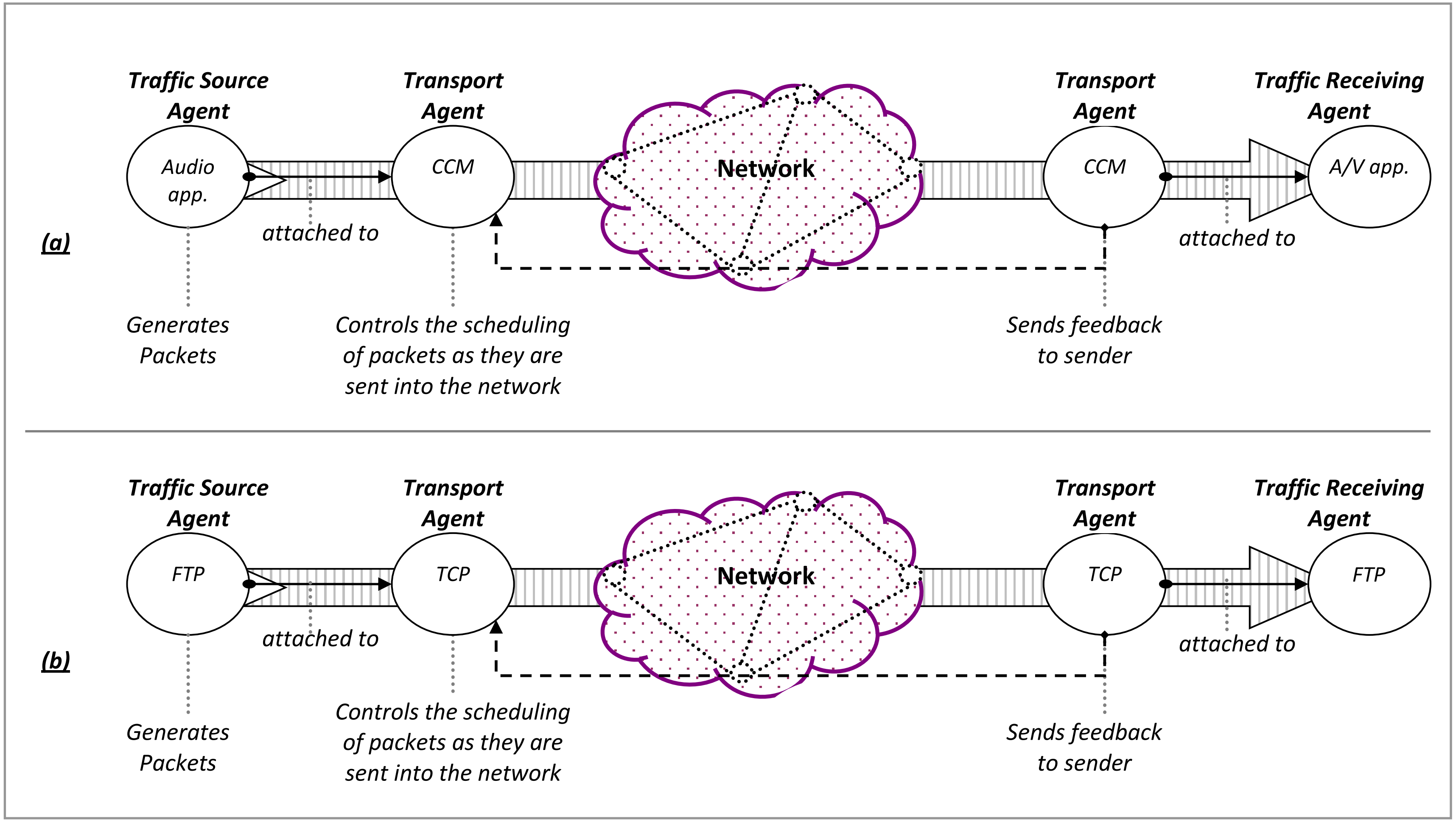


Figure 5.2, ns2 linkage between Traffic source and Transport Agent

5.2.2.2. Sender Buffer and Playout Buffer Size

The Playout Buffer is equal to 80 ms, where the buffer is set to 4 packets and the encoder frame rate is of 50 fps. A buffer of 4 packets is a typical configuration for voice application Playout Buffer's [15, 62] and the same size is used for the Sender Buffer, in order to maintain consistency, and this was seen in [6] also.

$$\mathbf{pb_d} = \frac{1}{FR} \cdot B \qquad \textit{from } \text{Equation 6.5}$$

$$= \frac{1}{50} \cdot 4$$

$$= \mathbf{80\ ms}$$

5.2.2.3. Network Delay, N_d

The network delay, N_d, is constraint to 200 ms and this is achieved by making use of Random Early Discard (RED) Active Queue Management (AQM). Further details on RED AQM configuration are given in section 5.2.3.2. An upper limit on network delay of 200 ms is of a reasonable constraint for interactive voice applications as stated by [63, 64]. A hard limit on the total end-to-end delay, D_T, of 400 ms can still maintain the minimum acceptable quality at R=60 (see Table 3.1), excluding the Sender Buffer Delay component and excluding any loss end-to-end, (see Figure 2.3).

The total end-to-end delay is expressed in Equation 5.11, which is shown below.

$$\mathbf{D_T\ (ms) = \left(\frac{1}{P_R} + S_D\right) + N_D + \left(R_D + \frac{1}{F_R}\right)}$$ *from* Equation 5.7

$$\mathbf{= \left(\frac{1}{P_R} + S_D\right) + 200 + \left(\frac{1}{F_R} \cdot B + \frac{1}{F_R}\right)}$$

Frame Rate (F_R) = 50 fps, Buffer Size (B) = 4

$$\mathbf{400\ ms > \left(\frac{1}{P_R} + S_D\right) + 200 + \left(\frac{1}{50} \cdot 4 + \frac{1}{50}\right)}$$

$$\mathbf{400\ ms > \left(\frac{1}{P_R} + S_D\right) + 300}$$ *Equation 5.11*

5.2.3. Simulation Design: Network Scenario Description

5.2.3.1. Topology Framework

Below, a network topology is used which illustrates the performance of the voice flows when using either the TFRC or TFGS congestion control mechanism (CCM). It consists of a single bottleneck. When voice flows are configured using the same CCM (either TFRC of TFGS), this is referred to as a homogenous traffic scenario. Voice flows competing against TCP traffic, are referred to as a heterogeneous traffic scenario. Table 5.3, Table 5.4, and Figure 5.5 summarise and illustrate how the simulation study is structured with respect to the traffic mixes.

Simulation Study 1 - Homogenous Traffic Scenario							
Simulation Set	**Flow (Application) Type**	**Codec Type**	**Duration (secs)**	**CCM**	**No. of Flows**	**Capacity, C, (bps)**	**RED Queue Parameters (Min / Max) in packets**
A	Voice	MPEG-2	60	TFRC	X	499,200	20 / 60
B	Voice	MPEG-2	60	TFGS	X	499,200	20 / 60

Table 5.3, Homogenous Traffic Scenario

Simulation Study 2 - Heterogeneous Traffic Scenario							
Simulation Set	Flow (Application) Type	Codec Type	Duration (secs)	CCM	No. of Flows	Capacity, C, (bps)	RED Queue Parameters (Min / Max) in packets
A	Voice	MPEG-2	60	TFRC	X	998,400	40 / 120
	FTP	n/a	60	TCP	X	998,400	40 / 120
B	Voice	MPEG-2	60	TFGS	X	998,400	40 / 120
	FTP	n/a	60	TCP	X	998,400	40 / 120

Table 5.4, Heterogeneous Traffic Scenario

In the Heterogeneous traffic mix the bottleneck capacity and queue configurations are twice that of the Homogeneous traffic mix. This increase caters for the extra TCP flows added. These adjustment parameters result in an equivalent network scenario when comparing the two: Homogeneous and Heterogeneous traffic mix scenarios.

The purpose of testing the CCM in both a homogeneous and heterogeneous traffic mix is to illustrate fairness across similar flows and other adaptive flows such as TCP.

The network topology used (commonly known as the dumbbell topology) is characteristic of the best-effort Internet, where all types of flows are converged over one link. Many other researchers have used a similar approach in testing and comparing their CCM, examples can be found in [3, 6, 9, 65, 66, 89]. Figure 5.5 illustrates the structure of the dumbbell topology.

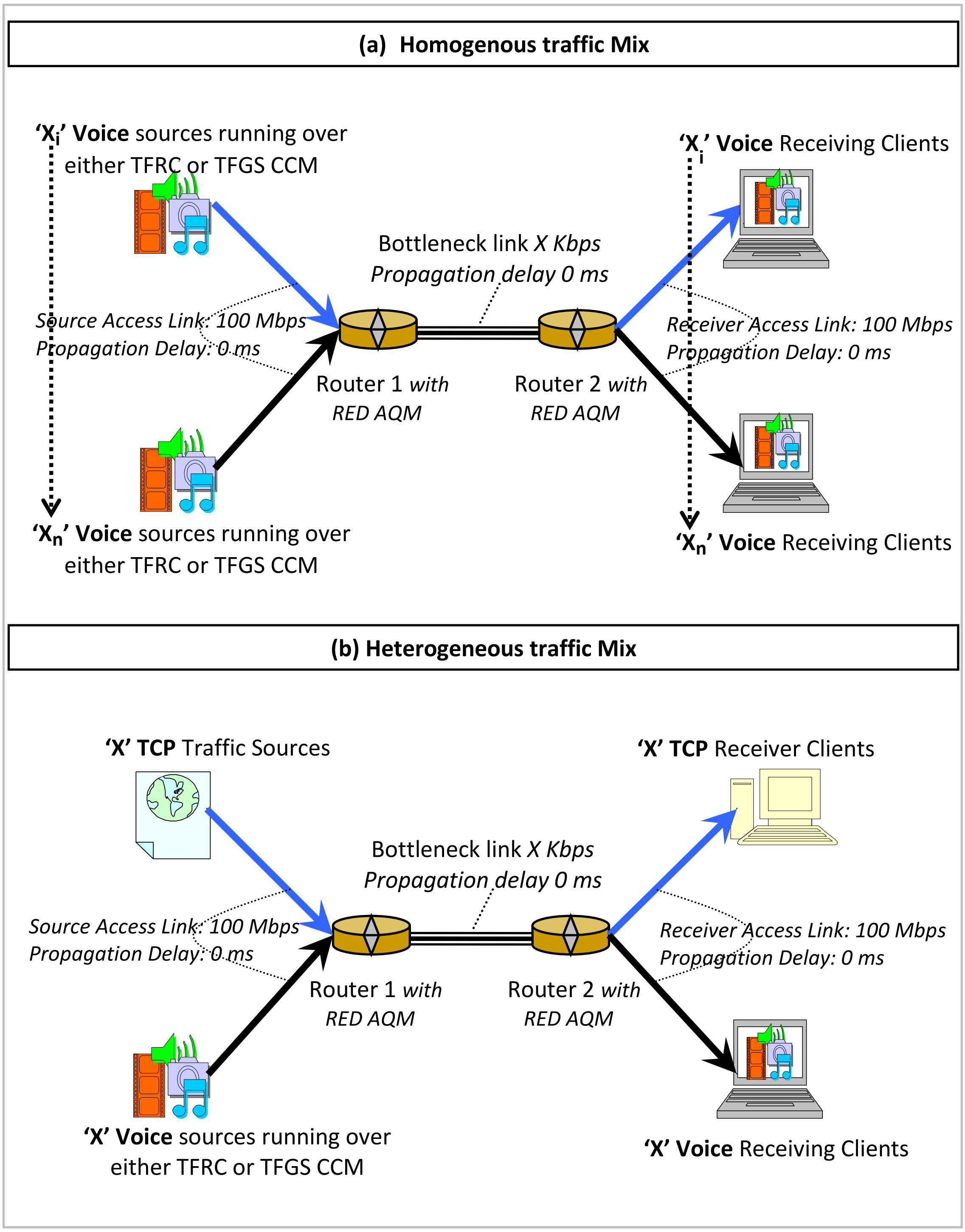


Figure 5.5, 'Network Topology Setup'

5.2.3.2. Queue Configuration and Parameters, and Link Capacity

The bottleneck link is configured with Random Early Discard (RED) Active Queue Management (AQM) [26, 65]. The reasons for choosing RED over droptail is:

a) to avoid global synchronisation of flows as defined in [65],

b) to achieve a fair share of dropping across flows,

c) to maintain control on the Queue size in order to reduce network delay variation,

d) to avoid developing full queues.

The benefits of using RED for TFRC CCM have been reported in [9].

The minimum threshold (Q_{min}) and maximum threshold (Q_{max}) of RED AQM are set in a manner that insures that network delay, N_D, should not try to exceed 200 ms in order to satisfy the hard limit of 400 ms for the total end-to-end delay (D_T) as defined in the previous section 5.2.2.3. The approximate maximum delay a packet may experience in the network is defined as:

$$N_D = \frac{P_{S(mean)}}{C} \cdot Q_S$$

Equation 5.12

C – Capacity

Q_S – Queue Size

To set the size of the queue, the above equation can be rearranged to:

$$Q_S = \frac{N_D}{P_{S(mean)}} \cdot C$$

Equation 5.13

Homogenous Traffic mix Parameter Settings

For a comparison study which is made up of a homogenous traffic mix where the bottleneck capacity (C) is set to 499,200 bps, this results in a recommended queue size of 60 packets. This value gives an indication of what the minimum and maximum thresholds should be when configuring the RED AQM.

[67] recommends setting the minimum threshold of RED to a third of the size of the maximum threshold (measured in packets) when flows in the network consist of TCP (or alike such as TCP-friendly).

$$\mathrm{Q_{max}} = 3 \cdot \mathrm{Q_{min}} \qquad \textit{Equation 5.14}$$

This configuration gives room for the TCP to adapt and avoids the saw-tooth behaviour where the TCP is frequently going into slow-start after aggressive dropping where the RED AQM is trying to maintain the queue size within the tight thresholds (minimum and maximum). Hence, this configuration of setting the maximum threshold 3 times the size of minimum threshold, keeps the TCP connections in congestion avoidance phase, where the TCP sender is adjusting its window size occasionally based on network conditions, rather than drastically reducing its window size in the case of slow-start.

Therefore, the values chosen here are of 20 and 60 packets respectively. Table 5.5 summarises the chosen parameter values for a homogenous traffic mix network scenario, where a maximum threshold of 60 packets reflects a maximum delay queue delay of 200 ms.

Homogenous Traffic Mix	
Bottleneck Bandwidth, (C), bps	499,200
Mean Packet Size (Bytes)	208
RED AQM, Q_{min} / Q_{max} (in packets)	20 / 60
RED AQM, Q_{min} / Q_{max} (in bytes)	4160 / 12,480
min/max Network delay in msecs *(based on mean packet size)*	66 / 200

Table 5.5, Queue Configuration for Homogenous Traffic Mix

Note:

Threshold in Bytes = Threshold in Packets x Mean Packet Size *Equation 5.15*

Further Details of RED Configurations in Byte Mode can be found in [68]

Heterogeneous Traffic mix Parameter Settings

In this comparative study a heterogeneous traffic mix of voice and TCP flows are competing for bandwidth. The bottleneck capacity (C) is set to 998,400 bps. This results in a recommended maximum queue size of 120 packets, calculated by Equation 5.13. The parameters set in the Heterogeneous traffic mix scenario are of such value that they provide a comparative scenario with respect to Homogenous traffic mix. The addition of TCP traffic increases the total number of flows in the network by 2. Therefore, the bandwidth is doubled with respect to the bandwidth in the Homogenous traffic mix. The increase in bandwidth by 2 results in an increase of the queue minimum and maximum thresholds by 2.

Table 5.6 summarises the chosen parameter values for a heterogeneous traffic mix network scenario, where a maximum threshold of 120 packets reflects a maximum delay queue delay of 200 ms.

Heterogeneous Traffic Mix	
Bottleneck Bandwidth, (C), bps	998,400
Mean Packet Size (Bytes)	208
RED AQM, Q_{min} / Q_{max} (in packets)	40 / 120
RED AQM, Q_{min} / Q_{max} (in bytes)	8320 / 24,960
min/max Network delay in msecs *(based on mean packet size)*	66 / 200

Table 5.6, Queue Configuration for Heterogeneous Traffic Mix

5.2.3.3. Congestion Level Environment

The two CCMs are tested in a range of congestion levels by increasing the number of flows, X, whilst keeping the same bandwidth constraint; the objective is to show the rate at which the quality of the voice call degrades with respect to R-value.

5.2.3.4. Call Generation

The voice streams are of a real-time interactive nature. The communication is unicast, i.e. between two users. Note that although voice calls typically transmit data in both directions, transmission in the two directions is logically independent, hence the simulation study here shows the performance in the context of one end-user i.e. simulated in one direction, from sender to the receiver.

Because the traffic sources are real-time interactive voice in nature, they impose a minimum and maximum limit on their bitrate. Therefore, these traffic sources should not be considered equivalent to music downloads, where the application sends data at the maximum available bitrate in the network. (Note: music downloads can be considered as file transfers, hence requiring no strict loss, delay and jitter requirements. They will adopt a 'best-effort' strategy).

Voice conversations can go idle for some time (at least in one direction while one user listens to what the other user has to say). This can be problematic for the CCM (both TFRC and TFGS) because it is designed to resume its sending bitrate after an idle period at a rate of 2 packets per Round-Trip-Time (RTT) and doubling every RTT until its previous rate is achieved. This is done in order to emulate the slow-start behaviour of the TCP congestion control. In order to avoid this problem the traffic sources are configured as long-lived continuous sources, which don't go idle.

The traffic sources (both voice and FTP) are of 60 seconds in duration, giving reasonable time to observe the long-term behaviour of the QoE and fairness measured. The traffic sources in [9] were of similar duration when TCP fairness was measured.

5.2.3.5. Simulation Runs

Each simulation set is repeated 25 times with a randomised seed to give a reasonable measure of the mean, standard deviation (STDEV), and confidence interval (CI) for the variables measured from the simulator.

The mean is calculated across all the simulation sets for each flow, which is referred as the 'Batch Mean'. Details can be found in Appendix III, Simulation Runs.

5.3. Validation of TFGS CCM's PST operation

This section validates that the code written for the TFGS CCM in '*ns2*' operates correctly in the manner described in section 5.1.

5.3.1. Network Scenario Description

The network scenario description is defined as in Table 5.3, further details can be found in section 5.2.3.

5.3.2. Results: (i) Non-Congested State

Over a bottleneck capacity of 499,200 bps only 2 voice flows are competing among themselves, either configured with TFRC or TFGS CCM. The maximum bitrate the flows can generate is of 83.2 kbps each, as shown below. Simulation results confirmed that in non-congested periods the voice flows bitrate is equivalent to that of the maximum bitrate.

$T_{ETR} = 8 \cdot P_{S(max)} \text{ x } P_R$

$= 8(168 + 40) \text{ x } 50$

Note: a) 40 Bytes for headers, b) Multiply by 8 to convert into bits

$= 83.2$ kbps

This gives a total bandwidth occupancy of 166.4 kbps leaving 332.8 kbps of capacity unused. The results illustrated that the flows are operating at a Packet Size of 208 bytes (including headers of 40 bytes) and at a Packet Rate of 50 pps.

The Packet Size remains fixed throughout the duration of the call, because there is no congestion in the network. Additionally, the Inter-Packet-Gap-Spacing (IPGS) remains fixed at 20 ms, which results in a Packet Rate of 50 pps $\left(P_R = {}^{1}/_{P_{IPGS}}\right)$, equivalent to the Frame Rate of 50 fps.

5.3.3. Results: (ii) Congested State

Over a bottleneck capacity of 499,200 bps when 8 voice flows are competing between themselves, and where the voice flows are configured with either TFRC or TFGS CCM, this equates to a 'fair-share' (FS) of 62.4 Kbps of bandwidth to each flow as calculated using

Equation 5.16. Figure 5.6 illustrates 1 flow of a total 8 for each of the CCMs. The figures give evidence that the Sending Rate (SR) varies around the 'fair-share' value for both the CCM.

$$Fair\ Share\ Sending\ Rate = \frac{Bottleneck\ Capacity}{Number\ of\ Flows}$$

$$\mathrm{T_{FS}} = \frac{\mathrm{C}}{n_{flows}}$$ Equation 5.16

The difference seen in the variation of the Sending Rate when comparing the two CCMs is because the 8 flows are multiplexed at different times where each flow's start time is randomised therefore, resulting in different 'loss event rates' (LER). The LER and SR graphs look like mirror images of each one, see Figure 5.7, and Figure 5.6 respectively.

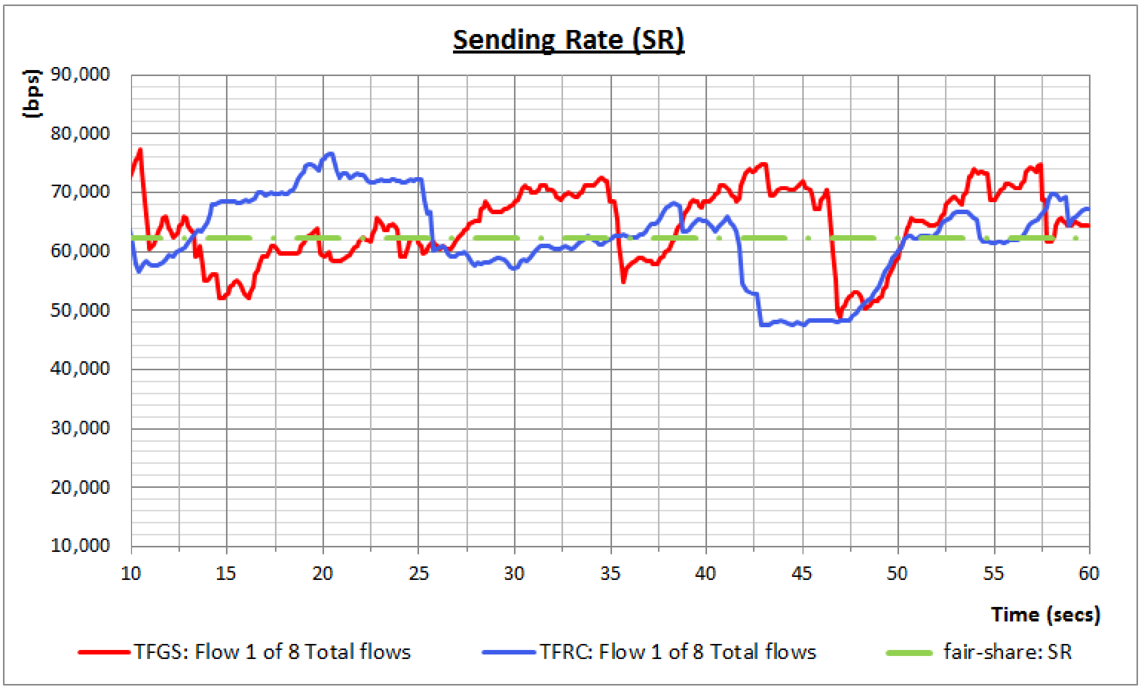


Figure 5.6, Sending Rate

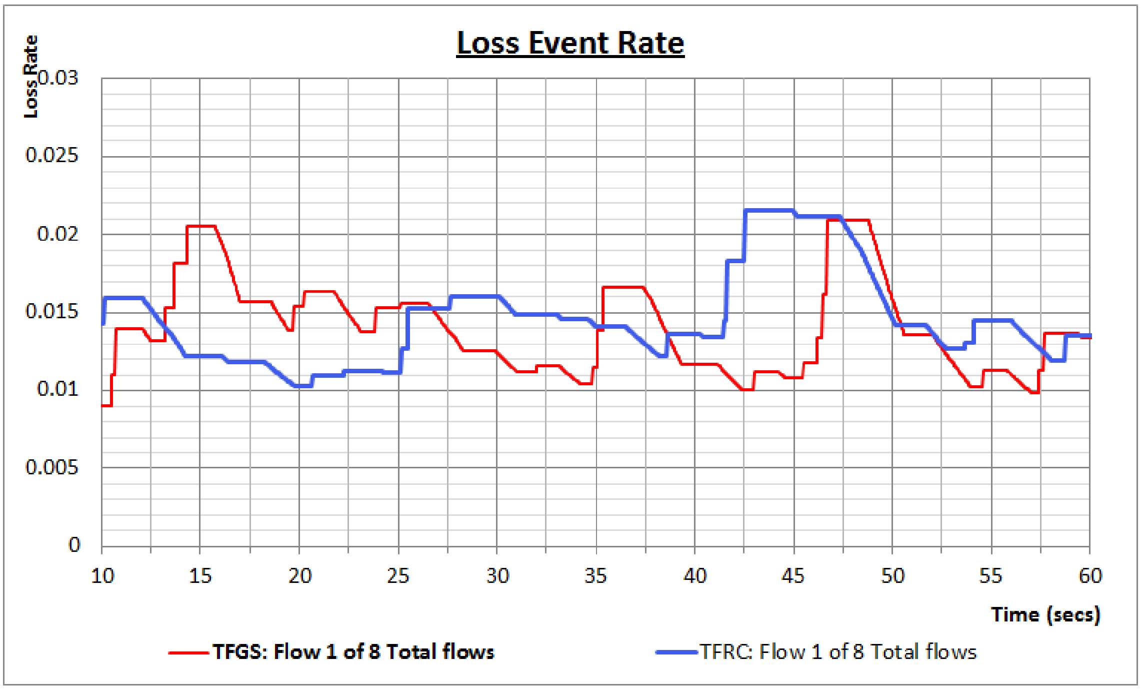


Figure 5.7, Loss Event Rate

The response to congestion for the TFRC CCM is to increase the Inter-Packet-Gap-Spacing ($IPGS_S$), i.e. resulting in a reduced Packet Rate, whereas in the case of the TFGS CCM it truncates the Packet Size (PS). This is illustrated in Figure 5.8 and Figure 5.9 respectively.

$$\mathrm{T_{FS}} = \frac{\mathrm{C}}{n_{flows}}$$ *from* Equation 5.16

$$T_{FS}\ (bps) = 8 \cdot P_{S(FS)}\ \times P_{R(FS)}$$

Fair Share Packet Size, $P_{S(FS)}$:

$$\mathrm{P_{S(FS)}} = \frac{\mathrm{T_{FS}}}{8 \cdot \mathrm{P_{R(FS)}}}$$ Equation 5.17

Fair Share Packet Rate, $P_{R(FS)}$:

$$\mathrm{P_{R(FS)}} = \frac{\mathrm{T_{FS}}}{8 \cdot \mathrm{P_{S(FS)}}}$$ Equation 5.18

$$\mathrm{P_{IPGS}} = \frac{1}{\mathrm{P_R}}$$

Equation 5.19

Fair Share Inter-Packet-Gap-Spacing, ($P_{IPGS(FS)}$) :

$$\mathrm{P_{IPGS(FS)}} = \frac{8 \cdot \mathrm{P_{R(FS)}}}{\mathrm{T_{FS}}}$$

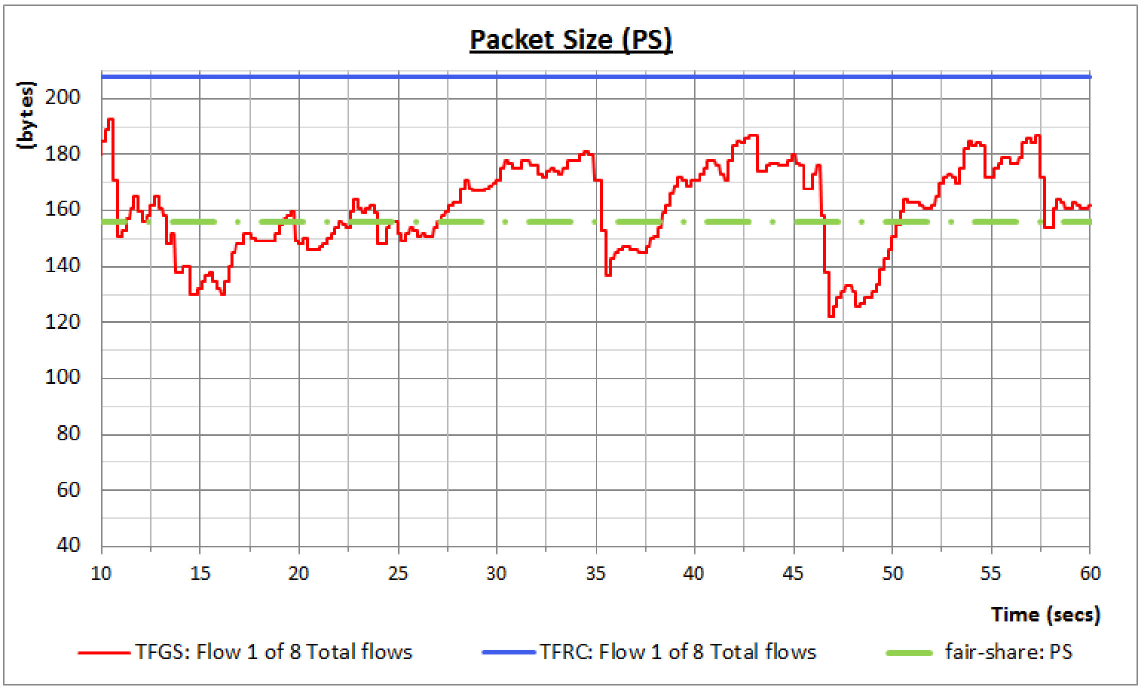


Figure 5.8, Packet Size

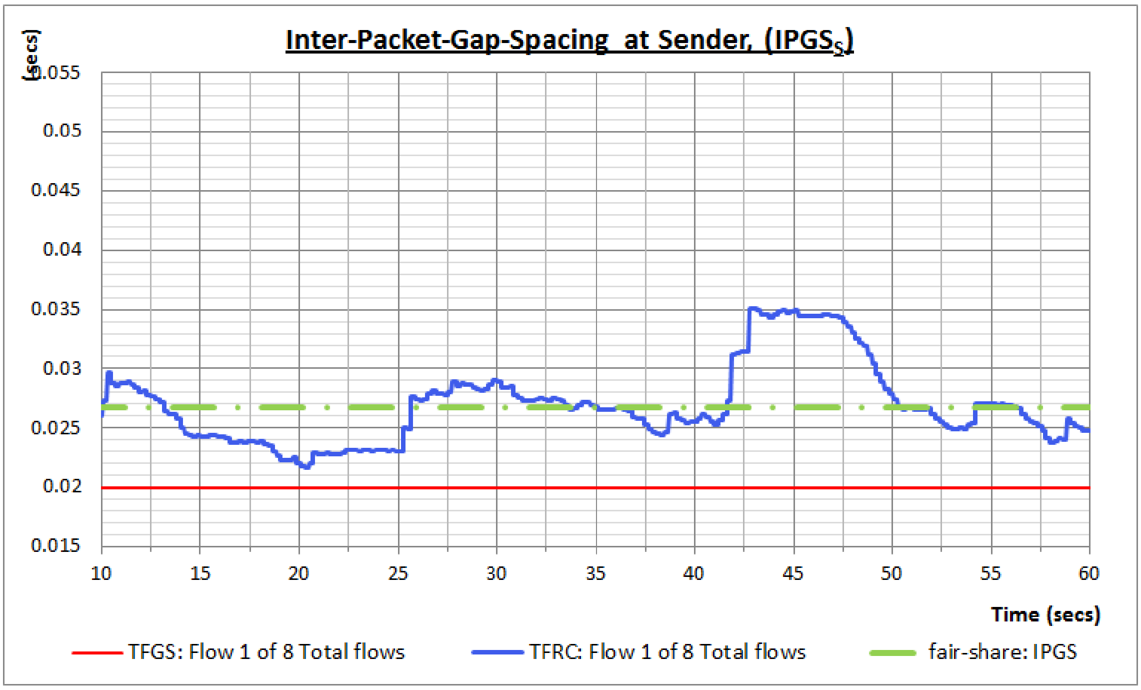


Figure 5.9, Inter Packet Gap Spacing at Sender, (IPGS$_S$)

The results show that the implemented PST function of the TFGS CCM in *ns*2 operates successfully. The CCM responds to congestion by varying the Packet Size and aims to achieve a 'fair-share' of network resources across all flows.

This simulation of 8 voice flows was repeated 25 times with a randomised seed to calculate the overall Batch Mean, $\hat{X}$, from 25 individual Batch Means, X_Y, of each simulation run. These batch means were used to calculate:

a) The standard deviation (STDEV), i.e. a measure of how far all the batch means (of each simulation run) deviate from the overall batch mean of all Simulation Runs, $\hat{X}$.
b) Confidence interval (CI), i.e. a measure of what percentage of the data set is within a given distance from the overall batch mean, $\hat{X}$.

The standard deviation (STDEV) and confidence interval (CI) across the 25 batch means for 8 voice flows were within a tight range. For example, the batch mean for the 'inter-packet-gap-spacing' (IPGS) at the sender was 2.56×10^{-2} secs (25.6 ms), for flows running over the TFRC CCM. The standard deviation and (95%) confidence interval was 8.4×10^{-5} and 3.3×10^{-5} respectively. Further details can be found in Table 11.1 and Table 11.2 (of Appendix III, Simulation Runs) for flows operating over either TFRC or TFGS CCM, respectively.

6. Performance Evaluation Methodology

This chapter provides a measurement toolset that enables the end-user to quantify the level of quality degradation from packet loss, delay, delay-jitter, and codec impairment. Packet loss, delay, and delay-jitter can occur at the end-system and network, and this is referred to as 'Packet Level Impairment'. Using Adaptive coding (which reduces bitrate by truncating Frame Size) has a proportional effect on the quality of the multimedia connection. This is referred to as 'Codec Impairment'. The truncation of Frame Size is achieved by reducing the frame size in bytes, therefore in the text below 'Codec Impairment' may also be referred to as 'Byte Level Impairment'.

Using the E-model, the Packet and Byte Level impartments can be subtracted in a scalar manner from the original quality value using an analytical expression. This enables the quantification of the QoE of a voice connection when it is running over either of the two CCM (TFRC or TFGS).

The last section in this chapter will give a quantitative comparison between the two CCMs. It will illustrate how a reduced Packet Rate (a mode of operation used by the TFRC CCM) that introduces 'Packet Level Impairment' arising from Sender Buffer delay and loss has an impact on the QoE. In comparison to how a Frame Size truncation (a mode of operation used by the TFGS CCM) that introduces 'Byte Level Impairment', has an impact on QoE.

These two forms of congestion response (PR reduction and Frame Size truncation) introduce packet and byte level impairment, respectively, because of the method of congestion response. This is quantified using the E-model and the results will illustrate the rate of quality degradation, measured in R-value on a scale of 0 to 100.

The analysis conducted in this chapter is focused on the impairment arising from the method of congestion response used by the CCM, i.e. how a reduced available network bitrate can have an impact on the sender side of the multimedia connection. This can result in buffering of frames which introduces 'Packet Level Impairment' or Frame Size truncation which introduces 'Byte Level Impairment'.

This can be referred to as 'sender side impairment' of the multimedia connection, excluding the network and receiver impairment. However, these two latter impairments are examined in chapter 7, where a simulation study is conducted, illustrating how both the CCM perform in various network congestion environments. The analysis will highlight the impairment caused by network loss and delay, and receiver loss due to packets arriving 'too-late'. The overall performance analysis in the simulation study will embrace end-to-end impairment, including sender side, network, and receiver impairment.

6.1. Packet Level Impairment

This section illustrates the components involved at end-to-end which contribute to the overall 'Packet Level Impairment'. This consists of packet delays and losses at the end-system (sender and receiver) and during transport (i.e. within the network).

6.1.1. *Packet Delay*

The total end-to-end delay, D_T, (also known as mouth-to-ear) is expressed by Equation 2.3 (in chapter 2) which is shown below.

$$\mathbf{D_T = (E_D + Z_D + S_D) + (G_D + Q_D + t_D) + (R_D + Z_{D(R)} + E_{D(d)})} \qquad \textit{from } \text{Equation 2.3}$$

$\mathbf{Z_D} = P_{IPGS(s)} = \frac{1}{P_R}$ *from* Equation 4.7

Therefore,

$\mathbf{D_T}$ *(ms)* $\mathbf{= (E_D + P_{IPGS(s)} + S_D) + (G_D + Q_D + t_D) + (R_D + Z_{D(R)} + E_{D(d)})}$ Equation 6.1

The TFRC CCM schedules packets at a Packet Rate different from the Frame Rate of the encoder. A lower Packet Rate of the Congestion Control Mechanism (CCM) causes packets to be temporarily buffered, introducing waiting delay. This increases the probability of packets arriving 'too-late'. Furthermore, as the Packet Rate (P_R) reduces, this increases the Inter-Packet-Gap-Spacing (IPGS) between the packets sent ($P_{IPGS(S)} = {}^{1}/_{PR}$), as illustrated in Figure 6.1. The increase in $IPGS_S$ of packets sent has a proportional increase in the $IPGS_R$ of received packets. A larger $IPGS_R$ of received packets demands for a larger de-jitter buffer (Receiver Buffer) to avoid packets being discarded as they have arrived 'too-late' in time sequence. Such a scheme can reduce the interactivity of the multimedia stream if the end-to-end delay becomes too large.

The novel TFGS Multimedia Adaptation Architecture (MAA) does not use a Sender Buffer (SB). Therefore, the SB delay component is removed. Hence the total end-to-end is reduced to:

$\mathbf{D_T(TFGS)}$ *(ms)* $\mathbf{= (E_D + P_{IPGS(s)}) + (G_D + Q_D + t_D) + (R_D + Z_{D(R)} + E_{D(d)})}$ *Equation 6.2*

The TFGS CCM schedules packets at the same rate of the encoder therefore the $P_{IPGS(S)} = F_I$, as illustrated in Figure 6.2. This configuration removes SB delay, SB loss, and removes any additional packetization delay, because this delay is fixed to the $IPGS_S$.

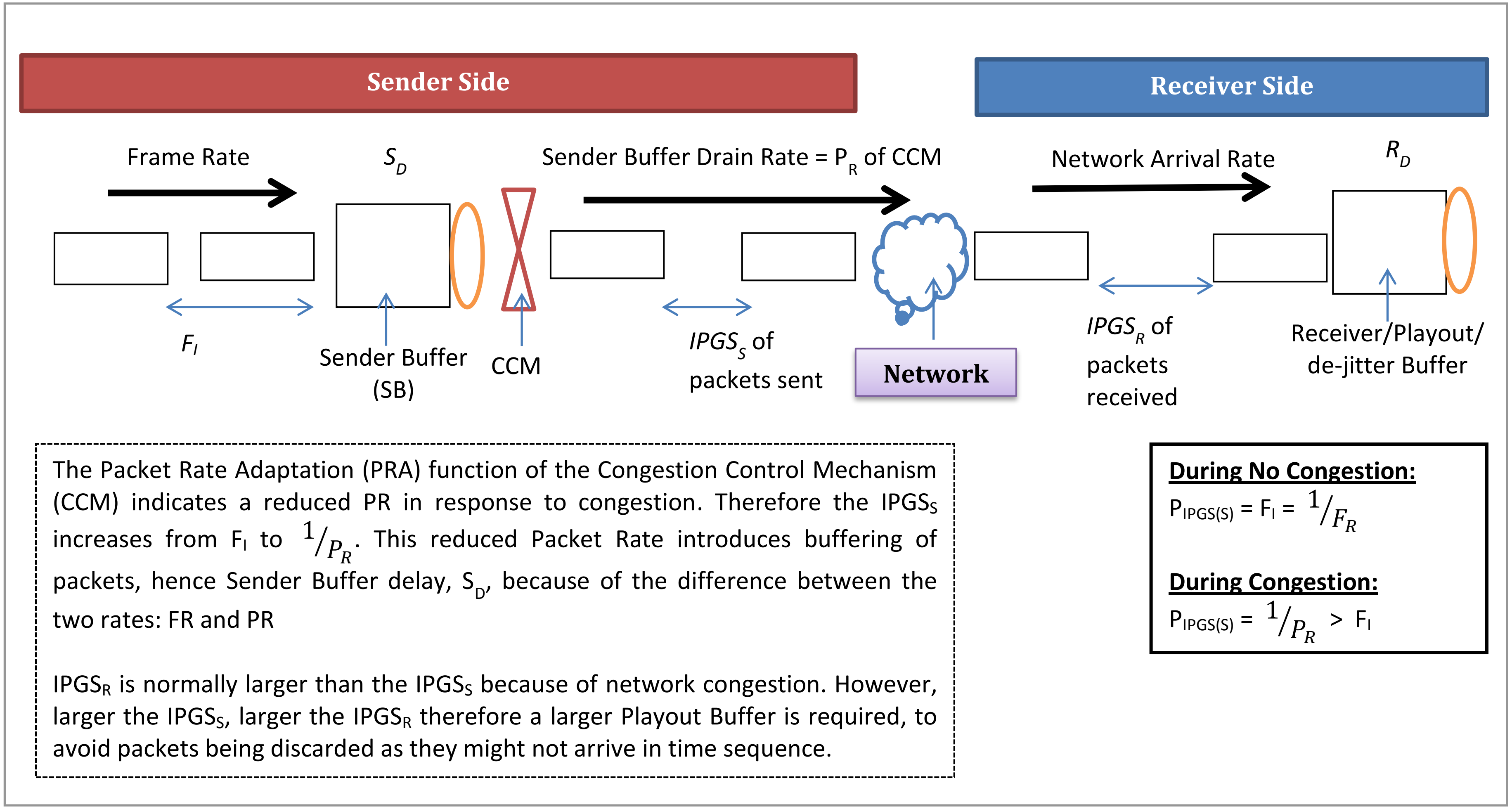


Figure 6.1, TFRC MAA

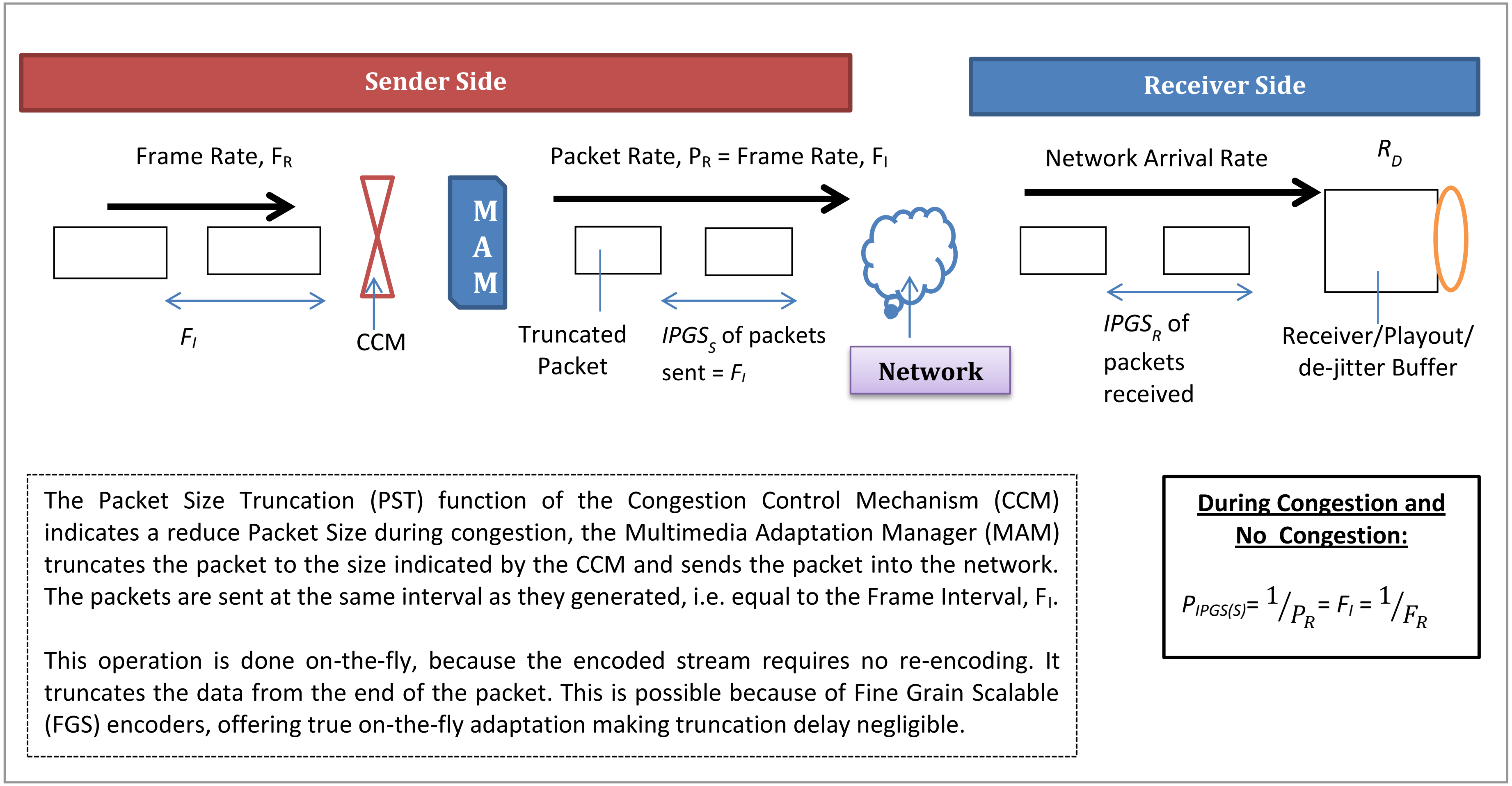


Figure 6.2, TFGS MAA

However, the novel TFGS MAA enables packets to be scheduled at the same rate at which they are generated. Hence, packets experience no delay at the sender. This results in a reduced end-to-end delay for each packet. Such an approach improves the interactivity of the multimedia. Real time interactive applications are time critical in nature, therefore minimising delay and reducing consecutive losses (when SB is full for example) results in better interactivity, and improves the perceived quality of the stream.

6.1.2. Packet Loss

The total loss ratio, $T_{L(R)}$, is expressed by Equation 2.2.

In the case of the TFRC Multimedia Adaptation Architecture, which implements a discard threshold at the Sender Buffer (SB), frames are discarded, if the sender buffer is full. Once the SB is full the multimedia connection will experience consecutive losses. Such losses can greatly make loss recovery difficult for Forward Error Correction (FEC) schemes. Consecutive losses results in silence periods in a voice call making the conversation unintelligible.

In comparison the TFGS Multimedia Adaptation Architecture, does not have a Sender Buffer, hence it does not operate in a manner which requires packets to be discarded. However, packets may be truncated during periods of network congestion; this degrades the quality of the frame, but packets are not dropped. Therefore, the loss from the Sender Buffer, 'S_L' is completely removed, hence the total loss, 'L_T', experienced by the multimedia stream is reduced to network loss, 'N_L', and loss of packets arriving 'too-late' at the receiver[6], 'R_L'.

[6] *Playout Buffer loss, P_L, is also referred to as Receiver Buffer loss, R_L, in this text.*

$$\mathbf{L_T = N_L + R_L}$$ *Equation 6.3*

Note that a lost packet is different from a loss of bytes in a packet. Truncation (loss of bytes) of a packet can still convey useful information to the end user, whereas a lost packet has nothing to convey. The impact of loss of byte is illustrated in the 'Byte Level Impairment' section in this chapter, which quantifies the impact on Quality of Experience (QoE) of byte loss from a frame.

Receiver Buffer loss, R_L, refers to packets that have arrived 'too-late' for playout. One method[7] to estimate this loss is to take the difference between the network delay of each packet, N_D, and the mean network delay of all packets, $\overline{N}_D$ [55]. This is shown in Equation 6.4. If the difference is greater than the size of the playout buffer then count it as a lost packet.

$$\delta = N_D - \overline{N}_D$$

If $\delta > R_D$ (measured in seconds) ;

Then count as packet loss. *Equation 6.4*

Note:

$$R_D = \frac{1}{F_R} \times \mathrm{B}$$ *Equation 6.5*

Where 'R_D' is referred to as Receiver Buffer delay and also known as Playout Buffer delay.

6.2. Byte Level Impairment (Codec Impairment)

The previous section looked at 'Packet Level Impairment' arising from losses and delays which occur at the end-system and during transport (i.e. in the network). This

[7] Other methods to calculate RB loss can be found in section 5.2.1.1

section looks at the impact of adaptive voice encoders that reduce their bitrate by truncating the Frame Size. This Frame Size truncation is usually required to match the available network bitrate. This form of impairment is referred to as 'Byte Level Impairment' or 'Codec Impairment'.

As discussed in chapter 3, there are three main schemes for adaptive encoding: Layered, Coarse-Grain, and Fine Grain Scalable (FGS), where FGS offers a higher degree of scalability compared to the other two. Although this thesis proposes to use the adaptive FGS encoding (which offers scalability at byte-level increments) at the time of writing the FGS codec has not undergone subjective listening tests, therefore a standard measure of its quality is not available at its respective frame sizes.

Therefore, an adaptive voice codec known as MPEG-2 is chosen to illustrate the impact of 'Byte Level Impairment'. MPEG-2 offers coarse-grain scalability to those multimedia applications that use this encoding technique.

The performance evaluation of the MPEG audio encoder has been investigated by [56] with respect to Mean Opinion Score (MOS). This shows what the MOS value is at the respective Encoder Bitrate. The results are shown in the
Table 6.1.

These results can be translated in to a Quality of Experience (QoE) assessment model known as the E-model. Using Equation 2.5 the MOS values are converted into R-values (a unit of measurement defined by the E-model to express perceived quality on a scale of 0 to 100), as shown in
Table 6.1 and illustrated in Figure 6.3.

Encoder Bitrate (bps)	MOS [0-4.5]	R-value [0-100]
19,200	2.55	49.52
28,800	3.23	62.53
38,400	3.92	77.35
48,000	4.15	83.53
57,600	4.30	88.48
67,200	4.41	93.24
76,800	4.35	90.46

Table 6.1, Quality value at relevent Encoder Bitrates

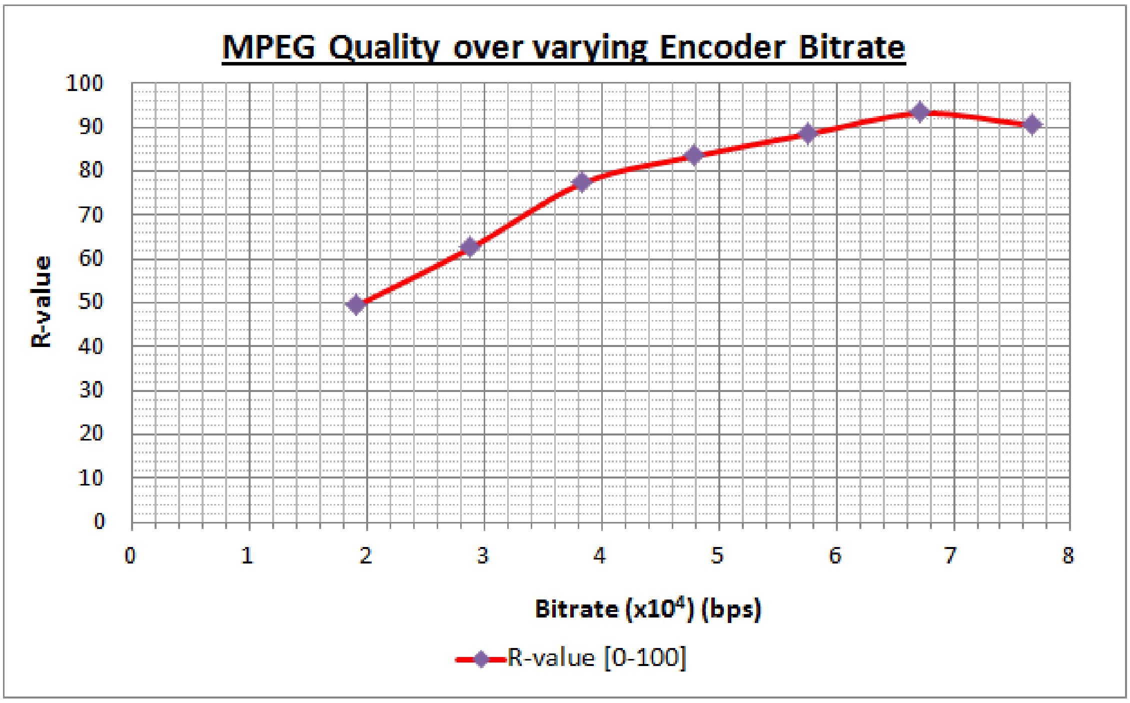


Figure 6.3, MPEG Bitrate v.s. R-value

The respective R-value for the corresponding encoder bitrate is shown in

Table 6.1, however to address the impact of quality degradation in the form of frame size truncation. The encoder bitrate needs to be expressed in Frame Size, F_S, and this can be achieved by using Equation 3.2. Where the Encoder Bitrate, T_{EB}, is divided by its Frame Rate, F_R.

$$F_S = \frac{T_{EB}}{8 \cdot F_R}$$ *Equation 6.6*

As common to most voice encoders such as G.711 and G.729 they generate frames at fixed Frame Rates of 50 fps. The corresponding Frame Size calculated using Equation 6.6 at the relevant encoder bitrates is shown in

Table 6.2. Figure 6.4 illustrates the corresponding quality value at the relevant Frame Sizes. This enables to quantify the level of quality degradation when the Frame Size is truncated in response to network congestion. With different levels of Frame Size truncation, Equation 6.7 can quantify the quality at different frame byte sizes.

Encoder Bitrate (bps)	**MOS [0-4.5]**	**R-value [0-100]**	**Frame Size (bytes) @ Frame Rate 50 fps**
19,200	2.55	49.52	48
28,800	3.23	62.53	72
38,400	3.92	77.35	96
48,000	4.15	83.53	120
57,600	4.30	88.48	144
67,200	4.41	93.24	168
76,800	4.35	90.46	192

Table 6.2, Quality value at relevent Frame Size

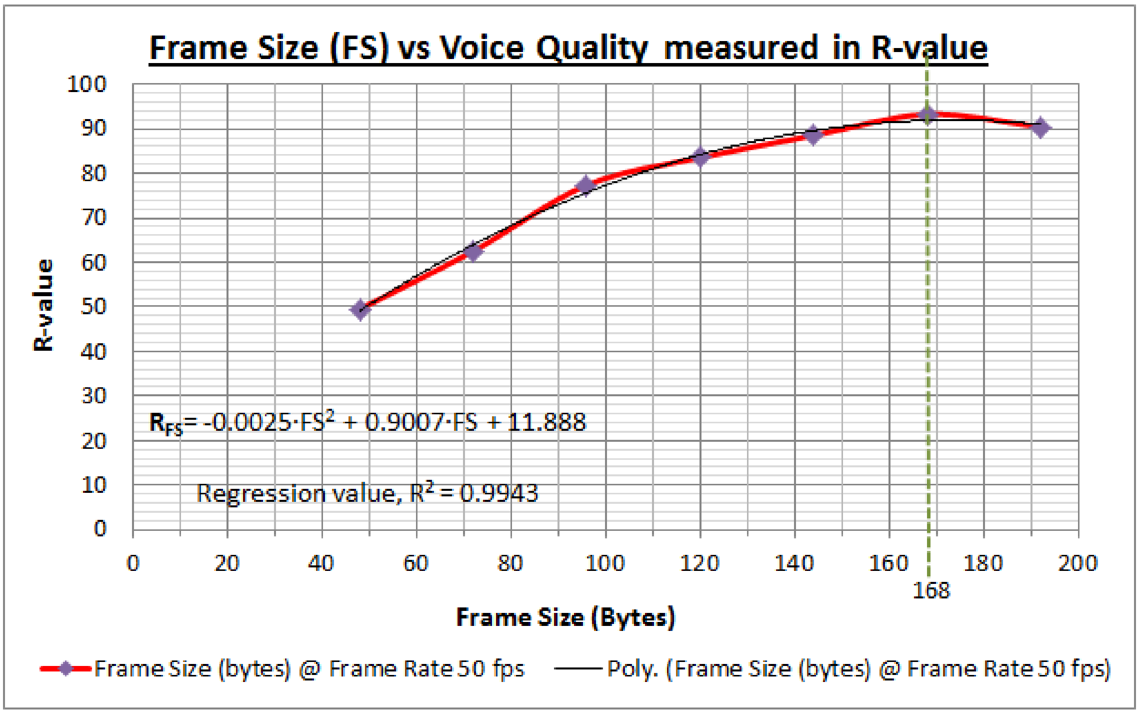


Figure 6.4, Frame Size v.s. R-value

Figure 6.4 shows that the maximum the quality that can be achieved by the MPEG audio codec is of 93.24 at a Frame Size of 168 Bytes, and after that size the voice quality starts to level out as Frame Size continues to increase. However, a Frame Size lower than 168 bytes illustrates a trend expressed in the form of Equation 6.7.

$$\text{R-value}_{FS} = -0.0025 \cdot F_S^{\ 2} + 0.9007 \cdot F_S + 11.888 \qquad \textit{Equation 6.7}$$

R-value$_{FS}$ *is a line of best-fit of the trend shown for R-value vs. Frame Size. This 'line of best-fit' was achieved using the built-in function found in Microsoft Excel. The Regression value of the* R-value$_{FS}$ *equation is 0.9943 (out of a range of 1), the closer the value to 1 indicates a closer match to the original data.*

6.3. Expressing Quality of Experience (QoE) Impact of both Packet and Byte Level Impairment using the E-model

To summarise the impact that both Packet and Byte Level Impairment have on the QoE, the E-model is used. Using the E-model, these two impairments can be subtracted in a scalar manner from the original quality value using an analytical expression.

The E-model equation is expressed in Equation 2.6, taken from the analytical work produced by [18], where the R-value is a measure of quality over a scale of 0 to 100. A value of greater than 70 is equivalent to a public switched telephone network (PSTN) voice call, and a value below 60 is poor.

Range (R-value)	Quality of Experience (QoE)
90 - 100	Best
80 - 89	High
70 - 79	Medium ≡ PSTN
60 - 69	Acceptable
< 60	Poor

Table 6.3, QoE Table

Byte Level Impairment (Codec Impairment), (100 – I_S)

The adaptive MPEG-2 codec gives a maximum R-value of 93.24 at a Frame Size of 168 bytes. The R-value degrades in the manner expressed in Equation 6.7, when the Frame Size is reduced.

$$(100 - I_S) = R_{FS} = -0.0025 \cdot FS^2 + 0.9007 \cdot FS + 11.888 \qquad \textit{from}\text{ Equation 6.7}$$

Packet Level Impairment, I_L, I_D

This is expressed using Equation 2.7 and Equation 2.8 for packet loss, I_L, and packet delay impairment respectively, I_D.

New Quality of Expression (QoE) expression

Bringing all this together, the MPEG-2 Quality of Experience function can be expressed as:

$$\text{R-value (MPEG-2)} = (100 - I_S) - I_L - I_D + A$$

$$\text{R-value (MPEG-2)} = -0.0025 \cdot F_S^2 + 0.9007 \cdot F_S + 11.888 - 30 \cdot ln(1+15 \cdot L_{T(R)}) - (0.024 \cdot D_T + 0.11(D_T - 177.3) \cdot H(D_T - 177.3)) + A$$

Equation 6.8

6.4. Quantifying the Quality of Experience (QoE) when using the TFRC and TFGS CCMs

In this section the E-model is used to address the impact TFRC and TFGS method of congestion response has on the perceived QoE. The TFRC CCM responds to congestion by varying the Packet Rate whilst the encoder maintains a fixed Frame Rate. In contrast, the TFGS CCM responds to congestion by varying the Packet Size to which the encoder can adapt, by truncating the Frame Size. The MPEG-2 codec will be used for both the CCMs, to illustrate the impact of mismatch of the two rates (Frame Rate and Packet Rate) and the impact of Frame Size truncation on the perceived quality. The MPEG codec, which takes into account the Frame Size reduction, will illustrate the benefits of keeping the two rates equivalent (Frame Rate and Packet Rate). The TFGS

CCM achieves this by responding to congestion in the form of Packet Size reduction, which means a reduced Frame Size with respect to the encoder.

The analysis conducted in this section is focused on the impairment arising from the method of congestion response used by the CCM, i.e. how a reduced available network bitrate can have an impact on the sender side of the multimedia connection. This can result in buffering of frames which introduces 'Packet Level Impairment' or by Frame Size truncation which introduces 'Byte Level Impairment'.

This analysis can be referred to as sender side impairment of the multimedia connection excluding the network and receiver impairment. However, these two later impairments will be focussed in chapter 7, where a simulation study is conducted, illustrating how both the CCM perform in various network congestion environments. The simulator study will highlight the impairment caused by the network loss and delay, and receiver loss due to packets arriving 'too-late'. The overall performance analysis in the simulation study will combine the end-to-end impairment including the sender side, network, and receiver impairment.

This evaluation below is based on Sender Side impairment arising from SB delay, SB loss, and PS truncation.

The text below illustrates how the relevant delays and losses are calculated.

6.4.1. Delay (D_T)

$\mathbf{D_T}$ *(ms)* $\mathbf{= (E_D + P_{IPGS(s)} + S_D) + (G_D + Q_D + t_D) + (R_D + Z_{D(R)} + E_{D(d)})}$ *from* Equation 6.1

Encoding delay is the time required to digitize a raw analogue multimedia signal, by producing a stream of frames at a fixed interval. Decoding delay is the time required to convert the digital signal back to an analogue so it can be heard/seen by the receiving end. This delay is subject to processor constraint, i.e. is dependent on the hardware specification, for example a mobile/PDA will have a slower processer than of a desktop computer. Hence, the delay on a mobile/PDA will be larger than that of a desktop. However, this delay will remain fixed during the connection between the end-users. Considering that a live implementation is not used in this study here, therefore this delay is ignored.

Therefore, $\mathbf{E_D} = 0$, $\mathbf{E_{D(d)}} = 0$.

$\mathbf{D_T}\ (ms) = \mathbf{(P_{IPGS(s)} + S_D) + (G_D + Q_D + t_D) + (R_D + Z_{D(R)})}$ *Equation 6.9*

$\text{IPGS}_S = \frac{1}{PR_C}$

$\mathbf{D_T}\ (ms) = \mathbf{(\frac{1}{P_R} + S_D) + (G_D + Q_D + t_D) + (R_D + Z_{D(R)})}$ *Equation 6.10*

Network and receiver delay components are ignored in this study as the focus here is of sender side impairment. Therefore, Equation 6.9 is reduced to:

$\mathbf{D_T}\ (ms) = \mathbf{(\frac{1}{P_R} + S_D)}$ *Equation 6.11*

S_D for each Frame = $\frac{1}{P_R}$ x Discard Threshold size in Frames

$$S_D = \frac{1}{P_R} \times \mathrm{B}$$

from Equation 4.6

In this study the Packet Rate, PR, is considered the long-term Packet Rate of the connection. Hence, it is assumed that the buffer is full once the Packet Rate, falls below the Frame Rate.

Discard Threshold size = 4 = Sender Buffer Size (B), as defined in section 5.2.2.2.

$\mathbf{D_T}$ **(TFRC)** *(ms)* **=** $\left(\frac{\mathbf{1}}{\boldsymbol{P_R}} + \frac{\mathbf{1}}{\boldsymbol{P_R}} \cdot B\right)$

Buffer Size = 4, therefore,

$$= \left(\frac{\mathbf{1}}{\boldsymbol{P_R}} + \frac{\mathbf{1}}{\boldsymbol{P_R}} \cdot 4\right)$$

$$\mathbf{D_T}\ \textbf{(TFRC)}\ \textit{(ms)} = 5 \cdot \left(\frac{\mathbf{1}}{\boldsymbol{P_R}}\right) \qquad \text{Equation 6.12}$$

TFGS does not employ a Sender Buffer, therefore $\mathbf{sb_d}$ = 0

$$\mathbf{D_T}\ \textbf{(TFGS)}\ \textit{(ms)} = \left(\frac{\mathbf{1}}{\boldsymbol{P_R}}\right) \qquad \textit{Equation 6.13}$$

Packet Rate, P_R, is equal to Frame Rate, F_R, 50 fps.

$\mathbf{D_T}$ **(TFGS)** *(ms)* **=** $\left(\frac{\mathbf{1}}{\mathbf{50}}\right)$

$\mathbf{D_T}$ **(TFGS)** *(ms)* **= 20** *Equation 6.14*

6.4.2. Loss (L_T)

$$\mathbf{L_{T(R)}} = \frac{\mathbf{S_L} + \mathbf{N_L} + \mathbf{R_L}}{\mathbf{P_{NPG}}} \qquad \textit{from Equation 2.2}$$

$$\mathbf{L_{T(R)}} = \frac{\mathbf{S_L}}{\mathbf{P_{NPG}}} + \frac{\mathbf{N_L}}{\mathbf{P_{NPG}}} + \frac{\mathbf{R_L}}{\mathbf{P_{NPG}}}$$

$\frac{\mathbf{S_L}}{\mathbf{P_{NPG}}}$, Normalized Sender Buffer Loss,

$$\frac{S_L}{P_{NPG}} = \frac{F_R - P_R}{F_R}$$ *Equation 6.15*

Note: PR_C is bound to the maximum limit of the encoder therefore; the PR_C of the CCM cannot be greater than the Frame Rate of the encoder.

As this analysis excludes network and receiver characteristics, therefore the components of Network Loss, $\mathbf{N_L}$, and Receiver Buffer, $\mathbf{PB_L}$, are ignored.

i.e.

$\mathbf{N_L} = 0$ *and* $\mathbf{PB_L} = 0$

Therefore, the Total Loss ratio, $\mathbf{L_{T(R)}}$, is equal to:

$$\mathbf{L_{T(R)}} = \frac{F_R - P_R}{F_R}$$ *Equation 6.16*

TFGS CCM maintains a P_R equivalent to the Frame Rate, F_R, of the encoder therefore, TFGS does not buffer packets. Hence, TFGS experiences no Sender Buffer: loss and delay at the Sender Side.

$$\mathbf{L_{T(R)}}(\mathrm{TFGS}) = \mathbf{0}$$ *Equation 6.17*

6.4.3. Frame Size Truncation

The impact of the Frame Size, FS, truncation on quality impairment is expressed using Equation 6.7.

6.4.4. Quality of Experience (QoE) Impact

To illustrate the impact of Sender Side quality impairment from an end-to-end perspective constituting of delay, loss and FS truncation, Equation 6.8 is used.

6.4.4.1. Voice flows over TFRC CCM

For the TFRC CCM the R-value degrades when the Packet Rate, PR, reduces (see Figure 6.5). The figure shows the two components which constitute the total quality impairment of the Sender Buffer: delay and loss. The figure shows that a Packet Rate lower than 44 pps will result in a R-value below 60, which be of unacceptable quality level for the 'end-user'.

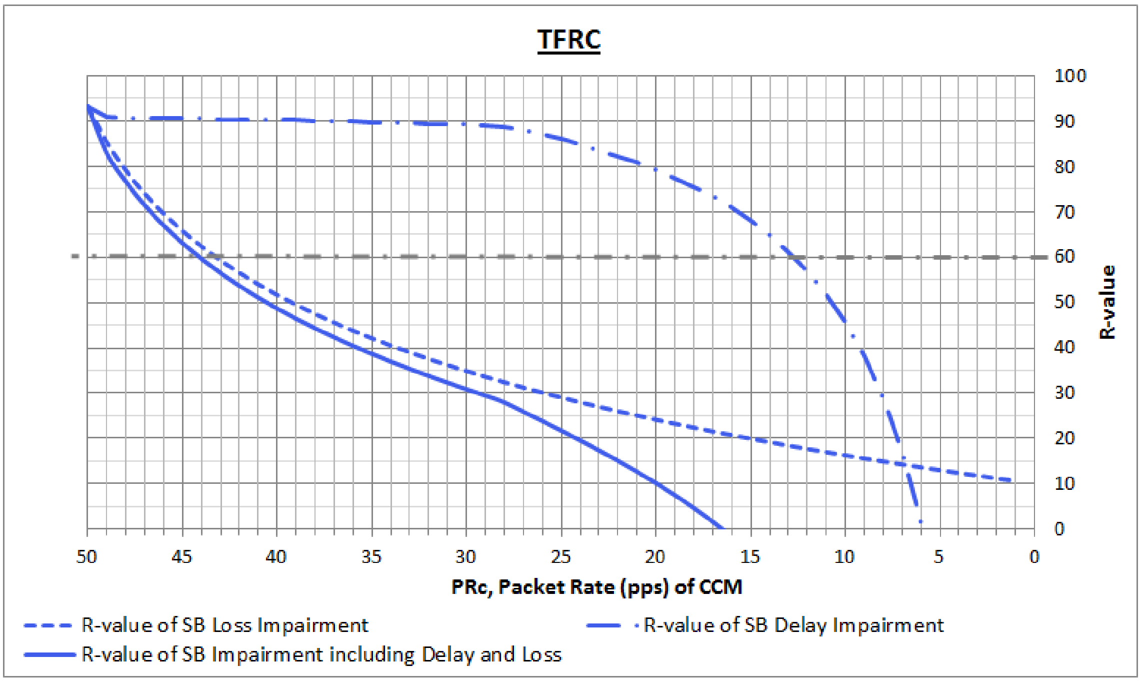


Figure 6.5, Impact on quality from the Packet Rate variation induced by the CCM, resulting in SB Loss and SB Delay

This is because when the Packet Rate reduces to a rate lower than that of the encoder Frame Rate, this introduces loss and delay at the Sender Buffer (SB). A long term reduced Packet Rate will result in a fully occupied SB, where all packets are experiencing a delay of: $5 \cdot \left(\frac{1}{P_R}\right)$ (as described in Equation 6.12) and a loss of: $\frac{F_R - P_R}{F_R}$ (as shown in Equation 6.15). Figure 6.5 illustrates the impact of SB loss on the quality of a voice connection. A Packet Rate lower than 43 pps will result in an overall loss of more than 14 percent, reflecting a Quality level below the minimum required (R-value=60). The loss impairment formulation modeled by Equation 2.7 is based on the overall loss over the duration of the connection. However, when the SB is full, the voice connection will experience consecutive losses and this impairment can be far greater than what is illustrated.

The reduction in Packet Rate will also result in an additional delay at the SB (see Figure 6.5), although the initial impairment caused by delay has far less impact on the quality of a voice connection that packet loss would produce, and this is because the delay factor has not approached above the 177.3 ms mark which starts to effects interactivity (see Equation 2.8). Once the Packet Rate approaches below 28 pps, this results in a delay of 178 ms $\left(5 \cdot \left(\frac{1}{28}\right)\right)$, and this is when the delay factor becomes more prominent. However, the accumulative effect from network and receiver delay will induce further quality degradation of the voice connection. These impairments (network and receiver) will be more apparent in the simulation study chapter 7, where these impairments will be considered.

6.4.4.2. Voice flows over TFGS CCM

For the TFGS CCM the R-value degrades when the Frame Size is reduced (see Figure 6.6). The figure shows that when the Frame Size, FS, is truncated to a size lower than

67 bytes, this will result in an R-value below 60, which is of unacceptable quality for the 'end-user'.

TFGS CCM maintains a Packet Rate equivalent to the Frame Rate of the encoder therefore, TFGS does not buffer packets. Hence, TFGS experiences no Sender Buffer: loss and delay at the Sender Side.

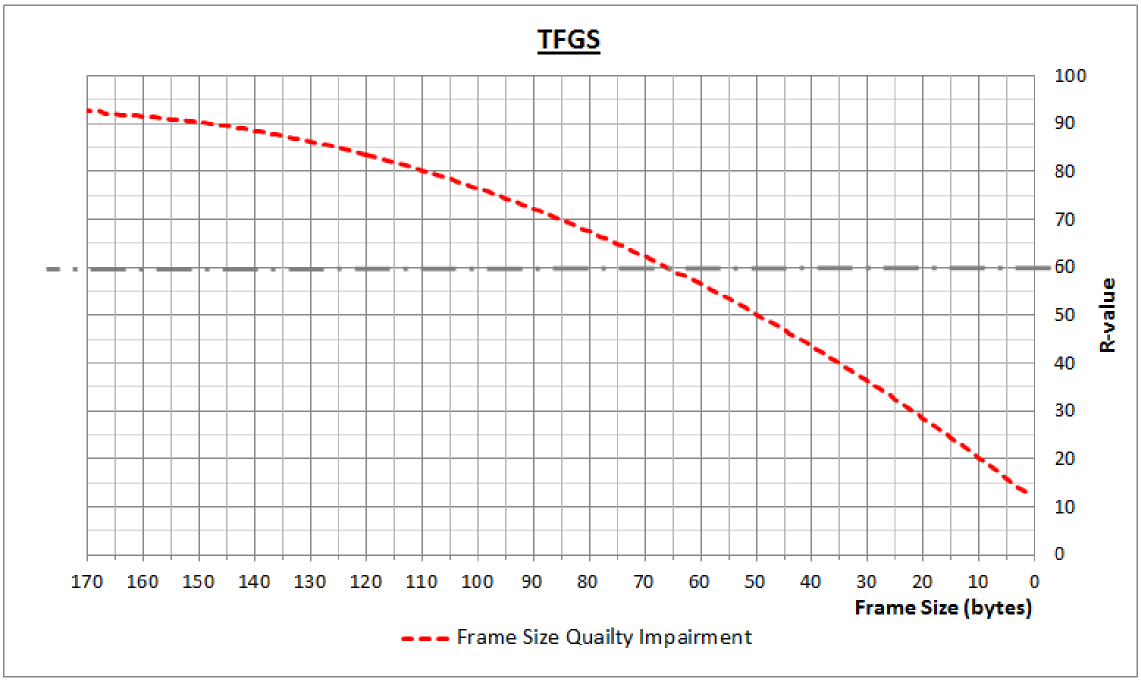


Figure 6.6, Impact on quality from the Frame Size truncation induced by the CCM

6.4.4.3. <u>*Voice flows over TFRC vs TFGS CCM*</u>

In order to compare the performance of the two CCM's (TFRC and TFGS), their method of congestion response is mapped on to a bitrate axis, using Equation 3.5 (from chapter 3).

$$\mathbf{T_{ETR} = P_R \times (8 \times P_S)}$$ *from* Equation 3.5

Where:

P_S = (FS + 40 Bytes),

40 bytes for headers (IP: 20 bytes, UDP: 12 bytes, RTP: 8 bytes)

P_R = Packet Rate

in the case of TFGS: $P_R = F_R$,

whereas for TFRC: $P_R = P_R$ of the CCM.

This enables both the CCM's to be compared on the same axis irrespective of the type of congestion response they perform. Figure 6.7 shows that voice connections running over a TFGS CCM are able to achieve a minimum acceptable voice quality of (R=60), with a bitrate of 42 kbps compared to 73 kbps when using the TFRC CCM. This means that the TFRC CCM requires a higher network resource in order to sustain the same minimum quality. The TFGS CCM can sustain its minimum quality at lower bandwidths than the TFRC CCM. This is beneficial to voice applications as well as the network. This is because a higher number of voice connections can operate simultaneously in a limited network resource, whilst maintaining their minimum quality.

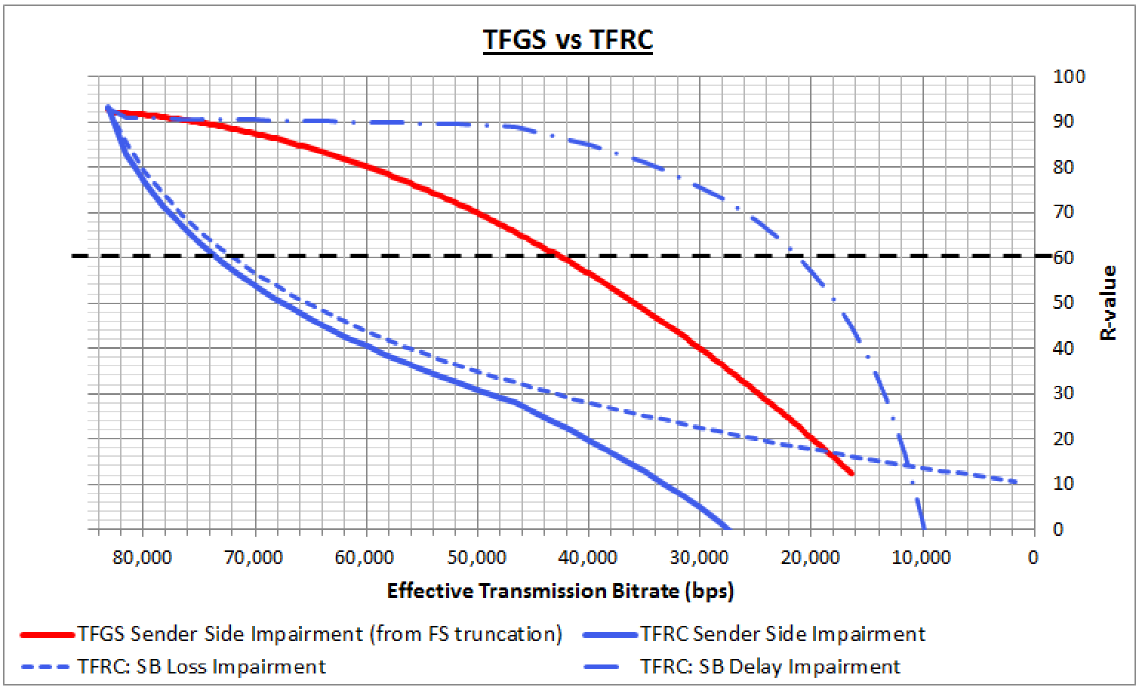


Figure 6.7, Impact on quality when reducing Transmission Bitrates

As seen from Figure 6.7, the TFRC CCM degrades at a faster rate than that of TFGS. This is because the reduction in the Packet Rate in response to congestion has a larger impact on packet loss and delay experienced at the SB compared to truncation from Packet Size.

Referring to the individual components of SB packet loss and delay as seen in Figure 6.7, their combined effect is more severe than that of Frame Size truncation. These two particular impairments (SB: loss and delay) are not present when responding to congestion in the form of Frame Size truncation (as performed by the TFGS CCM). The total loss and delay will be significantly lower than that of the TFRC CCM, when the network and receiver characteristics are taken into account. This will be more thoroughly discussed in the simulation study chapter, 7.

To summarise, a voice connection operating over the TFGS CCM will experience 'Byte-Level Impairment' due to Frame Size (FS) truncation, and 'Packet Level Impairment' at network and receiver. However, the TFRC CCM will only experience packet level impairments, but at 3 different places in the transmission process: sender, network and receiver. Although it will not experience any byte-level impairment (because it sends fixed sized frames) the impact seen from SB packet loss and delay has a greater impact on quality degradation compared to that of FS truncation.

6.5. Summary

This chapter looked at how to provide a measurement scheme for assessing the level of quality loss caused by Packet Level and Byte Level Impairment (also referred to as Codec Impairment). The Packet Level Impairments (such as delay and loss of packets) were quantified using the expressions found in [18]. In order to determine the Codec Impairment as a function of Frame Size, the available quality results had to go through a number of transformation processes before they conformed to the E-model expression; from encoder bitrate to Frame Size and then its respective MOS quality to R-value. This novel formulation enabled the combination of both the Packet and Byte Level Impairment into a scalar form, which can be subtracted from the original quality value using an analytical expression. This expression is referred to as the E-model equation for the MPEG-2 codec.

The results used to quantify the Codec Impairment were based on the MPEG-2 encoder, which provides coarse-grain adaptability with respect to encoder bitrate reduction. Although this thesis proposes to use Fine Grain Scalable (FGS) encoding, which can offer a higher degree of adaptability and quality at equivalent encoder bitrates to that of MPEG-2 [7], a quality measure for modelling the level of FGS Codec Impairment is not available because the codec has currently not undergone subjective listening tests.

Although the MPEG-2 codec is used, the FGS R-values are expected to be better. This is shown when the Signal-to-Noise-Ratio (SNR) comparisons are made between two codecs by [7], and summarized in Table 6.4. The higher SNR values of the FGS codec at the same Encoder Bitrates that of MPEG-2, indicate that the encoding quality technique is superior. However, the FGS codec has not undergone subjective listening tests; therefore a standard quality measure is not available for use in the E-model expression.

	SNR (dB) at the Following Encoder Bitrates (kbps)		
Codec	*32*	*64*	*128*
FGS	17.71	23.03	30.88
MPEG	13.99	17.99	26.06

Table 6.4, FGS vs MPEG, in terms of SNR value

The analysis conducted in this chapter illustrated the impact a reduced available network bitrate will have on the sender side impairment either in the form of Packet Level Impairment or Byte Level Impairment. The results conclude that a Congestion Control Mechanism (CCM) which responds to network congestion by reducing its Packet Rate (in the case of TFRC CCM) has a greater impact on quality degradation than that of Packet Size truncation used by the TFGS CCM. This is because the TFRC CCM (which reduces Packet Rate in results of congestion) introduces Sender Buffer (SB) delay and loss simultaneously, whereas as for the TFGS CCM the Packet Size truncation introduces only Codec Impairment in the form of sender-side impairment. Hence, the total impact of SB loss and SB delay is greater.

7. Simulation Results, Performance Analysis

This chapter illustrates the performance of the two CCMs by considering: (*1*) Different Traffic mixes in the network (Homogenous traffic mix and Heterogeneous traffic mix). (*2*) Different congestion environments (by varying the number of flows whilst keeping the bandwidth fixed). (*3*) The Quality of Experience (QoE) impact, examining the response by the Congestion Control Mechanism (CCM) at the a) Sender, b) Network, and c) Receiver side.

7.1. Homogenous Traffic Mix

7.1.1. Network Scenario Description

The network scenario description is defined as in Table 5.3, further details can be found in section 5.2.3.

7.1.2. Congestion Response:

7.1.2.1. Sending Rate (SR) and Throughput (TP):

Over a bottleneck of 499,200 bps, with a number of voice flows competing for the same bandwidth, the flows are configured with either one of the two CCMs. These respond to congestion by reducing their Sending Rate (SR). This is shown in Figure 7.1. The Sending Rate decreases in an approximately ‘fair-share’ manner as defined by

Equation 5.16, and it is seen that voice flows configured with either one of the two CCMs have approximately the same Sending Rate in 'bits per second' (bps). This proves that the overall congestion response whether Packet Rate or Packet Size is equivalent in terms of 'bits per second'. Figure 7.2 shows the throughput of flows at the receiving end. The achieved throughput is equivalent to the 'fair-share' rate. The closer the Sending Rate is to the 'fair-share' rate, the lower the amount of loss. This can be seen from Figure 7.3.

The CCM's Sending Rate closely follows the 'fair-share' rate indicating that the CCM's input load into the network takes into account the end-to-end constraint of the network capacity. As a result, it only injects that volume of load which the network can tolerate, avoiding as much network loss as possible (see Figure 7.3). This is the key strength of using the 'TCP-friendly' rate equation, which indicates the available network bitrate based on loss, delay, and other factors, see Equation 2.9 (in chapter 2).

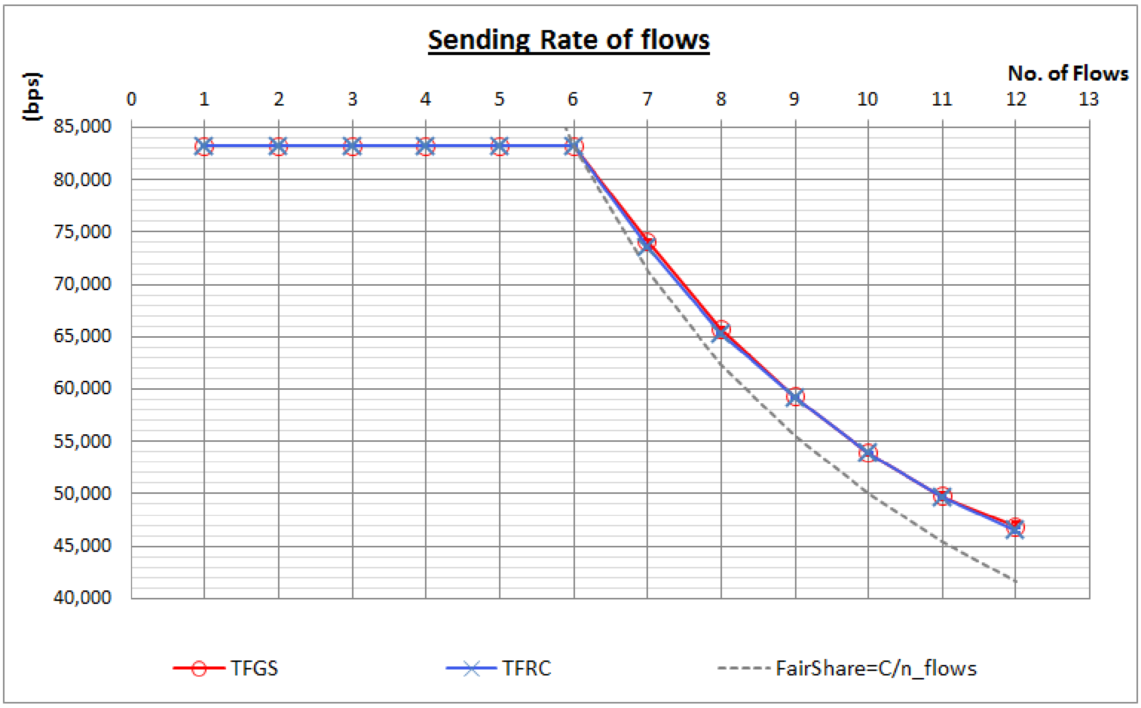


Figure 7.1, Sending Rate

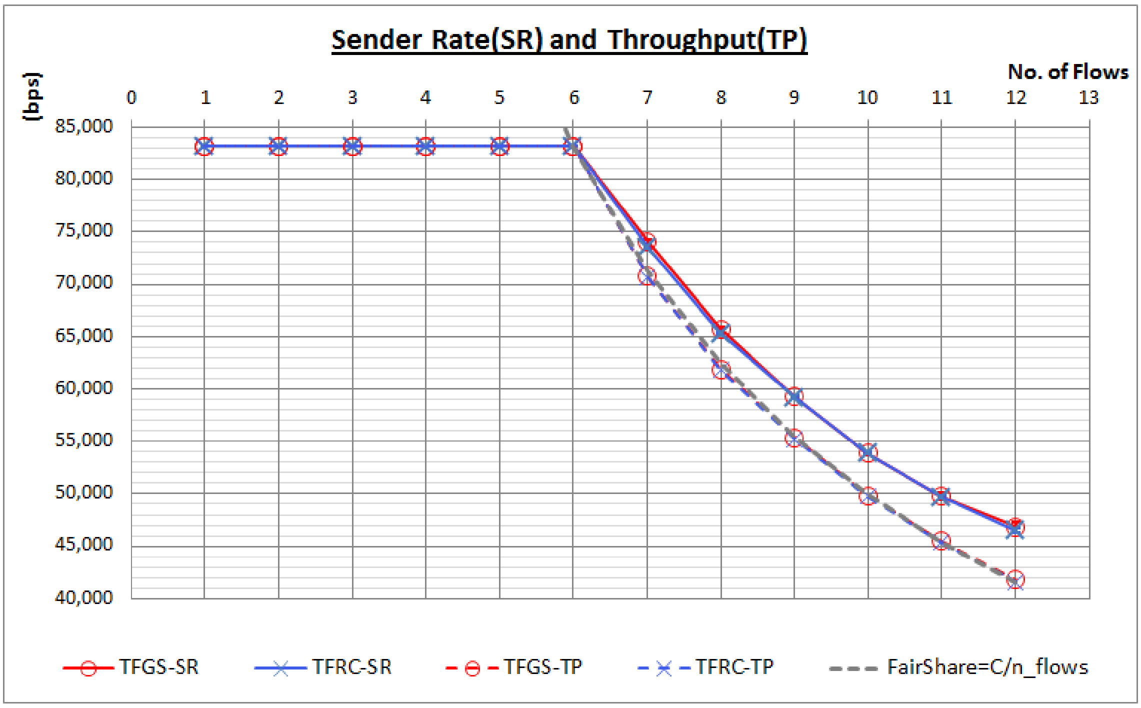


Figure 7.2, Sending Rate (SR) and Throughput (TP)

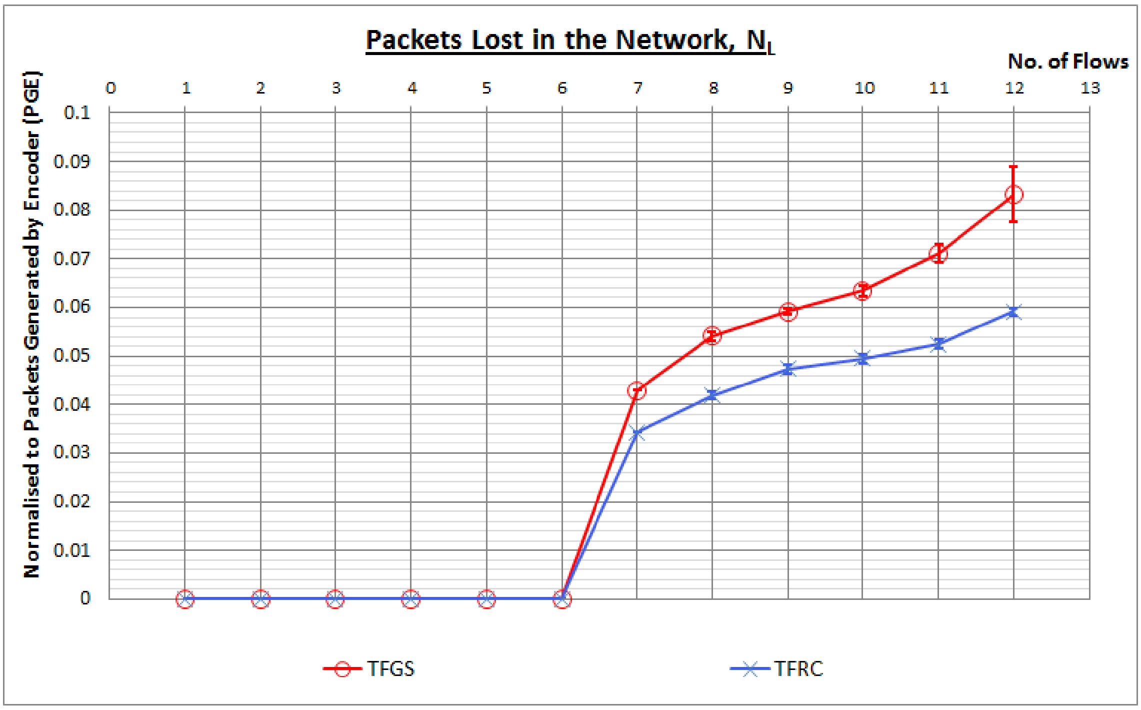


Figure 7.3, Network Loss, (Bars showing Standard Deviation)

7.1.2.2. Method of Congestion Response:

The response to congestion for the TFGS CCM is to truncate the Packet Size, as illustrated in Figure 7.4. For the TFRC CCM the response is to increase the "Inter-Packet-Gap-Spacing" (IPGS) resulting in a reduced number of packets being injected into the network, i.e. a reduced Packet Rate, PR_C ($P_R = {}^{1}/_{P_{IPGS}}$). This is illustrated in Figure 7.50.

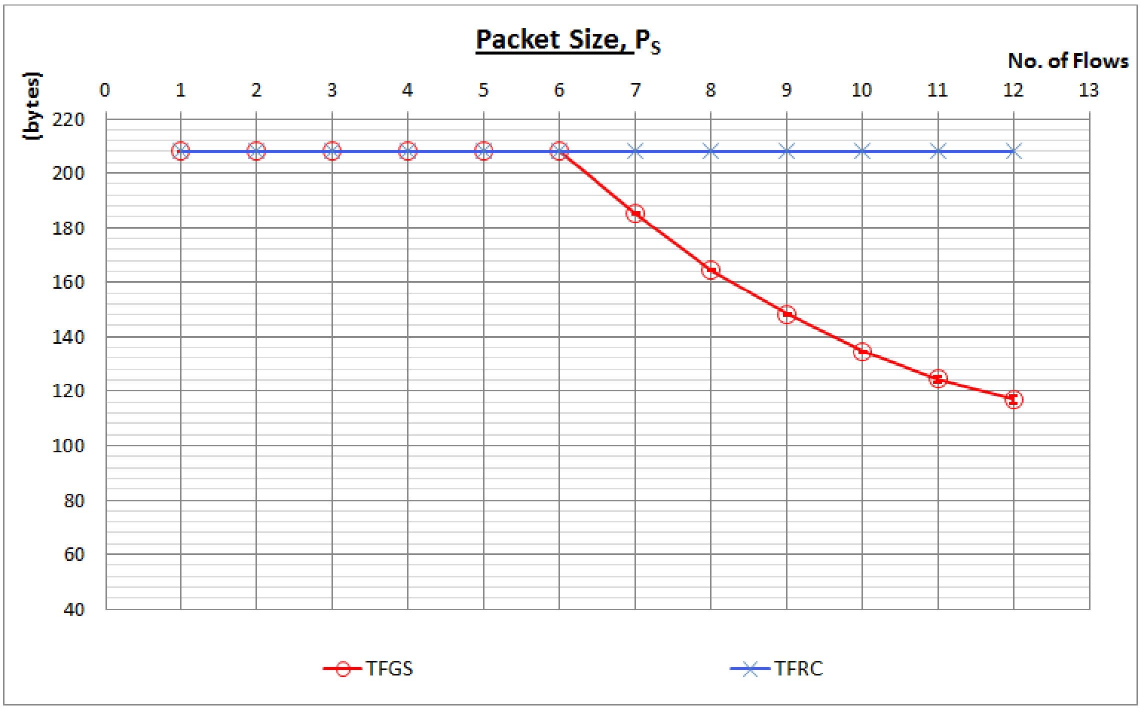


Figure 7.4, Packet Size

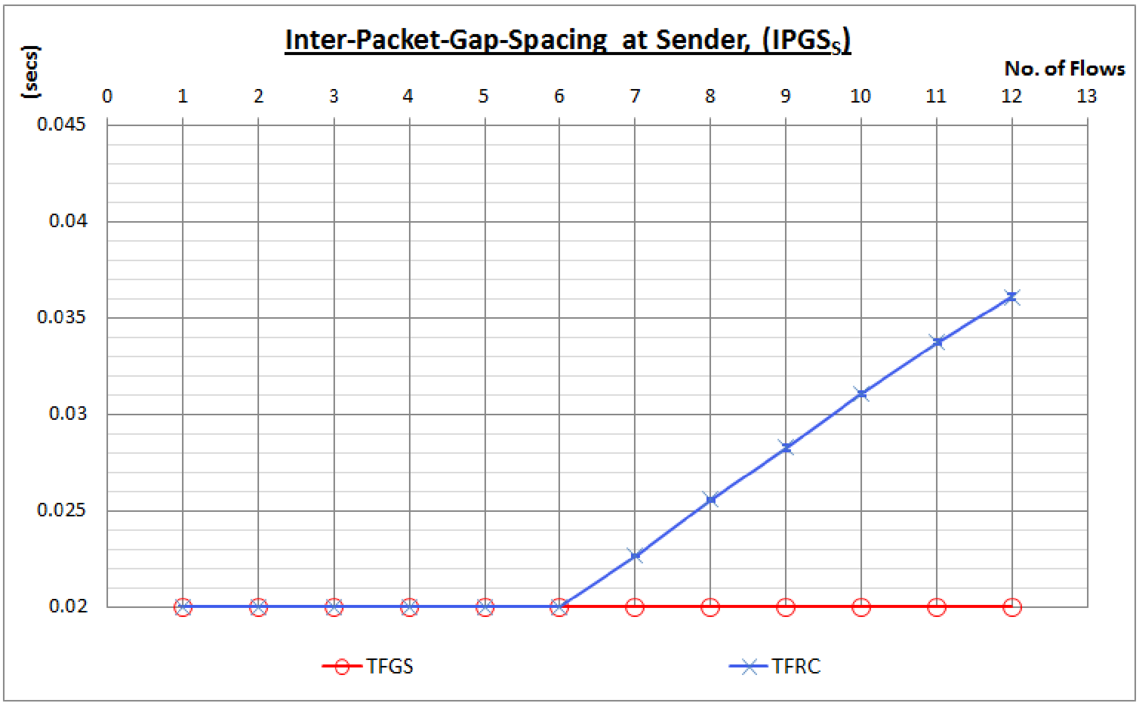


Figure 7.5, "Inter-Packet-Gap-Spacing"

7.1.2.3. A perspective in terms of 'Offered Load'

Figure 7.6 illustrates what the term 'offered load' means. To quantify the number of flows operating over the network in terms of 'offered load', a simple formulation is used, equating the maximum input load generated by the voice application and the fair-share of bandwidth.

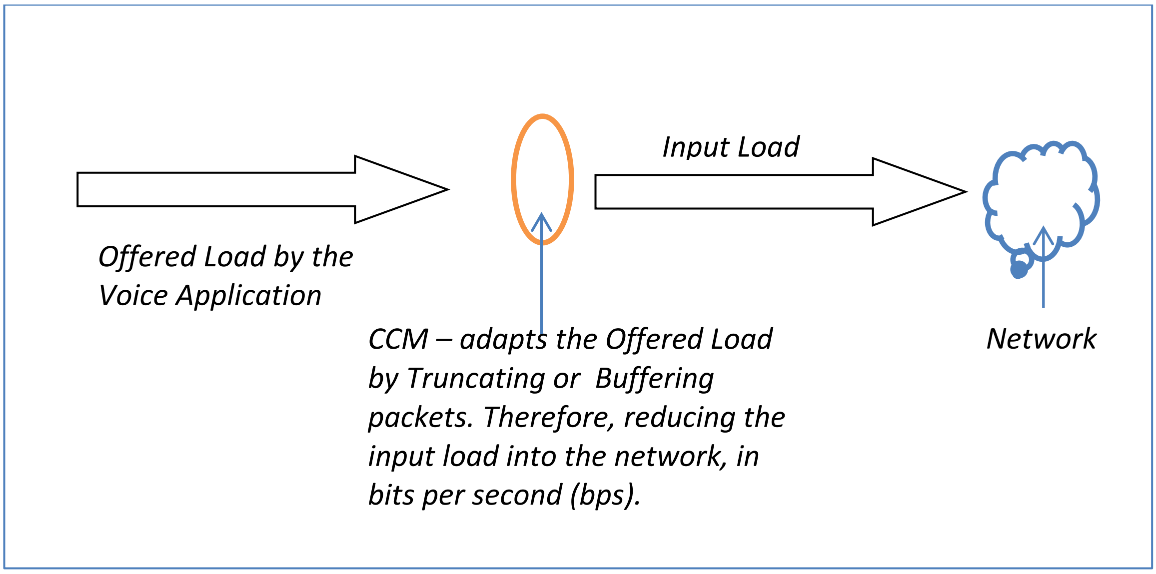


Figure 7.6, Offered Load

Maximum load generated by the Voice Application = Encoder Bitrate = VoIP Offered Load $=$ Packet Size $\left(\mathrm{P_{S(max)}}\right) \times$ Packet Rate $\left(\mathrm{P_{R(max)}}\right)$

$$\text{Fair Share of bandwidth among flows} = \frac{\text{Bottleneck Capacity (C)}}{\text{Number of Flows } (n_{flows})}$$

VoIP Offered Load with respect to Network Capacity

$$= \frac{\text{max Bitrate of a voice flow}}{\text{Fair Share of bandwidth among flows}}$$

$$\text{VoIP Offered Load} = \frac{\mathrm{P_{S(max)}} \times \mathrm{P_{R(max)}}}{\mathrm{C}} \cdot n_{flows}$$

Equation 7.1

A VoIP ‘offered load’ value between 0 and 1 indicates that the flows operating over the network are running at their maximum bitrate, and that the network has sufficient capacity to support the maximum rate. An offered load value higher than 1 indicates that the maximum bitrate cannot be supported, and that the flows need to adapt their bitrate in order to operate over the network at 'fair-share' rate. This adaptation is carried out by the CCM by either reducing the Packet Rate (in the case of TFRC CCM) or Packet Size (in the case of TFGS CCM).

The equivalence of number of flows to offered load is shown in Table 7.1.

No. of flows	VoIP Offered Load	VoIP Offered Load in terms of Percentage (%)
1	0.17	17
2	0.33	33
3	0.50	50
4	0.67	67
5	0.83	83
6	1.00	100
7	1.17	117
8	1.33	133
9	1.50	150
10	1.67	167
11	1.83	183
12	2.00	200

Table 7.1, VoIP Offered Load

7.1.3. Sender Side Impairment:

7.1.3.1. TFRC: Sender Buffer (SB): Loss and Delay, Quality Impairment

Both the CCMs have approximately the same Sending Rate. This means that the overall congestion response is equivalent in terms of 'bits per second'. However, the TFRC CCM reduces its Packet Rate. This causes a difference between the Frame Rate and Packet Rate, leading to buffering of packets at the sender side. Buffering of packets results in delay to the packets, and loss when the buffer becomes full. The impact of loss and delay is illustrated in Figure 7.7 and Figure 7.9 respectively.

Figure 7.8 and Figure 7.10 highlight the impact on the voice quality with respect to the R-value (on a scale of 0 to 100, where 100 represents excellent quality). As the loss increases, the quality degrades and a loss greater than 15 % (0.15 on the normalised left Y axis in Figure 7.8) will result in a voice quality lower than R=60, which is unacceptable to the user. Additionally, similar behaviour is seen with delay. As the delay increases the quality degrades, although the impact of delay is far less than that of loss see Figure 7.11. This is because the loss component in the E-model carries more weight than that of delay[8], and this is illustrated in Figure 7.12.

These components of SB delay and SB loss are not present with the TFGS CCM, because it sends frames as soon as they are generated (i.e. maintaining a Frame Rate equal to the Packet Rate). However, it responds to congestion by reducing the Packet Size. This will be discussed in the next section, 7.1.3.2.

[8] This delay excludes network and receiver delay however, if the total delay exceeds 170 ms then the impairment from delay will become more prominent, see Figure 7.12.

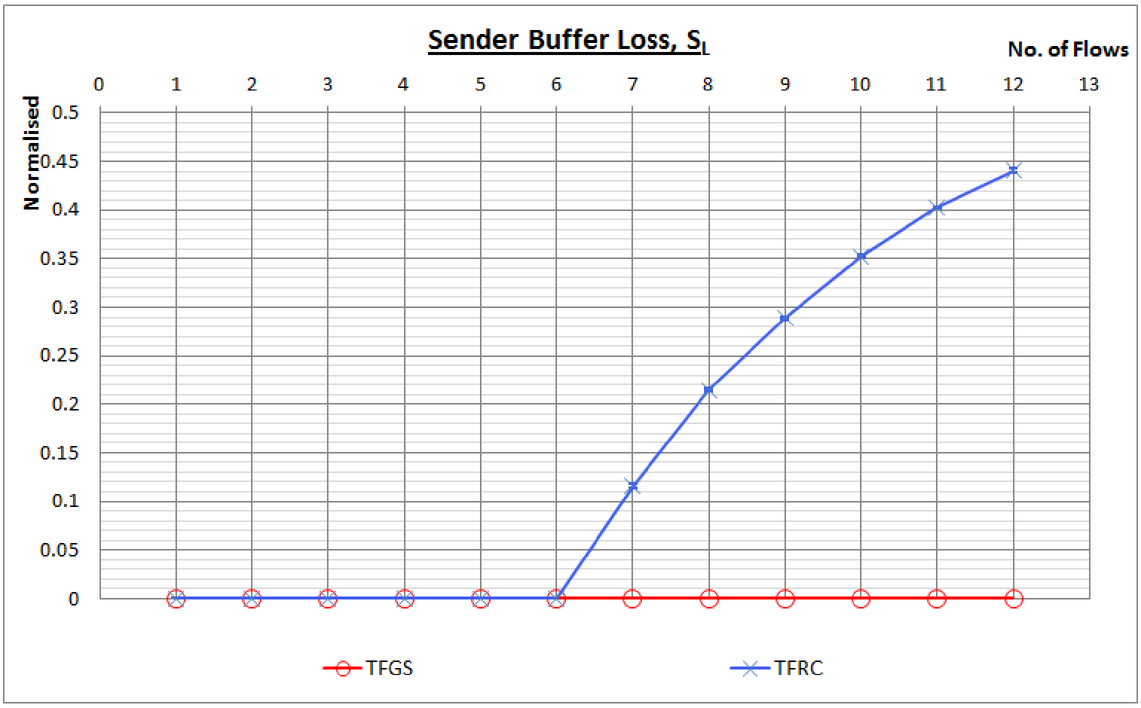


Figure 7.7, Sender Buffer Loss, S_L

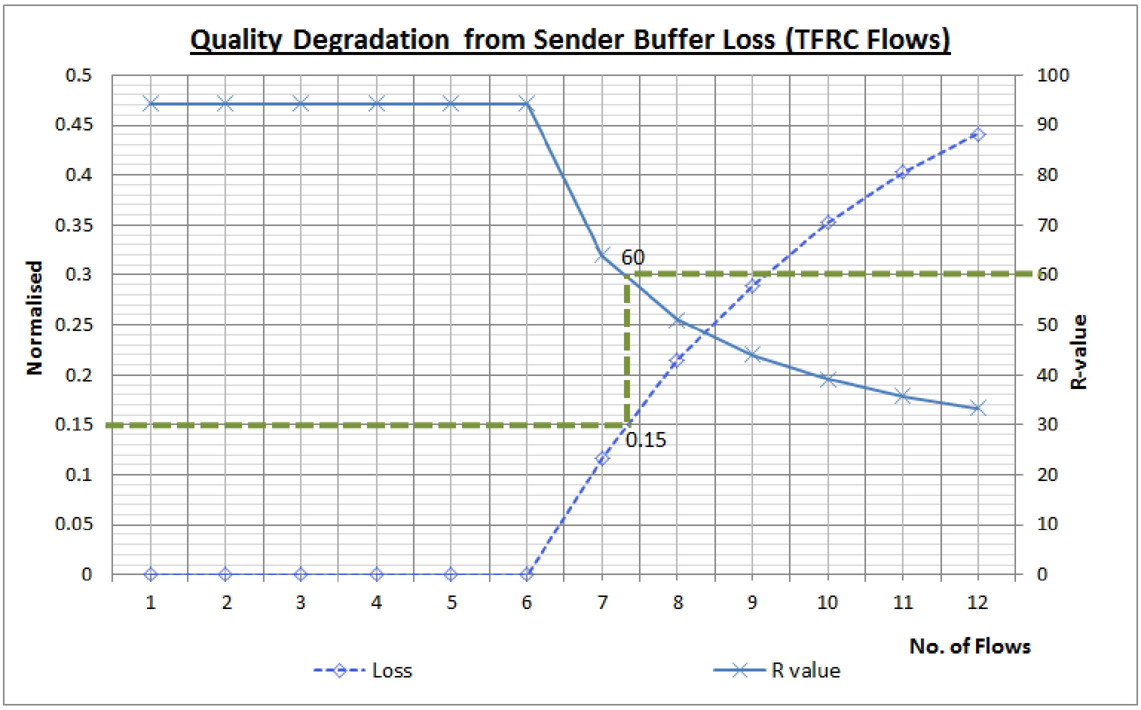


Figure 7.8, Quality Degradation from Sender Buffer Loss (TFRC Flows)

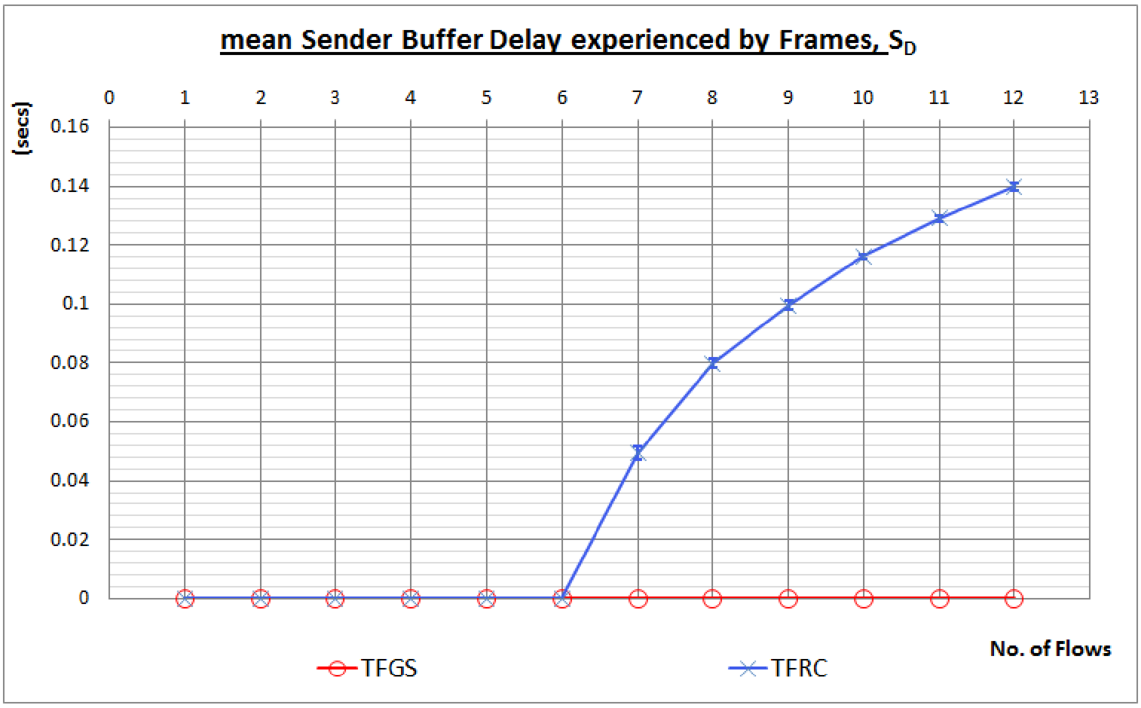


Figure 7.9, Sender Buffer Delay, S_D

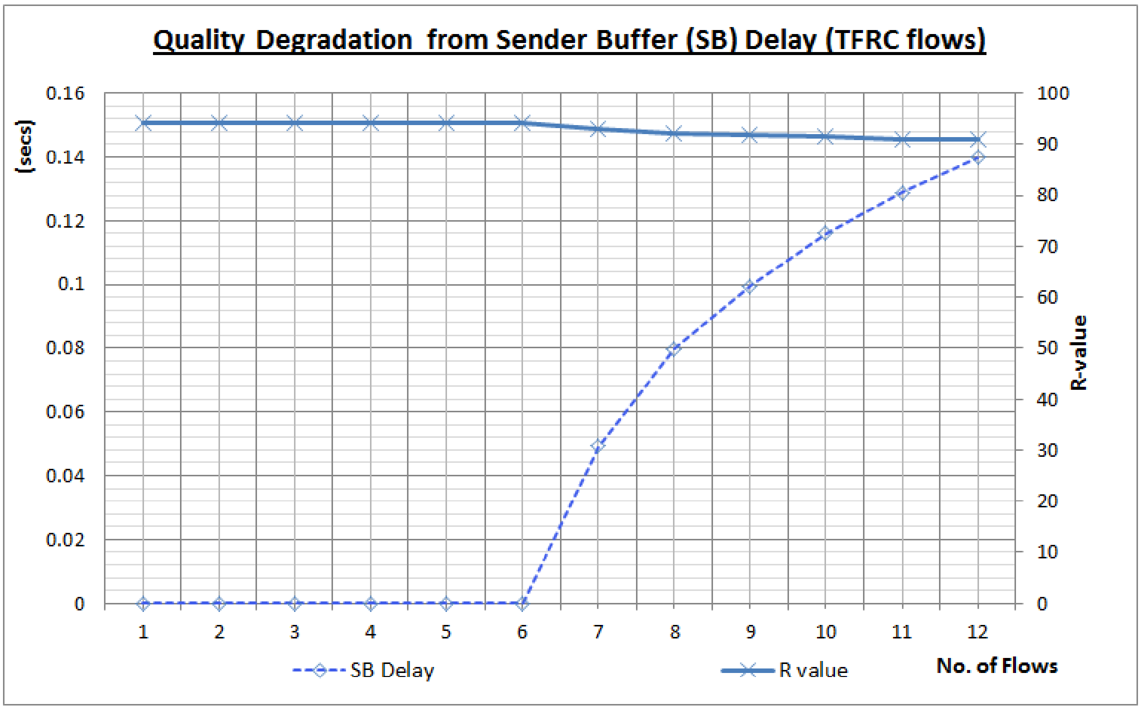


Figure 7.10, Quality Degradation from Sender Buffer (SB) Delay (TFRC flows)

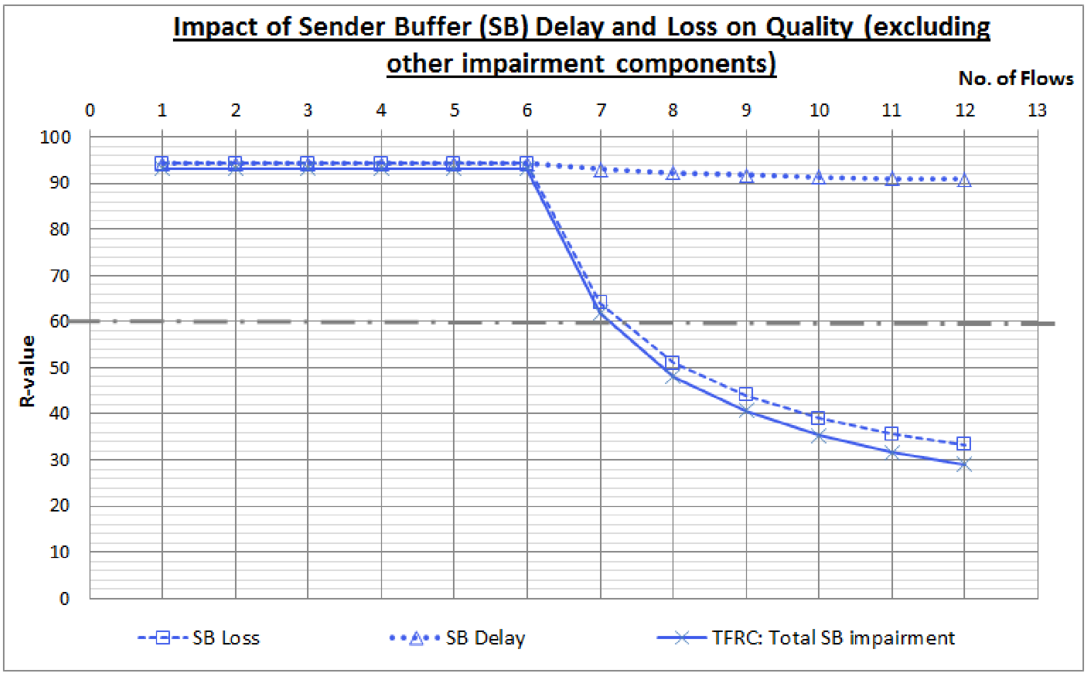


Figure 7.11, Impact of Sender Buffer (SB) Delay and Loss on Quality (excluding other impairment components)

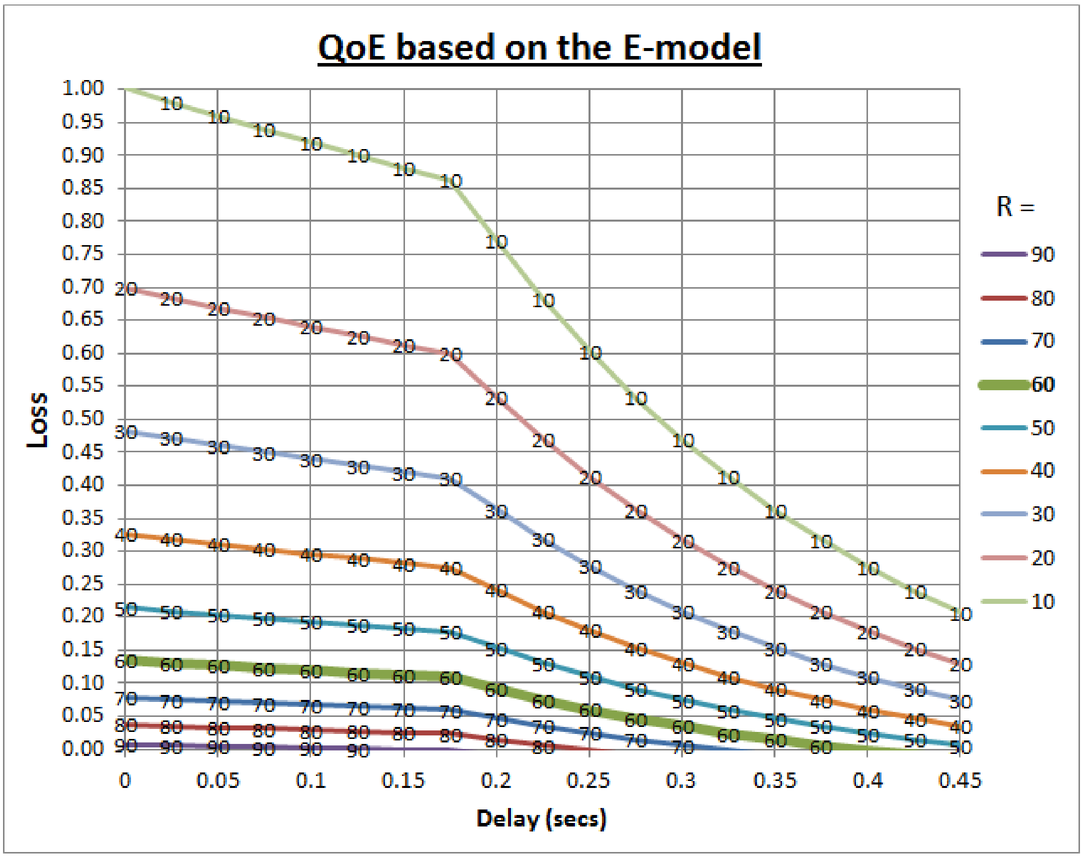


Figure 7.12, Impact of Delay and Loss on the R-value of a voice call

7.1.3.2. <u>TFGS: Packet Size (PS) Truncation, Quality Impairment</u>

Both the CCMs have approximately the same Sending Rate. This means that the overall congestion response is equivalent in terms of 'bits per second'. However, the TFGS CCM responds to congestion by reducing the Packet Size, as seen in Figure 7.4. The reduction in Packet Size results in reduced quality, the impact this has on the voice quality is shown in Figure 7.13.

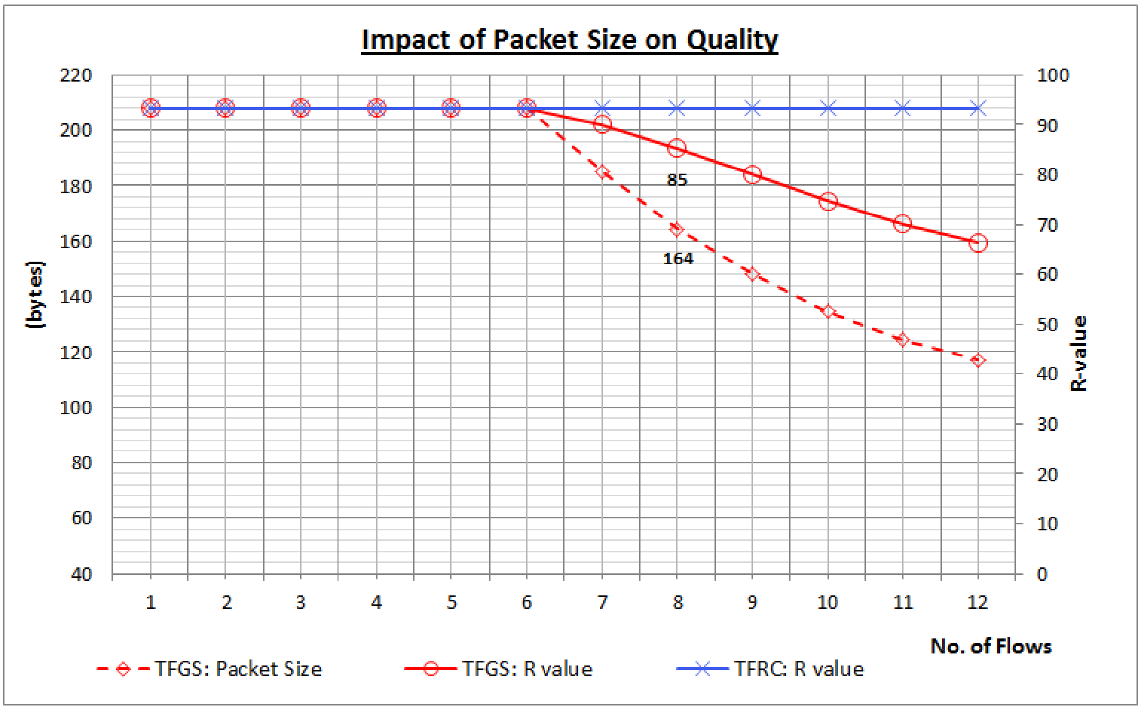


Figure 7.13, Impact of Packet Size on Quality

The degree of impact from Sender Buffer (SB) and Packet Size impairment on the QoE is illustrated in Figure 7.14. For example, when 8 flows are competing over the network using the TFRC CCM, the SB impairment (including SB loss and SB delay) results in a quality degradation to an R-value 48, which is lower than the minimum required acceptable quality, R=60. This level of quality degradation arising from the SB component excludes the impact of loss and delay from the remaining components;

(network and receiver). This illustrates how severe an impact the SB has on voice quality when the CCM responds to congestion by reducing the Packet Rate.

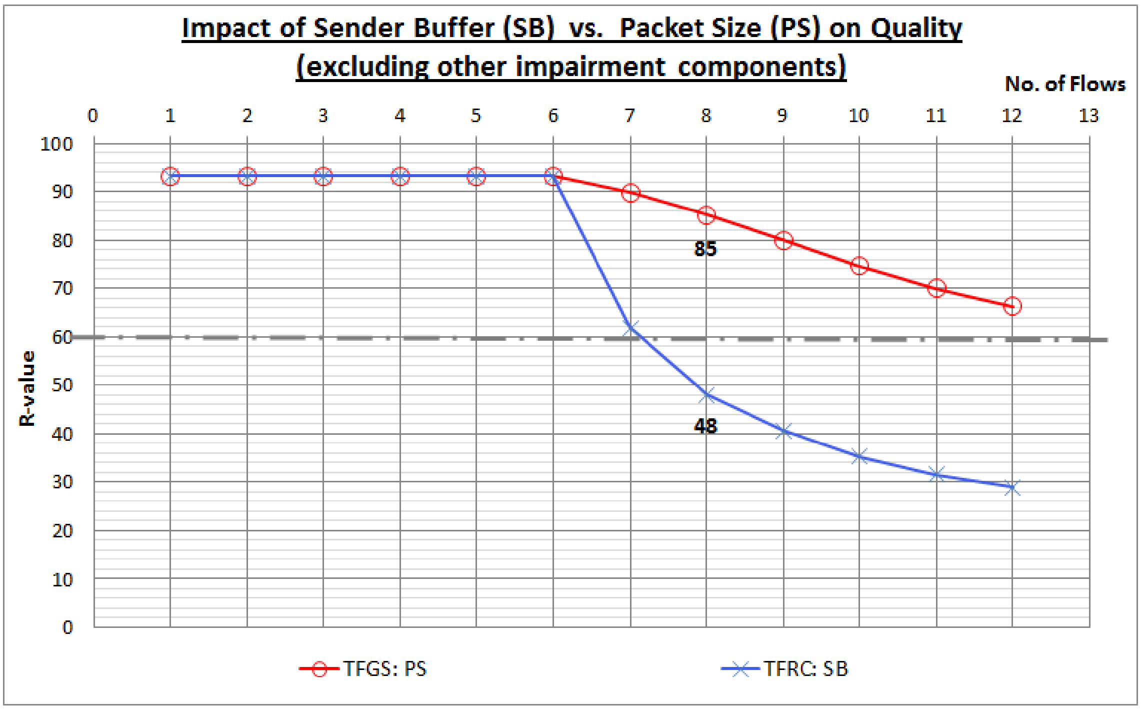


Figure 7.14, SB versus Packet Size quality impairment

In contrast the TFGS CCM, which responds to congestion by truncating the Packet Size, results in a quality degradation to an R-value 85 for the same 8 flow scenario described above. This indicates a far less quality degradation, a difference of ΔR=37. Additionally, Figure 7.14 illustrates comparatively the rate at which the impact the SB and Packet Size truncation affects quality. It is clearly evident that when using the TFRC CCM the SB impairment has a higher rate of quality degradation.

7.1.4. Network Side Impairment

Both the CCMs experience the same mean delay in the network as shown in Figure 7.15. However, the TFGS CCM experiences a higher network loss rate (see Figure 7.3).

This is because a greater number of packets are sent into the network by the TFGS CCM than the TFRC CCM, see Figure 7.16. The reason for this is that the TFGS CCM maintains a Packet Rate equivalent to the Frame Rate of the encoder, whereas TFRC CCM adapts its Packet Rate, buffering packets at the sender and losing them if the buffer becomes full.

The higher network loss rate of the TFGS CCM leads to a slightly higher quality degradation compared to that of TFRC, see Figure 7.17. However it isn't as significant as the Sender Buffer loss impairment of TFRC. For example, the difference between the two CCMs' network impairment is approximately ΔR=3, with 8 flows in the network. This is because the Network loss is less than 14% after this point the quality degradation from loss becomes significant see Figure 4.8.

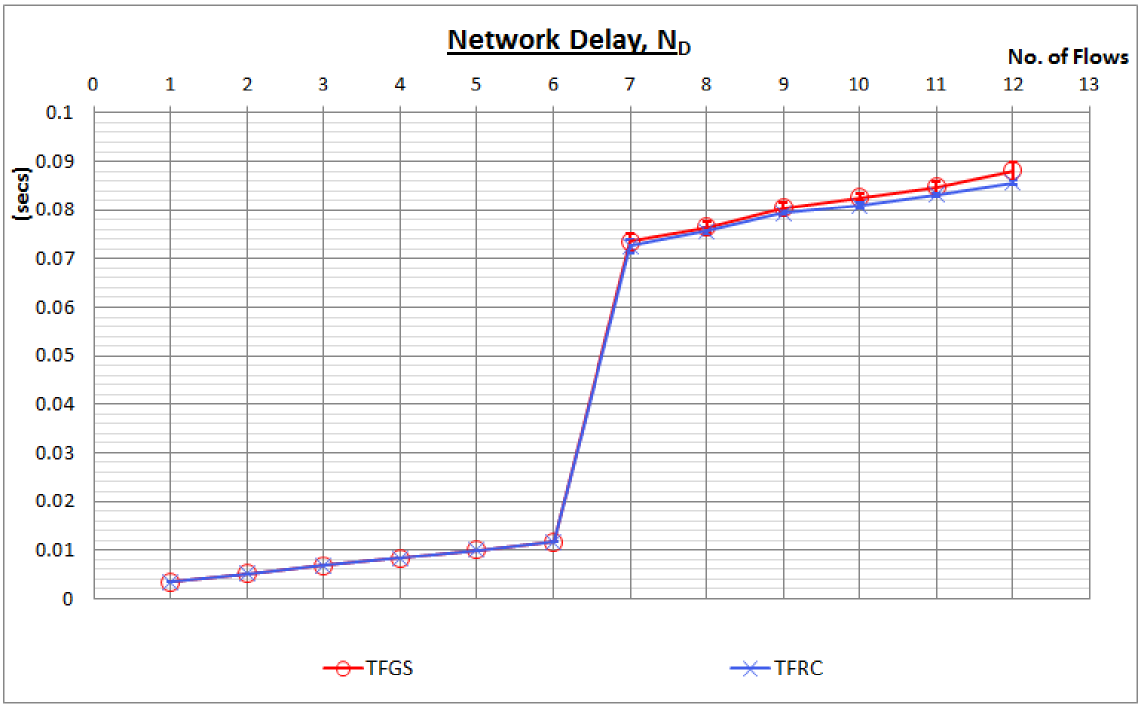


Figure 7.15, Network Delay, N_D

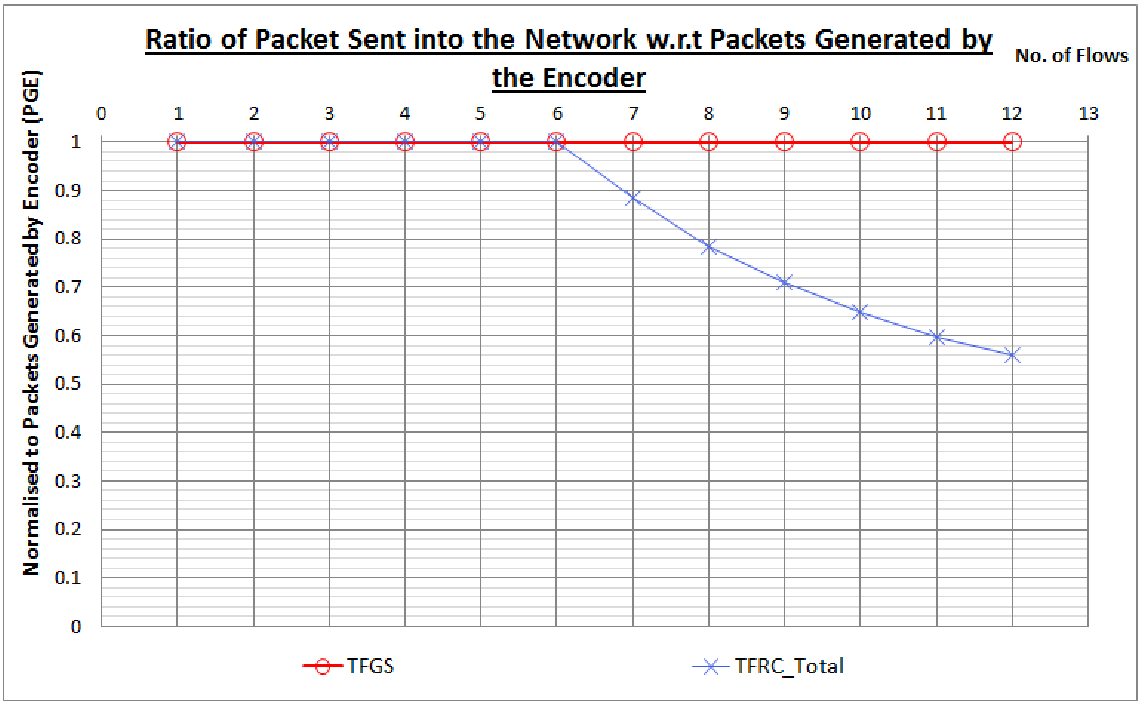


Figure 7.16, Ratio of Packet Sent into Network w.r.t Generated by the Encoder

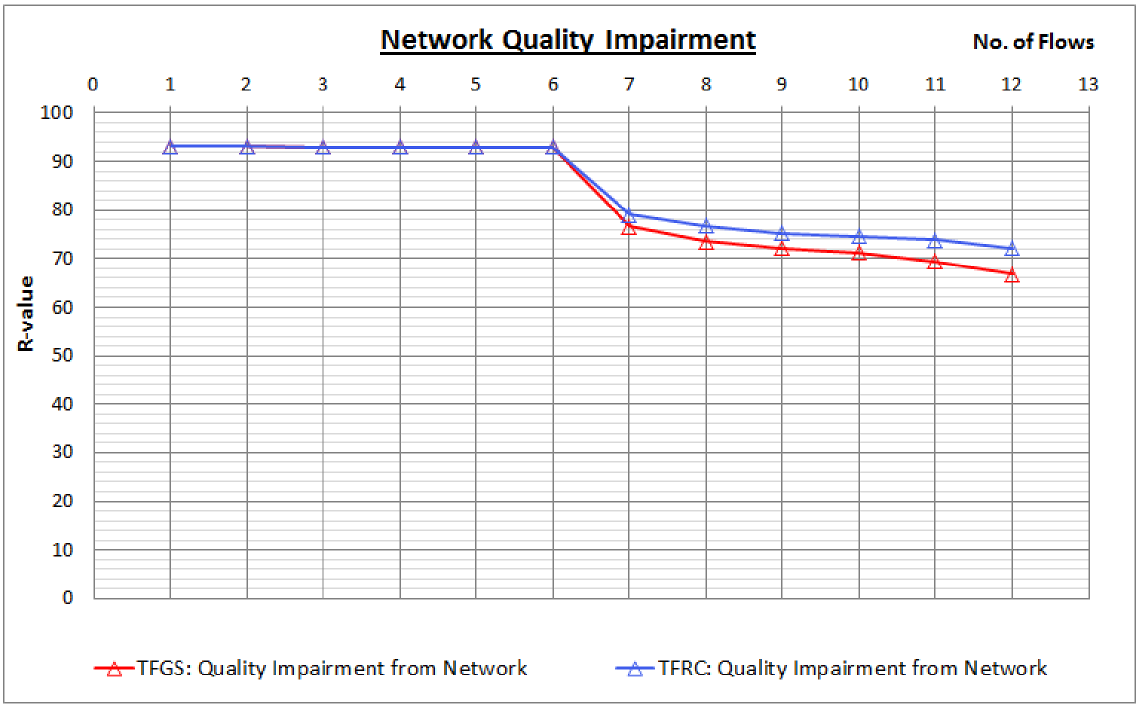


Figure 7.17, Network Quality Impairment

7.1.5. Receiver Side Impairment

Fixed sized de-jitter (Playout/Receiver) buffers are used for all flows. Therefore, the receiver delay is of 80 ms, which is equivalent to buffering 4 frames before playout begins.

The Receiver Buffer loss for both the CCMs is negligible see Figure 19 and 28 in Appendix I, Results: Homogeneous Traffic Mix. This is because when using Random Early Discard (RED) Active Queue Management (AQM) in the routers as shown in the network topology Figure 5.5a. The RED mechanism keeps the queue variation in a tight range, between the minimum and maximum thresholds. This reduces the amount of variation in the network delay. This can be seen from Figure 7.18 and Figure 7.19, which represent the Probability Distribution Function (PDF) of network delay for TFRC and TFGS respectively[9]. The calculation of the Receiver Buffer loss is based on the variation of the packet delay from the mean delay, see Equation 6.4. If the difference is greater than the size of the Playout Buffer (PB), here 80 ms, then it is considered lost. For example Figure 7.18 shows that the peak network delay occurs at 90 ms for the TFGS flows. Adding 80 ms to this, in order to take account the PB size, gives 170 ms (this method of calculation can be visualized using Figure 5.4). The probability that flows experience a network delay of 165 ms or greater is minute. Hence, the impact seen on the RB loss (and impairment) is also very small.

[9] The PDF represents all packets across each flow number, and of all simulation sets

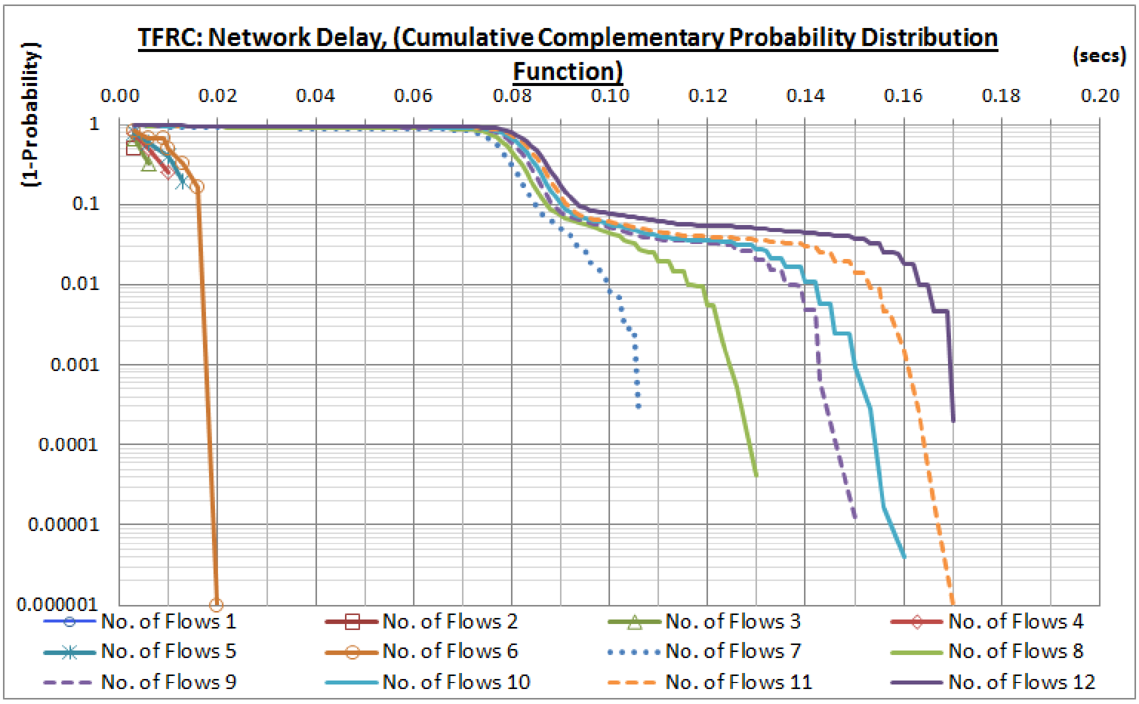


Figure 7.18, TFRC: Network Delay Probability Distribution Function (PDF)

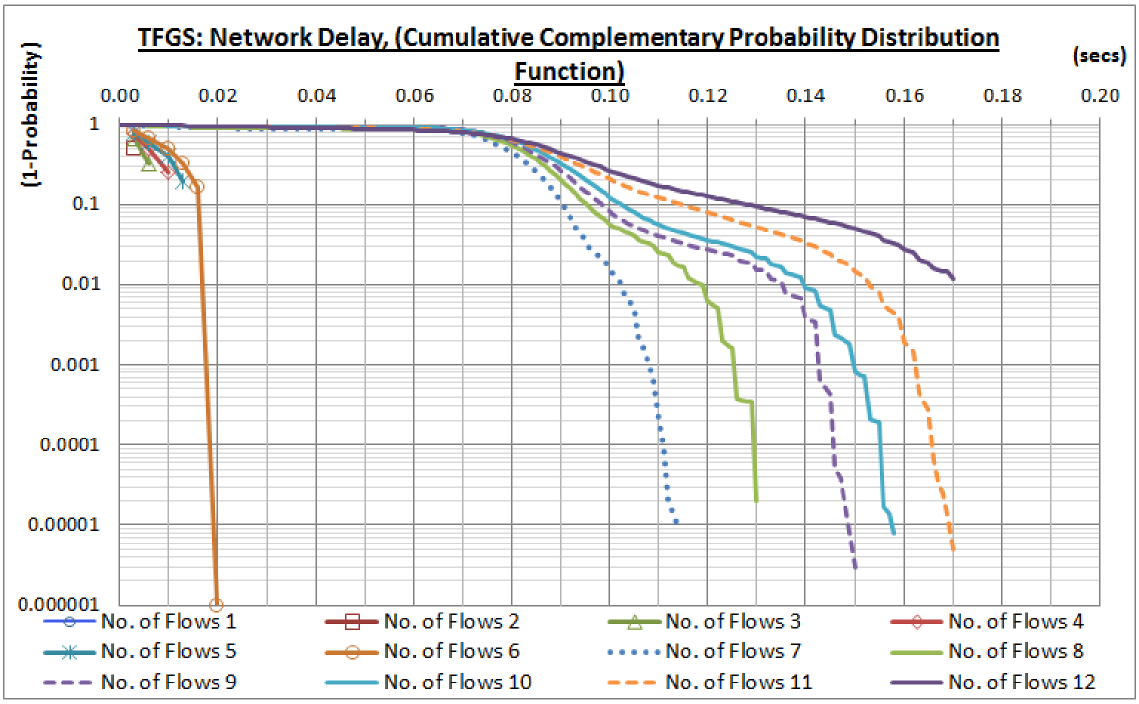


Figure 7.19, TFGS: Network Delay Probability Distribution Function (PDF)

7.1.6. Summarising the impact of Loss, Delay, and PS Truncation with respect to Quality of Experience (QoE)

Figure 7.20 and Figure 7.21 draw a comparison between the two CCMs in the context of delay and Figure 7.22 and Figure 7.23 do the same in the context of loss. Figure 7.4 illustrates the Packet Size truncation.

When looking at the 'total delay excluding SB' and 'total loss excluding SB' in (Figure 7.21 and Figure 7.23) the figures prove that the main difference between the two CCMs is due to SB component. Otherwise the performances in terms of delay and loss are approximately identical.

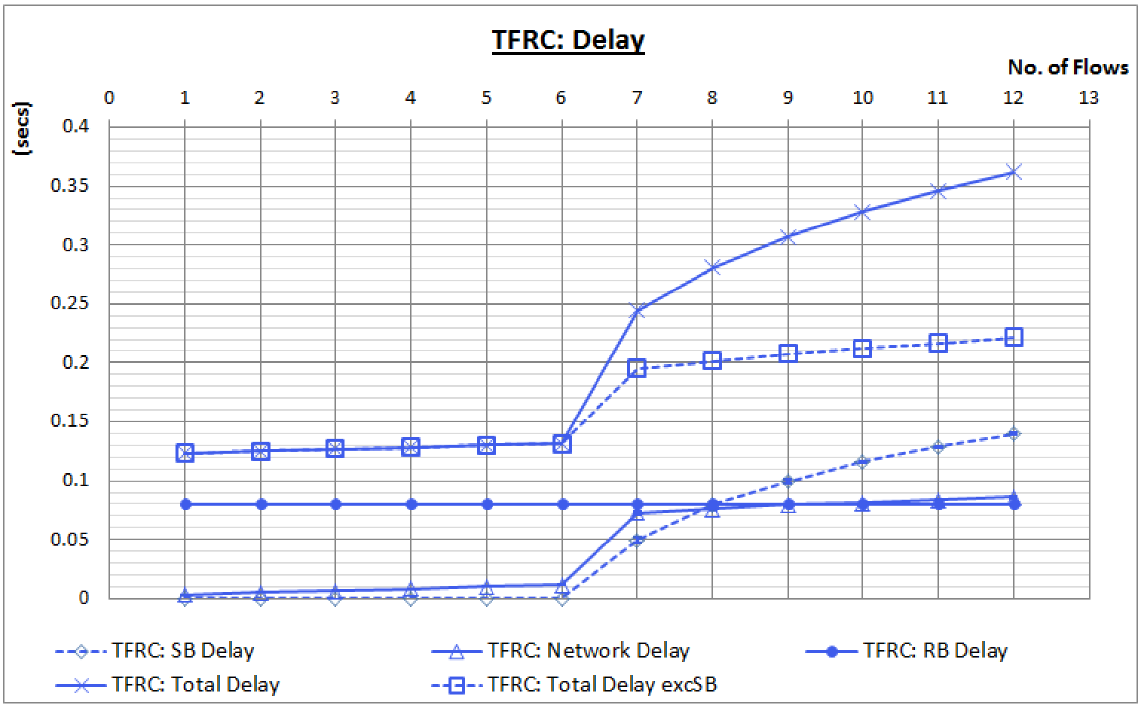


Figure 7.20, TFRC Delay

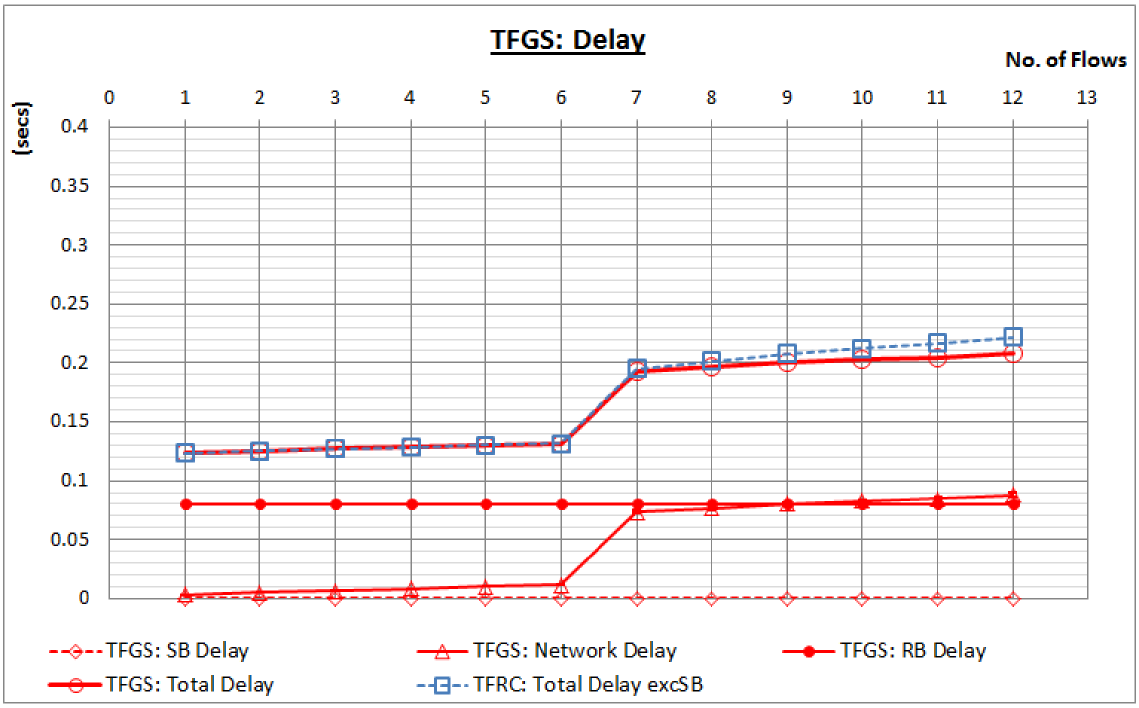


Figure 7.21, TFGS Delay

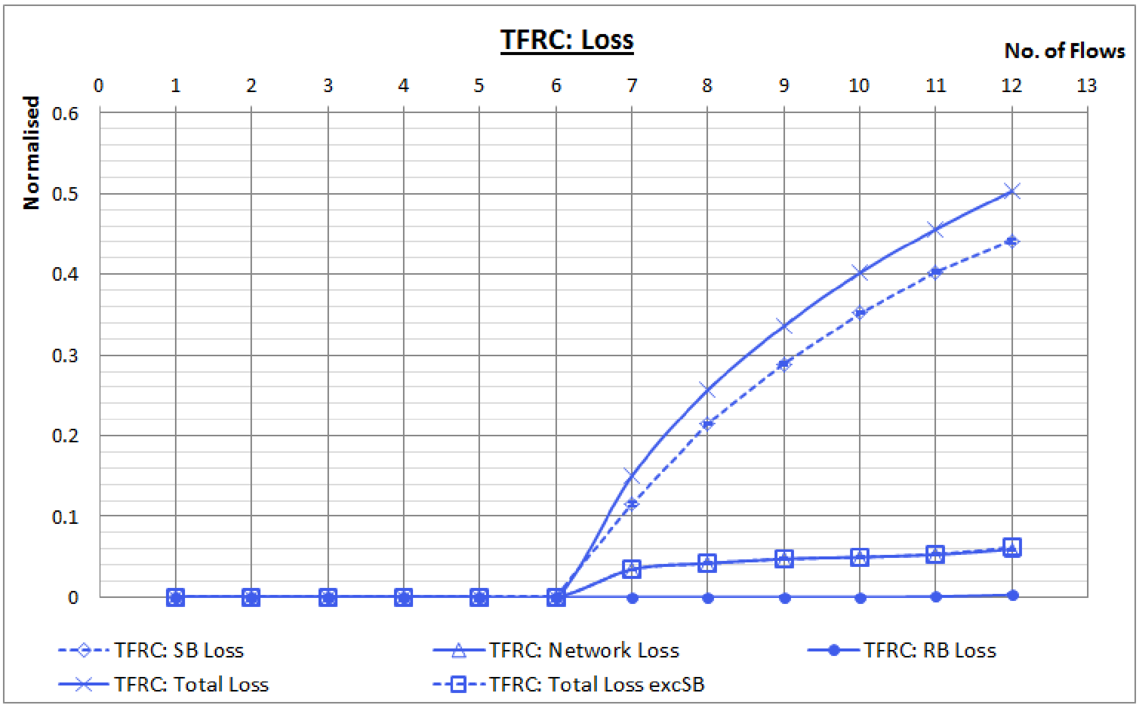


Figure 7.22, TFRC Loss ratio

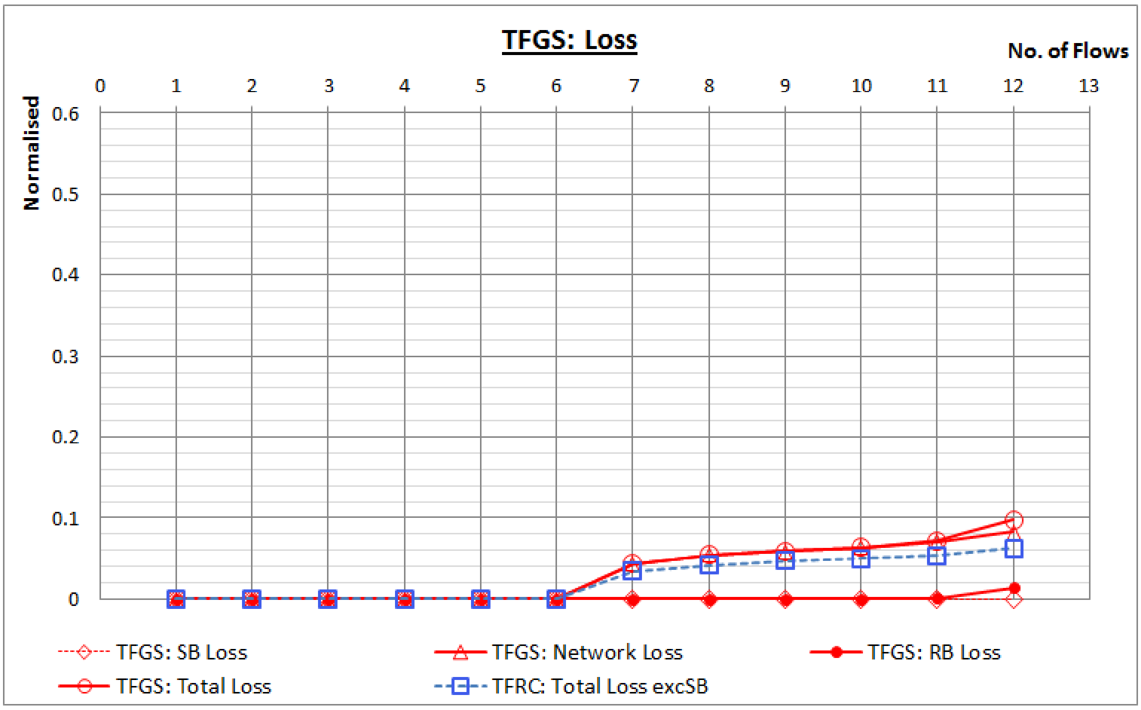


Figure 7.23, TFGS Loss ratio

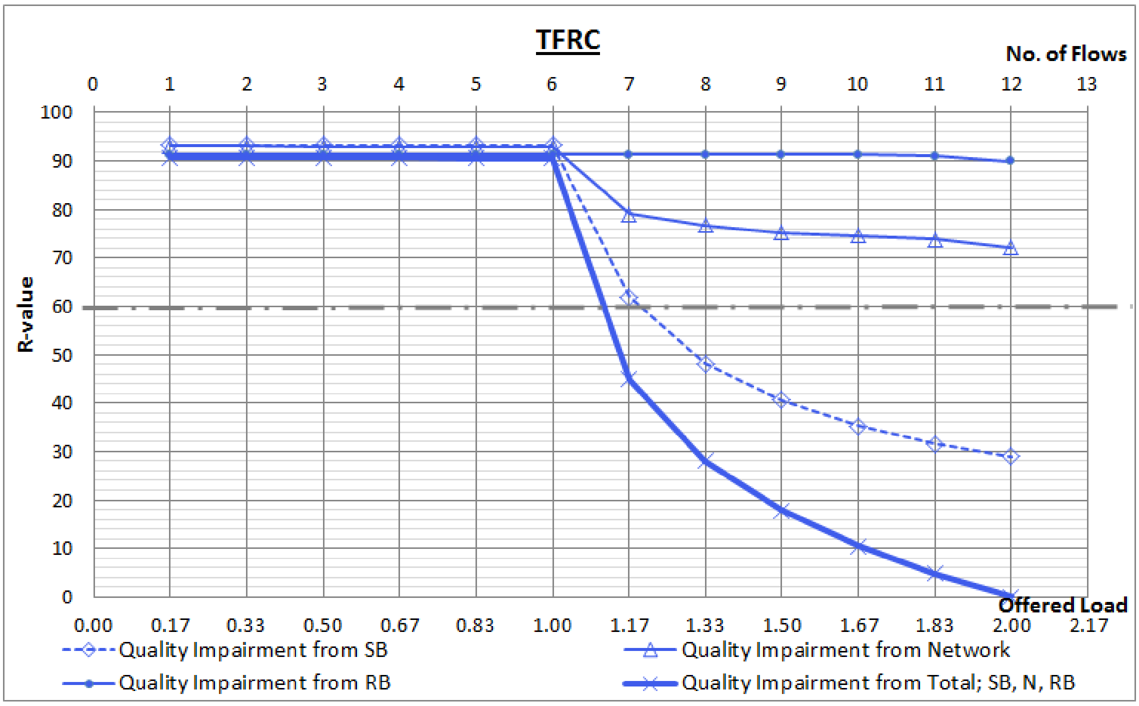


Figure 7.24, TFRC Quality Impairment

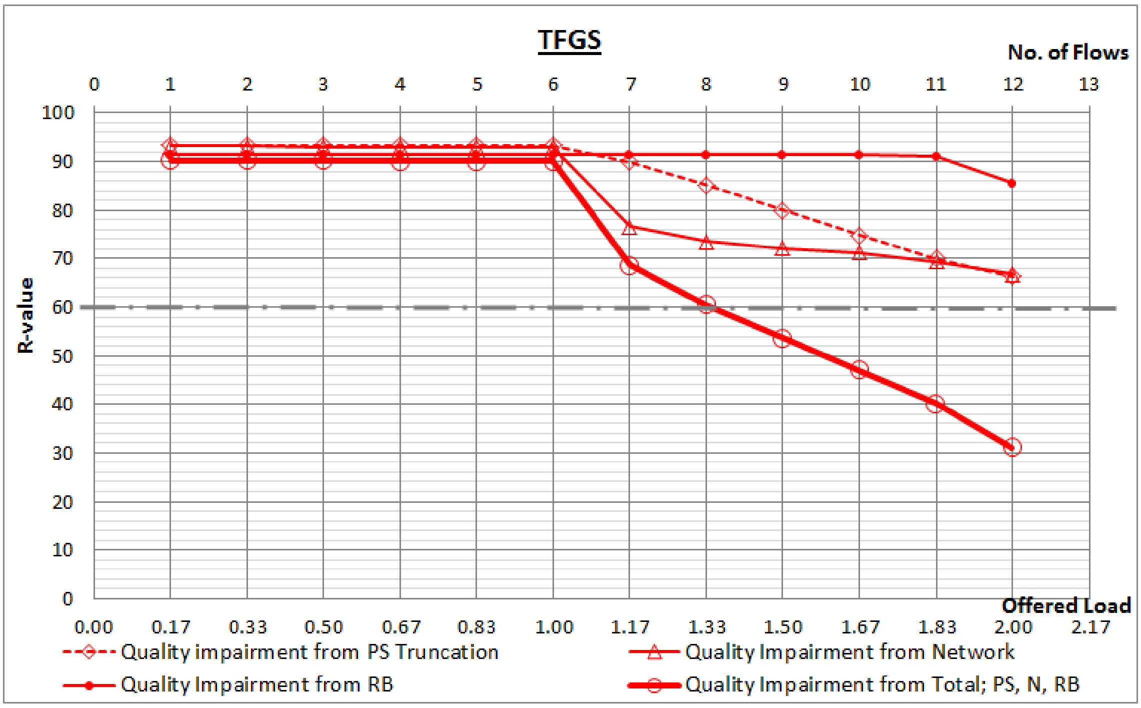


Figure 7.25, TFGS Quality Impairment

Figure 7.24 and Figure 7.25 illustrate the overall comparison of the two CCMs in terms of Sender, Network, and Receiver side quality impairment, and the net combined impairment. Figure 7.26 combines both sets of finding into one figure.

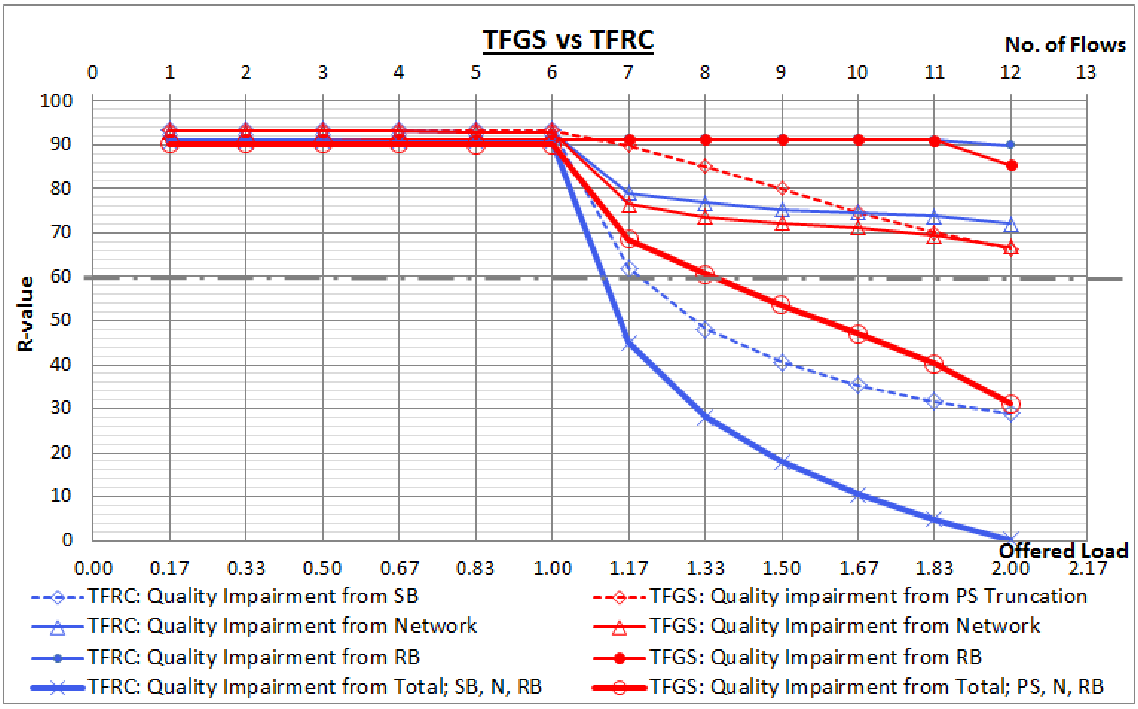


Figure 7.26, TFRC & TFGS Quality Impairment

Figure 7.26 illustrates the end-to-end QoE the flows experience when they operate over either TFGS or TFRC CCM. All flows, up to and including 6, running over the TFRC CCM are able to achieve their minimum quality (R=60). However, in the case of the TFRC CCM, once the total numbers of flows exceed 6, all flows operating over the network are incapable of satisfying the minimum quality level. In contrast, for the TFGS CCM the quality degradation is less drastic. It takes more than 8 flows to degrade the performance of all flows operating over the network.

From a network management perspective the figure above (Figure 7.26) illustrates that when using the TFGS CCM it is possible a) to carry more flows; b) to satisfy fair-share of bandwidth utilization and c) to maintain minimum quality (R=60). For example, in the case of the TFGS CCM, when the offered input load equates to 1.33 (133 in percentage) all the flows operating over the network are able to sustain the minimum quality requirement (R=60). This means that an extra 33% of the offered load is adapted by the CCM and Network Loss (details are discussed below) before it becomes unacceptable for the user and impractical for the network to support an offered load greater than 133%, as there is no point transmitting data which will be of not acceptable quality for the end-user. In contrast, using the TFRC CCM, all flows after an offered load of 100% are unable to achieve the minimum quality levels.

It is seen from Figure 7.26 that when the number of voice flows is greater than 6, there is a drastic quality degradation when using the TFRC CCM. The total offered load of the 6 flows equates to 499,200 bps and the bottleneck (link capacity) is 499,200 bps, so as soon as the TFRC CCM functions over the bottleneck capacity i.e. goes into the period of congestion, its method of congestion response is inadequate to maintain the desired minimum quality level (R=60).

In conclusion during periods of non-congestion the TFRC operates adequately, i.e. where VoIP offered load is equal to or less than 1. However, in periods of congestion, when the VoIP offered load exceeds bandwidth capacity (i.e. a load value greater than 1) the TFRC MAA congestion response is inadequate to achieve acceptable quality for the end-user.

In contrast, with the TFGS CCM the network can support 8 flows simultaneously. This equates to an offered load (in terms of bitrate) of 665,600 bps over a bottleneck of 499,200 bps; an extra 166,400 bps equivalent to an extra 33% offered load. The network is fully utilized as shown in Figure 7.27, where the input load is approximately at 104% for the 8 flows. Of the extra 33% of offered load, 29% is adapted by the CCM in the form of Packet Size truncation, and the remaining 4% is adapted in the form of packet loss in the network, this can be calculated using the equations below.

Load Adapted by CCM= Offered Load – Input Load Equation 7.2

Network Loss (in terms of load) = Input Load – 1.00 Equation 7.3

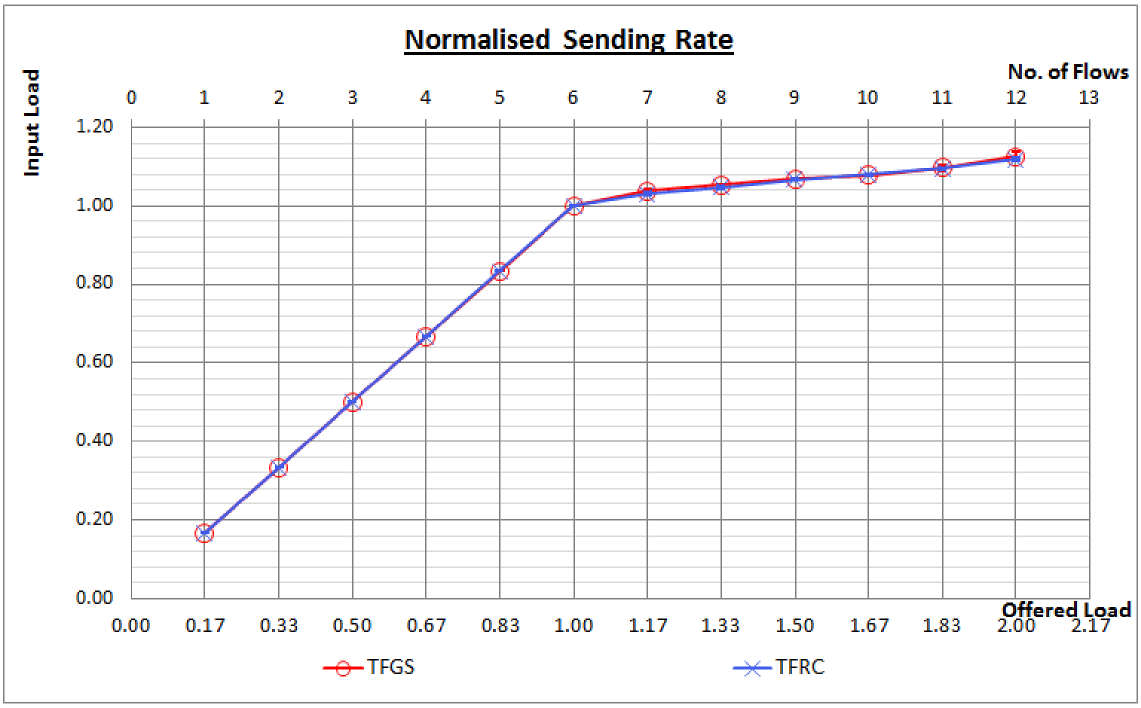


Figure 7.27, Normalized Sending Rate (NSR), in terms of input load and bottleneck Capacity (C), Equation 7.4 gives details of how the NSR is calculated

$$T_{NSR} = \frac{8 \cdot \text{Bytes Sent}}{C} \text{ x } \frac{n_{flows}}{\text{Duration}} \qquad \textit{Equation 7.4}$$

The main reason why voice flows using the TFRC CCM show worse quality than the TFGS CCM is the difference arising in the Frame Rate, F_R, (of the encoder) and the Packet Rate, P_R (of the CCM) during congestion response periods (i.e. offered load ≥ 1). Once the Packet Rate becomes lower than the Frame Rate, buffering of packets occurs at the sender. This adds delay to the packets, waiting to be sent into the network. Once the Sender Buffer (SB) is full there is inevitable packet loss. This reasoning can be verified by Figure 7.21 and Figure 7.23. Both the CCMs (TFRC and TFGS) experience approximately the same amount end-to-end loss and delay excluding the impact of the Sender Buffer (SB). However, once the impact of SB loss and SB delay are added, the difference between the two CCMs is quite significant. This can be seen when comparing the two lines in the figure. For delay, Figure 7.20: Total delay (TFRC_total) and 'Total Delay excluding Sender Buffer' (TFRC_excSB). For loss, Figure 7.22: Total loss (TFRC_total) and 'Total Loss excluding Sender Buffer' (TFRC_excSB).

In order to illustrate the comparison in impairment of the SB and PS truncation has on the total quality see Figure 7.24 and Figure 7.25, which segregate the impact of the SB component (or PS truncation) from the total impairment (inclusive of Network and Receiver Buffer impairment). It can be seen that the quality impairment resulting from the SB is far more severe with the TFRC CCM, causing a significant rate of degradation compared to that of PS truncation. Therefore, the method of congestion response used by the TFRC CCM is inadequate for interactive voice connections operating in a congested environment.

7.1.7. Improving QoE with respect to Playout / Receiver Buffer size Adjustment

Section 7.1.5 looks at Receiver Buffer (RB) loss from the perspective of network delay variation, and by using RED Active Queue Management (AQM) this limits the amount of loss experienced from delay variation. This section (7.1.7) looks at estimating the RB buffer loss in terms of inter-packet-gap-spacing (IPGS) at the receiver, and experimenting with adjusting the size of the RB to see if an improvement in QoE can be achieved. The same simulation data is used however; the post-processing of the data is changed in order to produce new set of graphs.

7.1.7.1. Flows over TFGS CCM

Using the novel TFGS CCM the frames are sent as soon as they are generated, i.e. the inter-packet-gap-spacing (IPGS) is equivalent to the fixed Frame Interval (FI) of the encoder. This can be seen from Figure 7.5 or Figure 7.28.

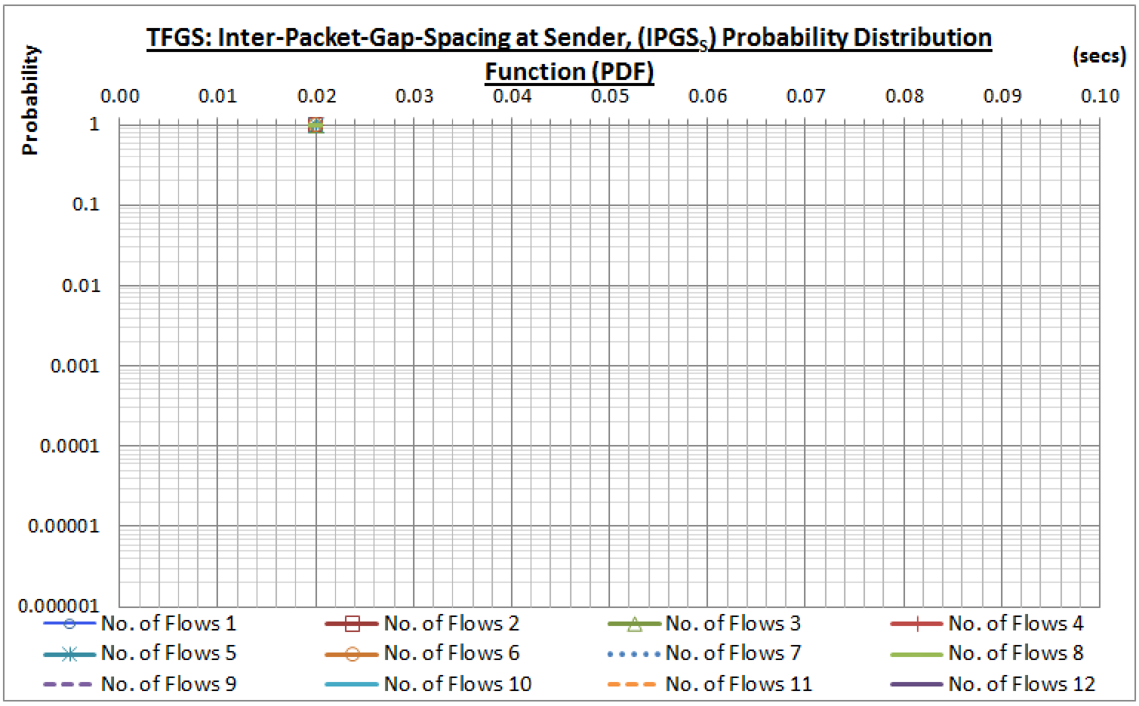


Figure 7.28, TFGS: Inter-Packet-Gap-Spacing at Sender, ($IPGS_S$) Probability Distribution Function (PDF)

The fixed $IPGS_S$ at the sender results in packets arriving within tight $IPGS_R$ at the receiver. This is illustrated by Figure 7.29 which is a Probability Distribution Function (PDF) of all packets across each flow number and of all simulation sets. In all flows the majority of the packets experience a more than 90 % an $IPGS_R$ (at the receiver) of less than 50 ms, see Figure 7.30. If the Playout Buffer (PB) size is reduced to 50ms, this would reduce the total end-to-end delay by 30 ms from the current 80 ms buffer size. The probability of loss at the PB is least significant, as it less than a probability of 1 % (0.01x100). This can be seen in Figure 7.31. The combined improvement in terms of QoE can be seen in Figure 7.32. Where the number of flows are 1 to 6 (i.e. where the offered load is less than or equal to 1), the flows can see an increase in the Quality of Experience (QoE) from its current value of at least ΔR=+1.44. Where the number of flows results in an offered load greater than 1, flows can see an increase in QoE of up to R=+3.22. For example if the PB size is reduced to 50 ms where the offered load is

1.33, (equivalent to 8 number of flows), the flows will see an increase in QoE of ΔR=+2.43 from its current QoE value at a Playout Buffer size of 80 ms.

The current QoE value is R=60 at PB size of 80 ms for 8 number of flows (see Figure 7.33). This is on the borderline of acceptable user quality, an increase of R=2.43, achieved by reducing the PB size to 50 ms, gives a new QoE value of R=62.43. This can successfully guarantee that all flows will experience an acceptable user quality (R=60).

This increase is however isn't significant enough to increase the quality from acceptable (R=60) to good (R=70), but the improvement is important in the context of Network Management where this can indicate to Network Providers that more successful VoIP flows can be carried at current network capacities.

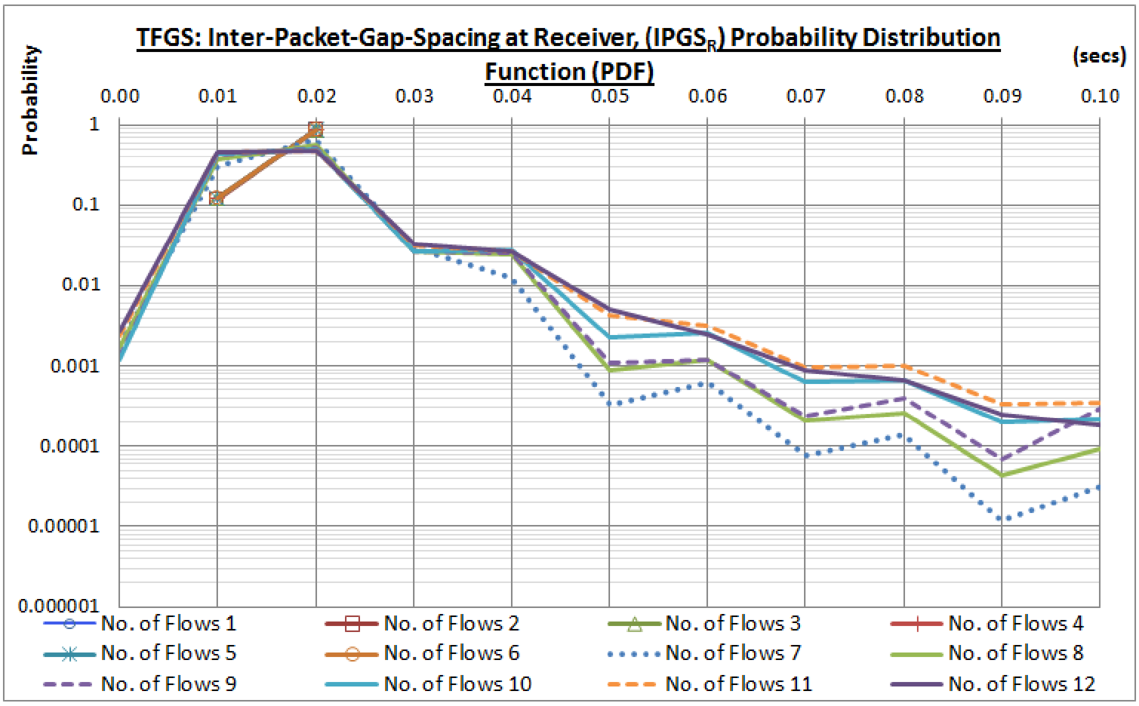


Figure 7.29, TFGS: Inter-Packet-Gap-Spacing at Receiver, ($IPGS_R$) Probability Distribution Function (PDF)

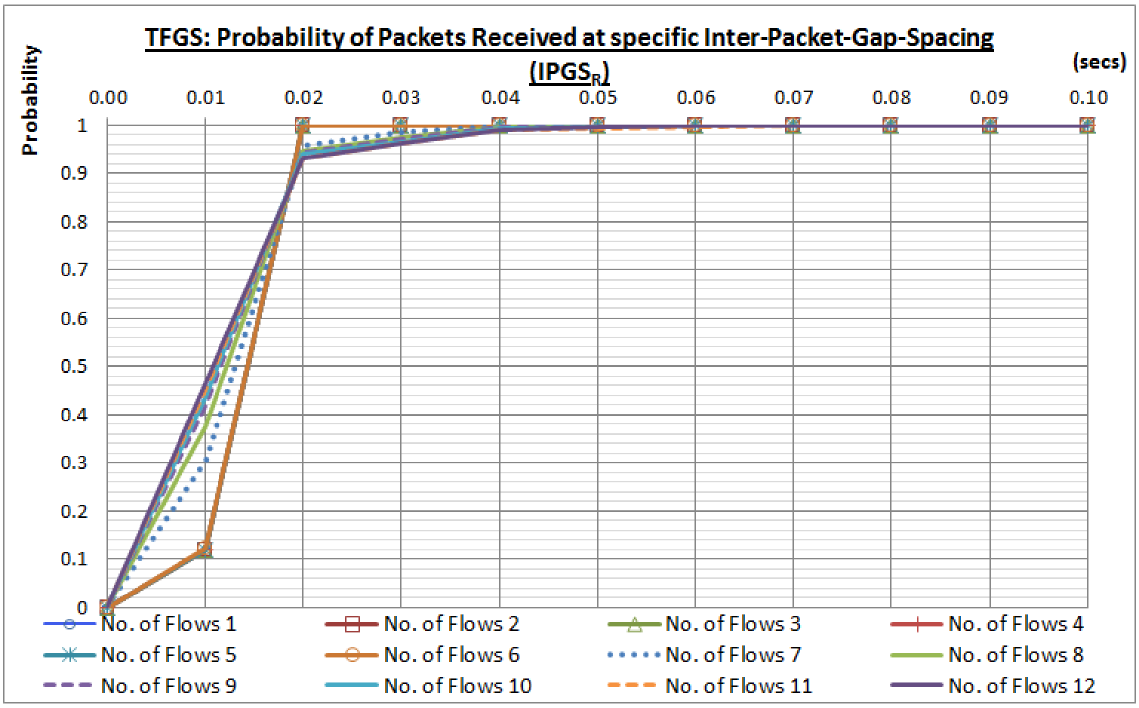


Figure 7.30, TFGS: Probability of Packets Received at specific Inter-Packet-Gap-Spacing ($IPGS_R$)

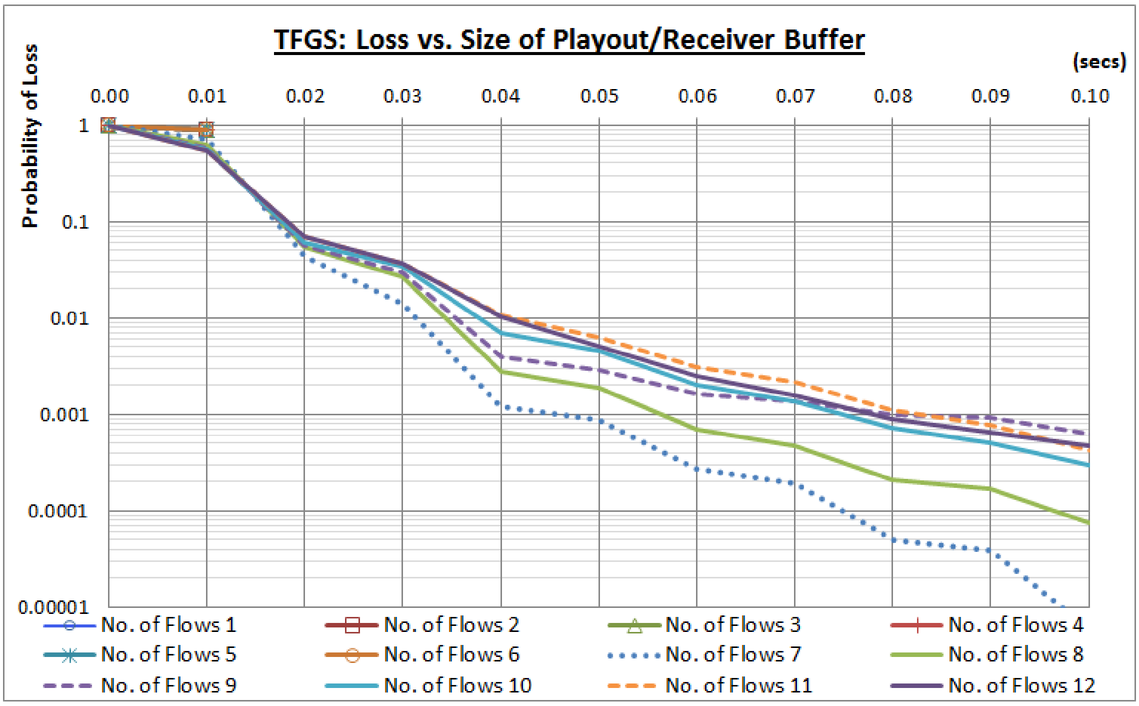


Figure 7.31, TFGS: Loss vs. Size of Playout/Receiver Buffer

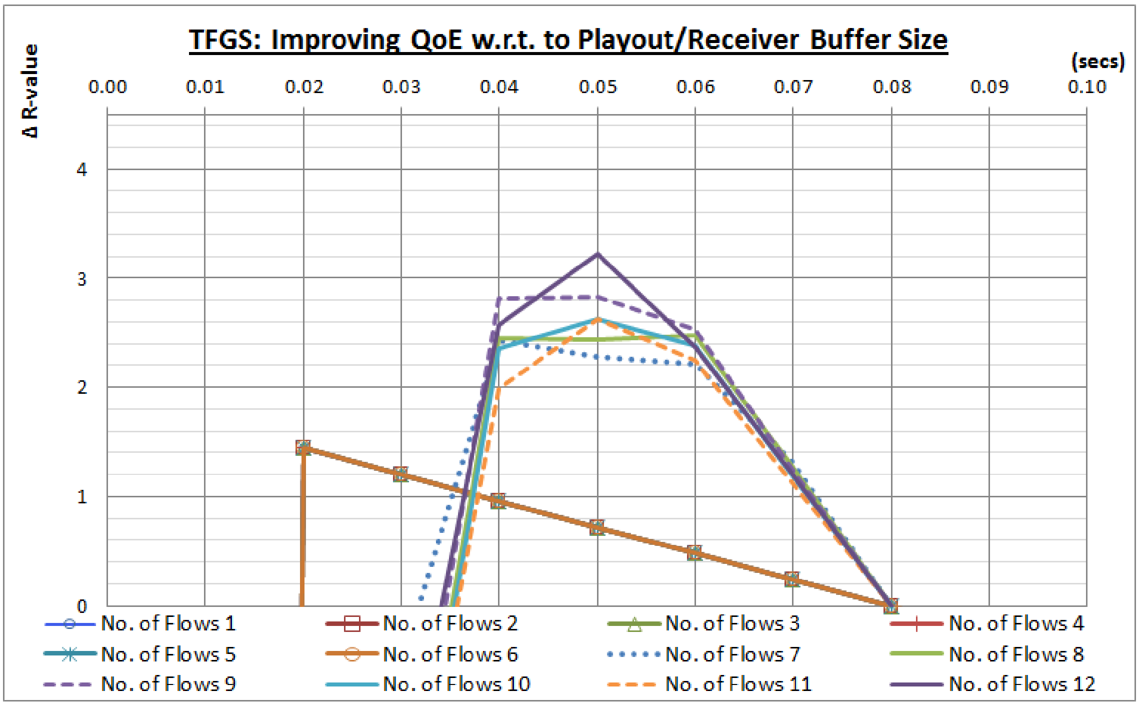


Figure 7.32, TFGS: Playout Buffer Quality Impairment in terms of Delay and Loss w.r.t. to Playout/Receiver Buffer Size

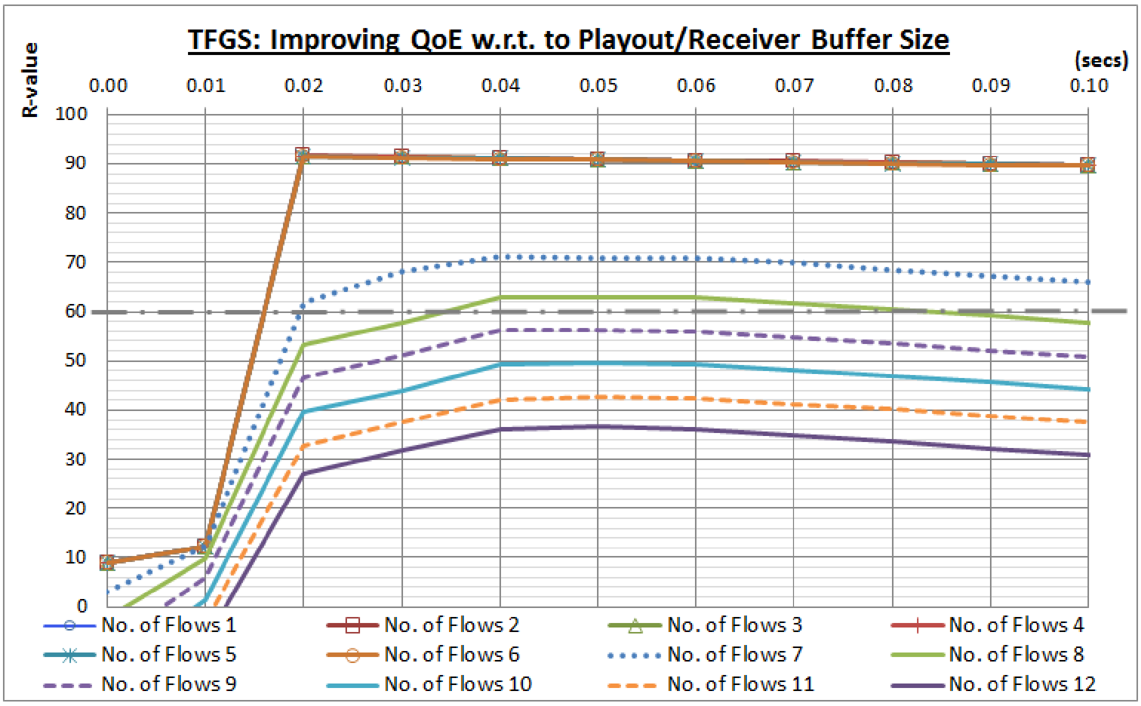


Figure 7.33, TFGS: Improving QoE w.r.t. to Playout/Receiver Buffer Size

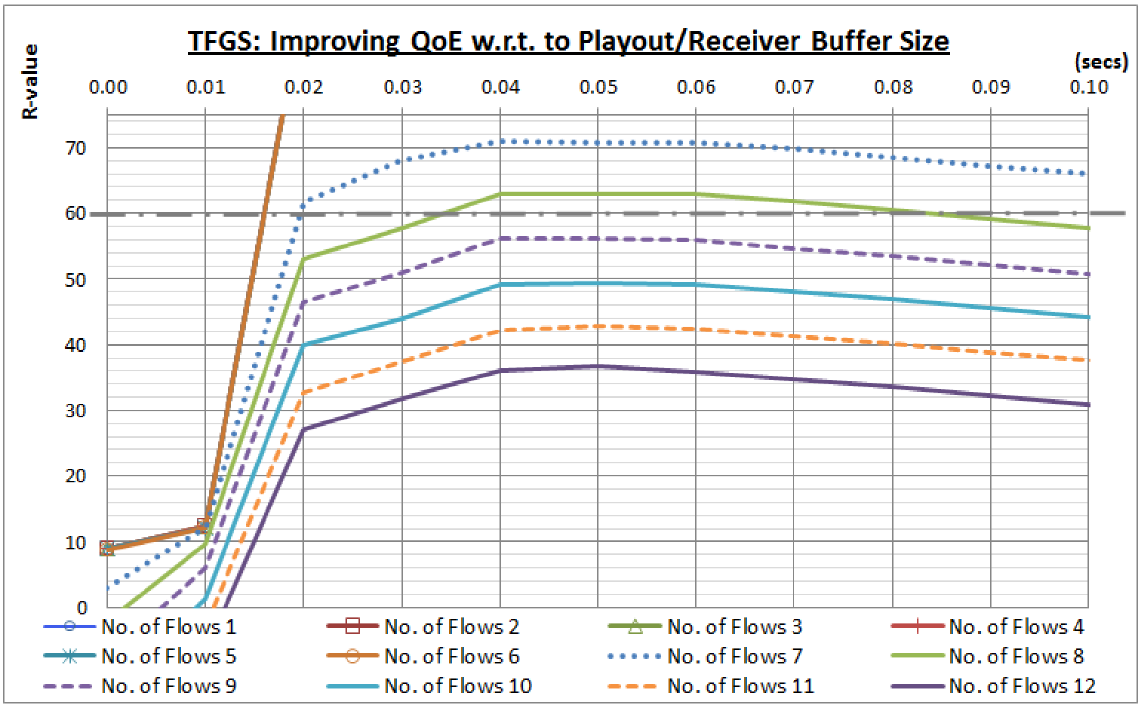


Figure 7.33(zoom), TFGS: Improving QoE w.r.t. to Playout/Receiver Buffer Size

7.1.7.2. Flows over TFRC CCM

In the case of TFRC CCM, packets are sent at varying $IPGS_S$. This is because the response to congestion is to vary the Packet Rate, PR_C, which is inversely proportional to the $IPGS_S$, (referrer to Equation 2.11).

Because the packets are sent at varying $IPGS_S$ they arrive at varying $IPGS_R$ at the receiver. This can be seen when comparing Figure 7.34 and Figure 7.35, which shows the Probability Distribution Function of IPGS at the sender and receiver respectively. The degree of variation in the $IPGS_S$ (at the sender) is proportional to the amount of congestion (i.e. the number of flows) in the network. Therefore, different amounts of $IPGS_R$ variation will be experienced by various flows; this can be seen in Figure 7.35. This demands different PB size adjustments.

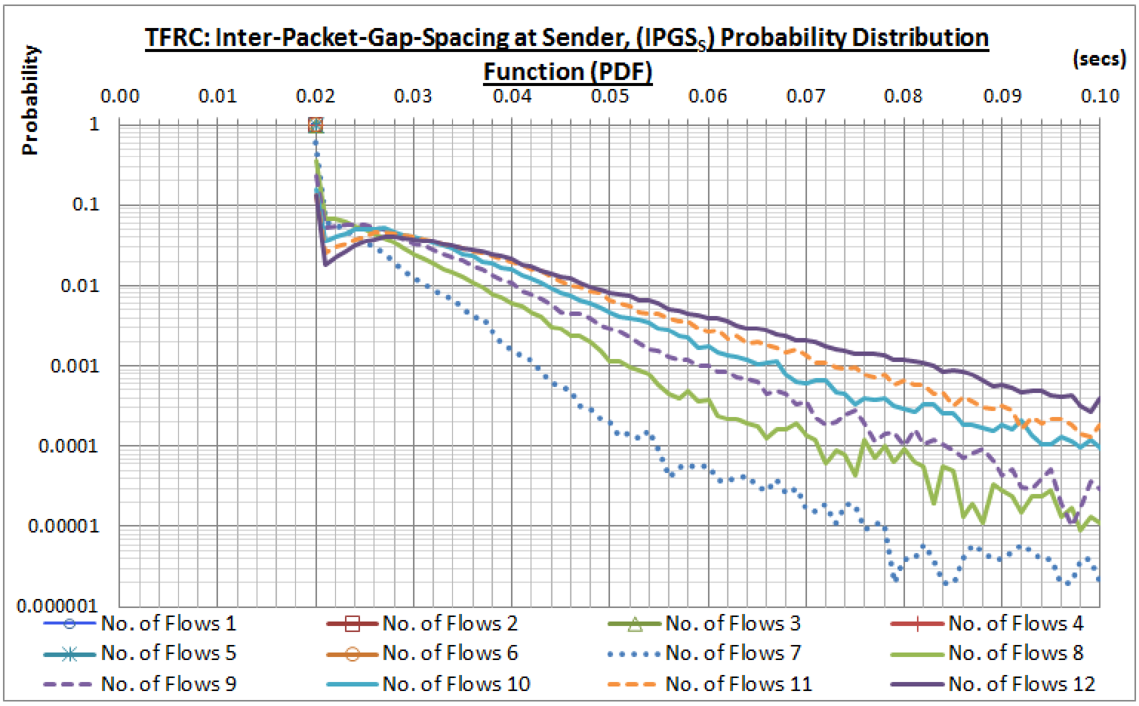


Figure 7.34, TFRC: Inter-Packet-Gap-Spacing at Sender, ($IPGS_S$) Probability Distribution Function (PDF)

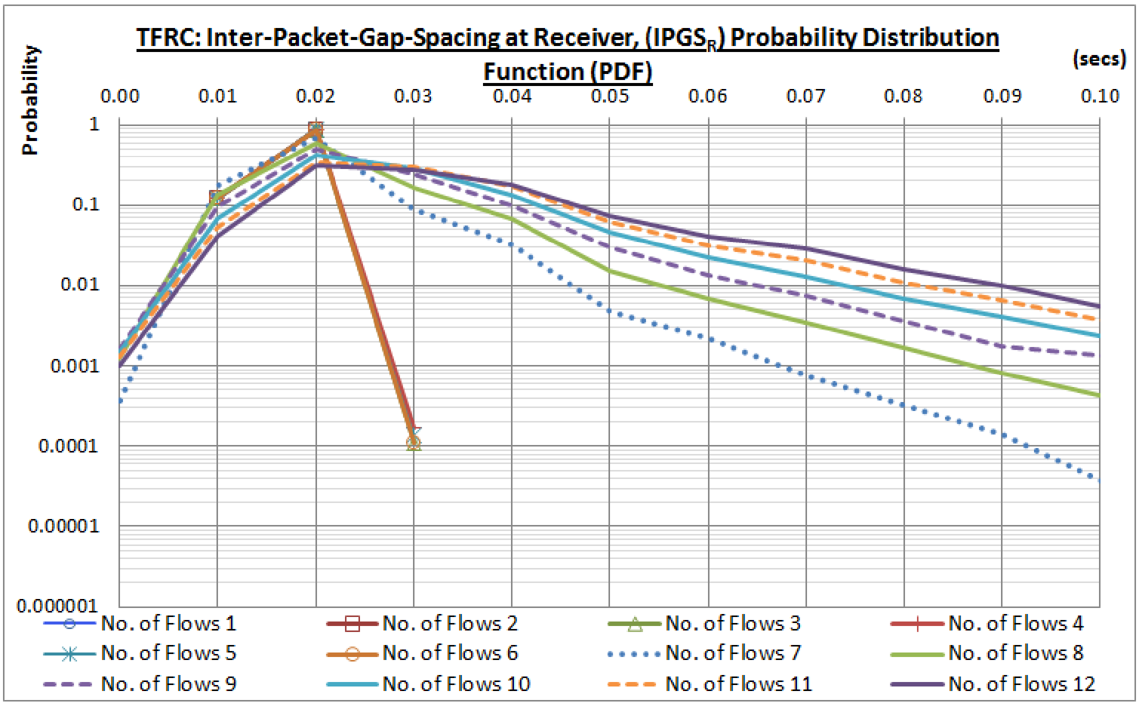


Figure 7.35, TFRC: Inter-Packet-Gap-Spacing at Receiver, ($IPGS_R$) Probability Distribution Function (PDF)

Figure 7.36 shows that reducing the PB size flows will see an increase in QoE. However, different PB sizes are required depending on the number of flows. This leads to a demand for adaptive PBs, increasing the complexity at the receiving end. Furthermore, the increase in QoE is not sufficient to push the total quality value above R=60 when the offered load is greater than 1. This can be seen in Figure 7.37 where none of the flows above 6 (number of flows) show an improvement in QoE greater than R=60 (minimum acceptable voice quality). Therefore, the benefit of reducing the PB size in order to improve the QoE is not seen in the case of voice flows running over TFRC CCM, particularly in periods of congestion when the offered load is greater than 1.

In contrast, the TFGS MAA benefits from (*a*) using smaller PB sizes, which results in an increase in QoE of the flows including those flows where the offered load is greater than 1, and (*b*) fixed PB configurations are adequate. There is no need for adaptive Playout buffering, which would increase complexity at the receiving end. (c) This increase in QoE is although small but can illustrate some improvement in QoE at current network capacities.

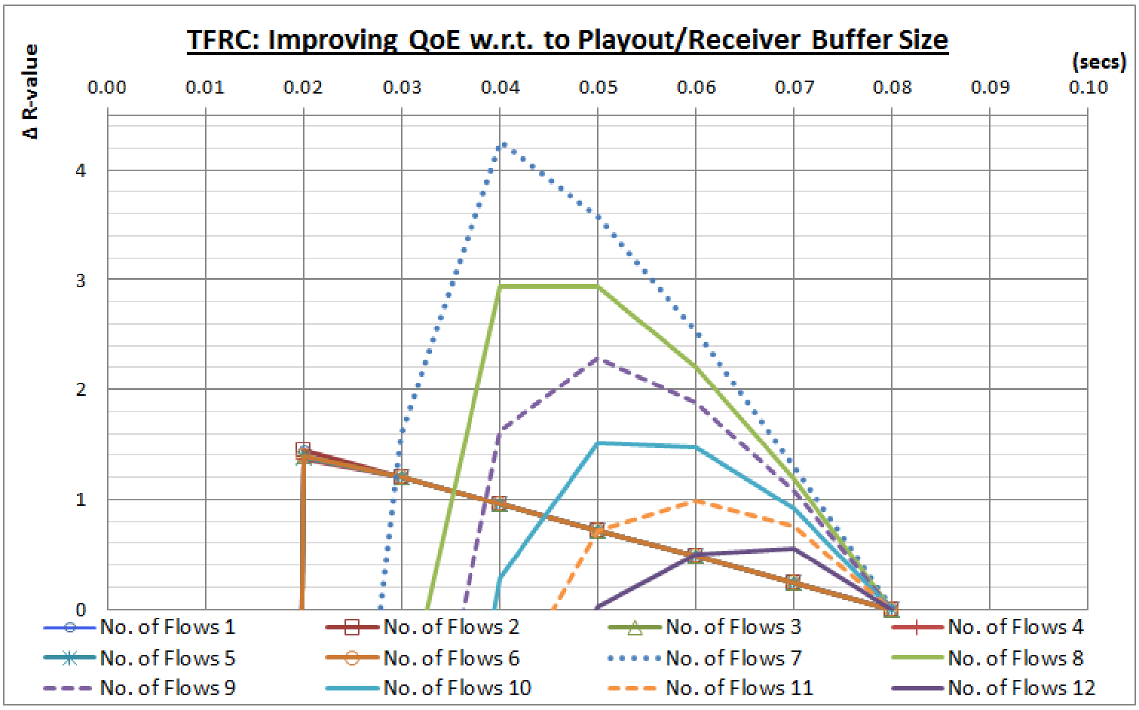


Figure 7.36, TFRC: Playout Buffer Quality Impairment in terms of Delay and Loss w.r.t. to Playout/Receiver Buffer Size

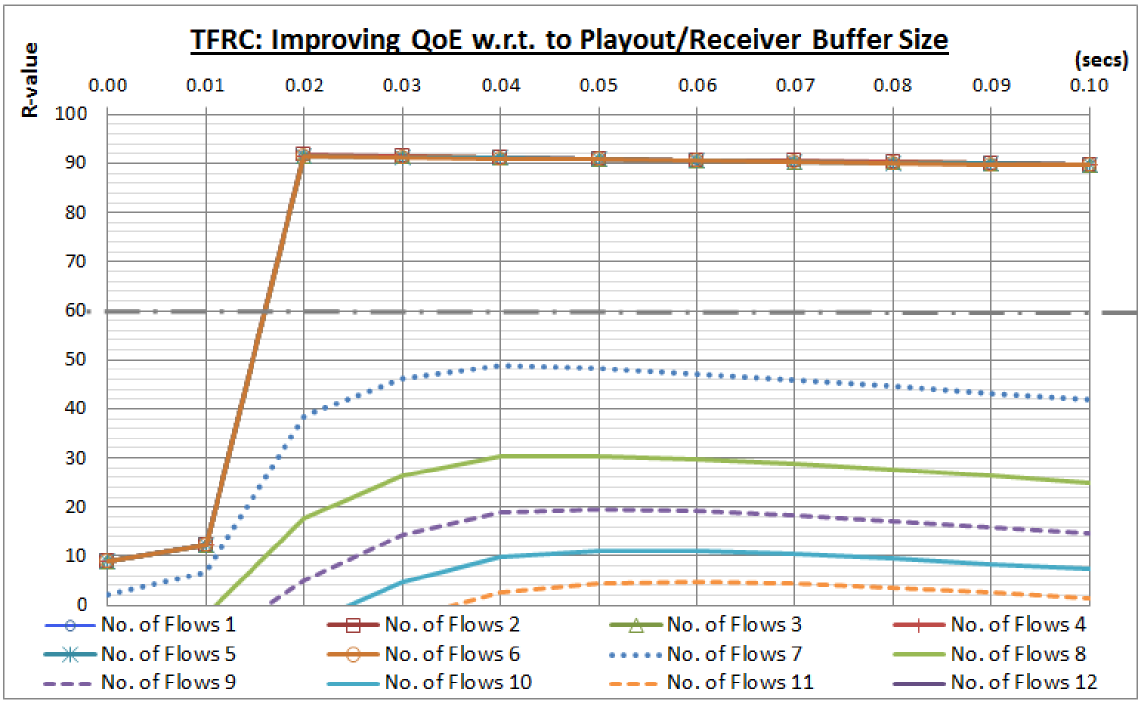


Figure 7.37, TFRC: Improving QoE w.r.t. to Playout/Receiver Buffer Size

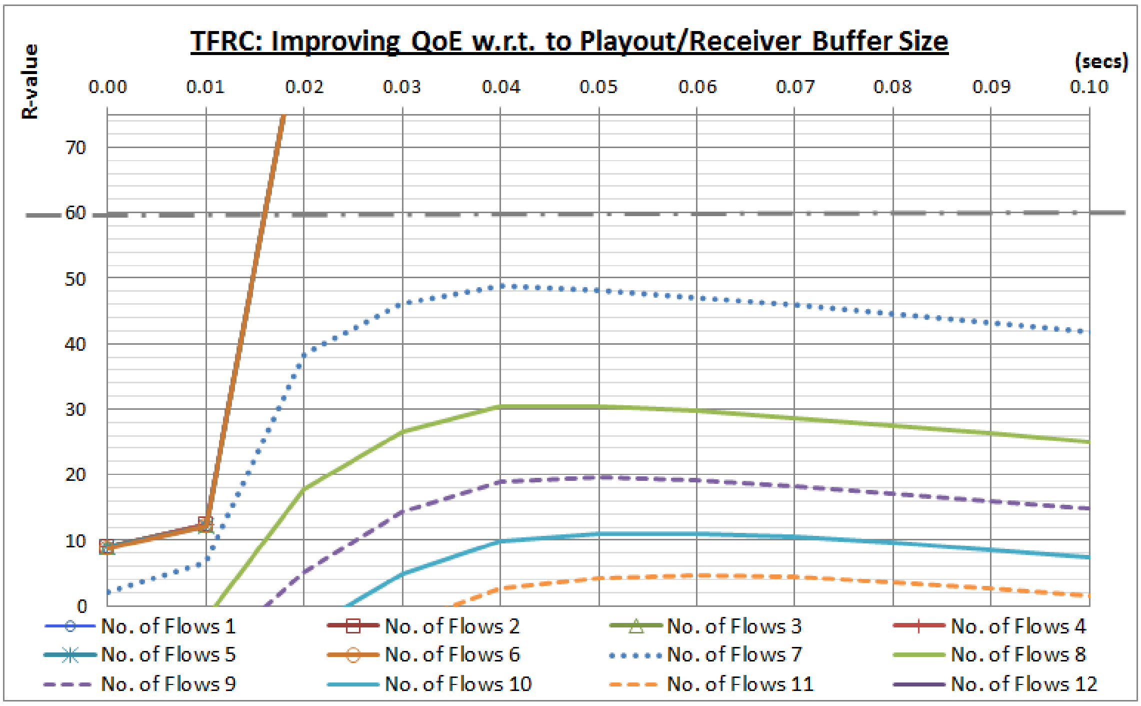


Figure 7.37(zoom), TFRC: Improving QoE w.r.t. to Playout/Receiver Buffer Size

7.2. Heterogeneous Traffic Mix

7.2.1. Network Scenario Description

The network scenario description is defined as in Table 5.4, further details can be found in section 5.2.3.

7.2.2. Congestion Response

7.2.2.1. Sending Rate (SR and Throughput (TP)):

Over a bottleneck of 998,400 bps, an equal number of voice and TCP flows are competing for bandwidth. The response to congestion is reflected in terms of the Sending Rate (SR) of the two applications; voice and TCP. Both the applications approximately follow the 'fair-share' (C/N) rate as shown in Figure 7.38. However, when the voice flows are operating around the maximum bitrate, i.e. during less periods of congestion, when there is excess bandwidth available. TCP flows make use of this excess resource. This can be seen from Figure 7.38 where the total number of flows is between 2 and 10. For example, when the total flows are 6 (which are made up of 3 voice and 3 TCP flows) the three voice flows will only be able to generate a maximum bitrate of 249,600 bps (83.2 kbps per flow) leaving 748,800 bps of bandwidth from a total of 998,400 bps to be shared among 3 TCP flows. This equates to 249,600 bps of bandwidth each for the TCP flows. This can be verified from the figure, which reflects approximately this SR for TCP flows.

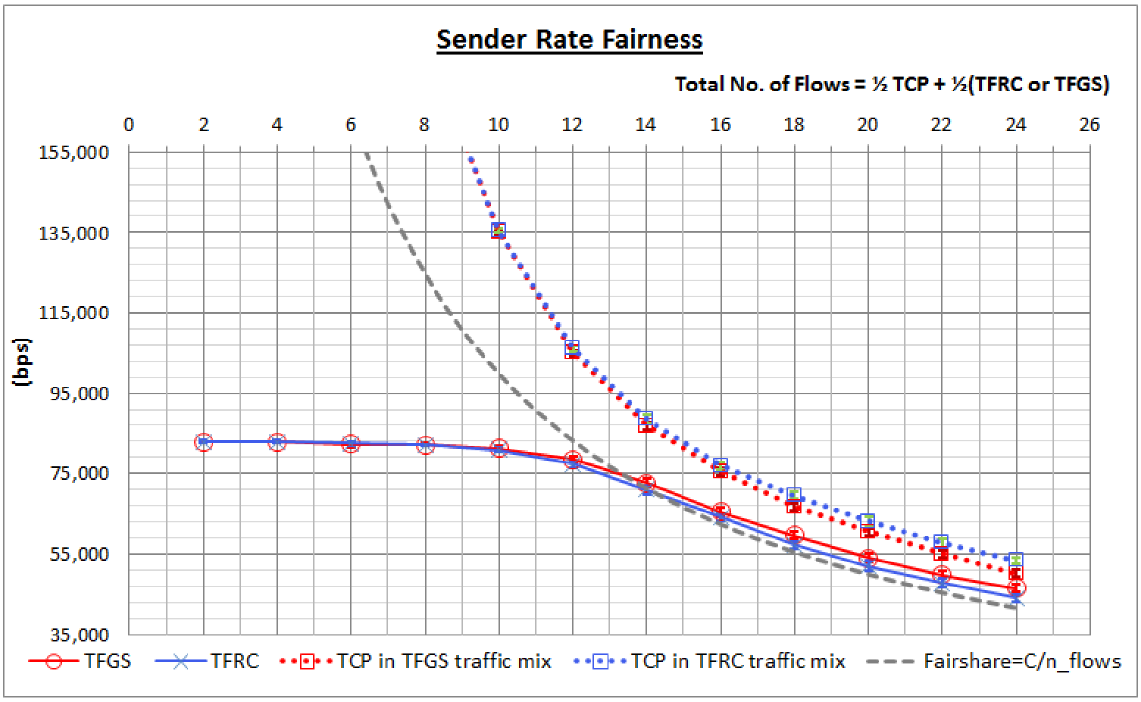


Figure 7.38(zoom), Sending Rate

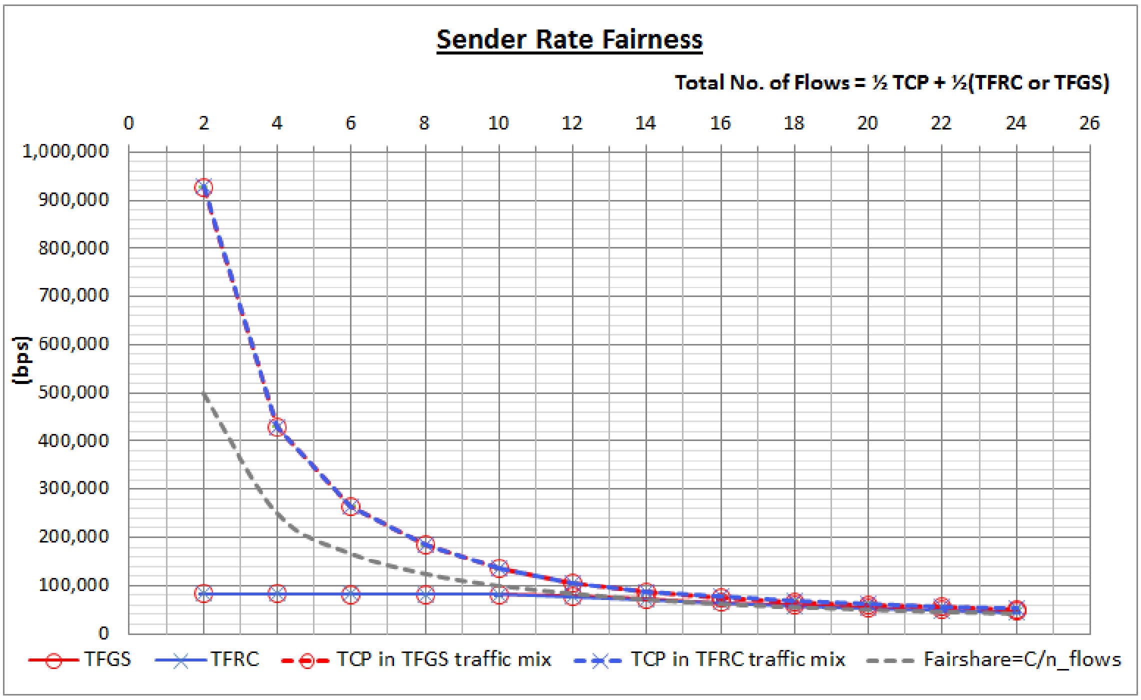


Figure 7.38, Sending Rate

Figure 7.39 and Figure 7.40 show the throughput of flows. The achieved throughput is roughly the same as the Sending Rate, indicating that the CCM's input load into the network takes into account the end-to-end constraint of the network's capacity. Hence, the CCM only injects that volume of load which the network can tolerate, avoiding as much network loss as possible (see Figure 7.41). This is the key strength of using the 'TCP-friendly' equation, which indicates the available network bitrate based on loss, delay, and other factors, see Equation 2.9 (in chapter 2). This statement also applies to TCP flows, where the Additive Increase Multiplicative Decrease (AIMD) mechanism adjusts its Sending Rate at a fair-share, based on the bottleneck capacity, see Figure 7.40 for details.

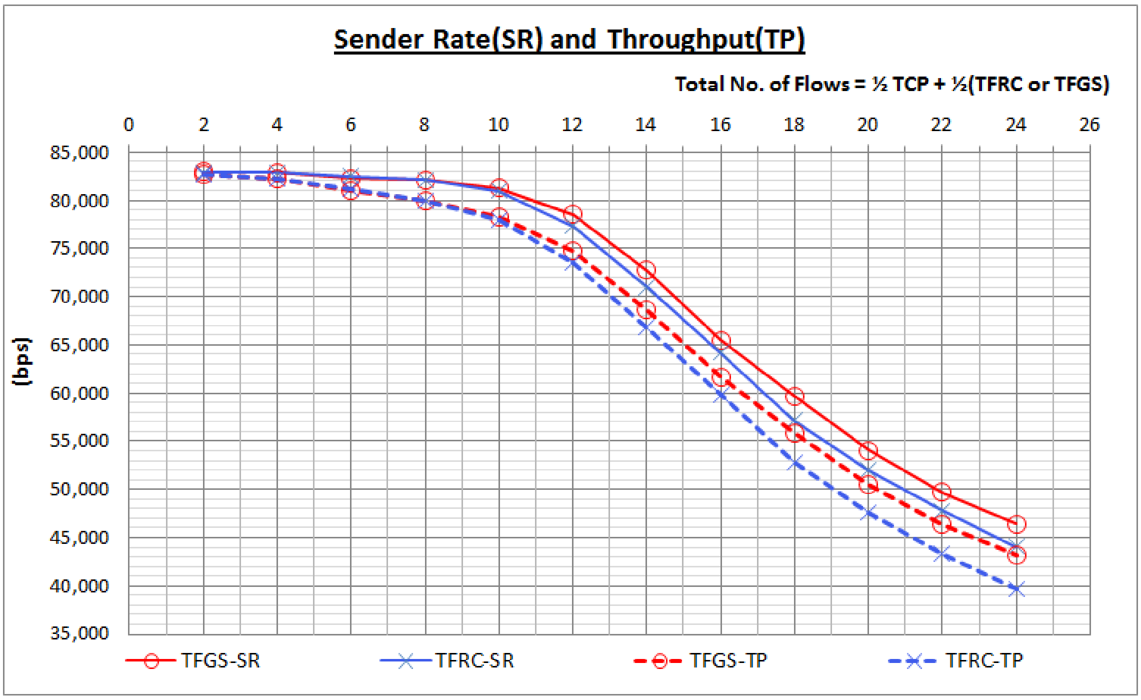


Figure 7.39, TFRC and TFGS Sending Rate and Throughput

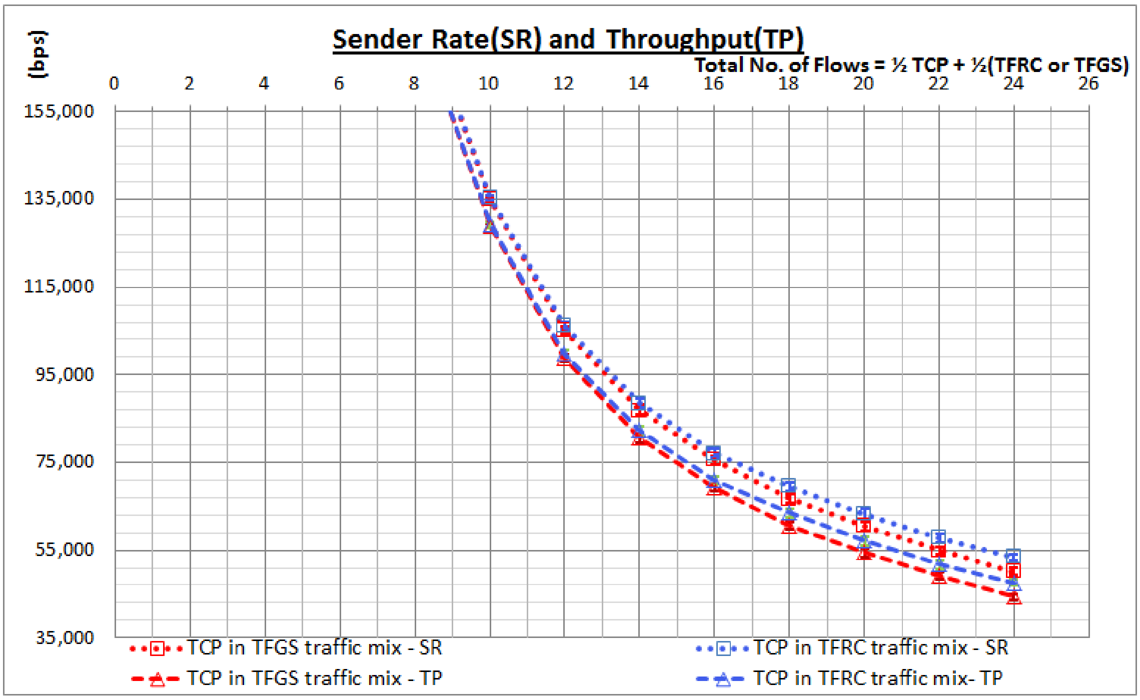


Figure 7.40, 'TCP in TFGS traffic mix' and 'TCP in TFRC traffic mix' Sending Rate (SR) and Throughput (TP)

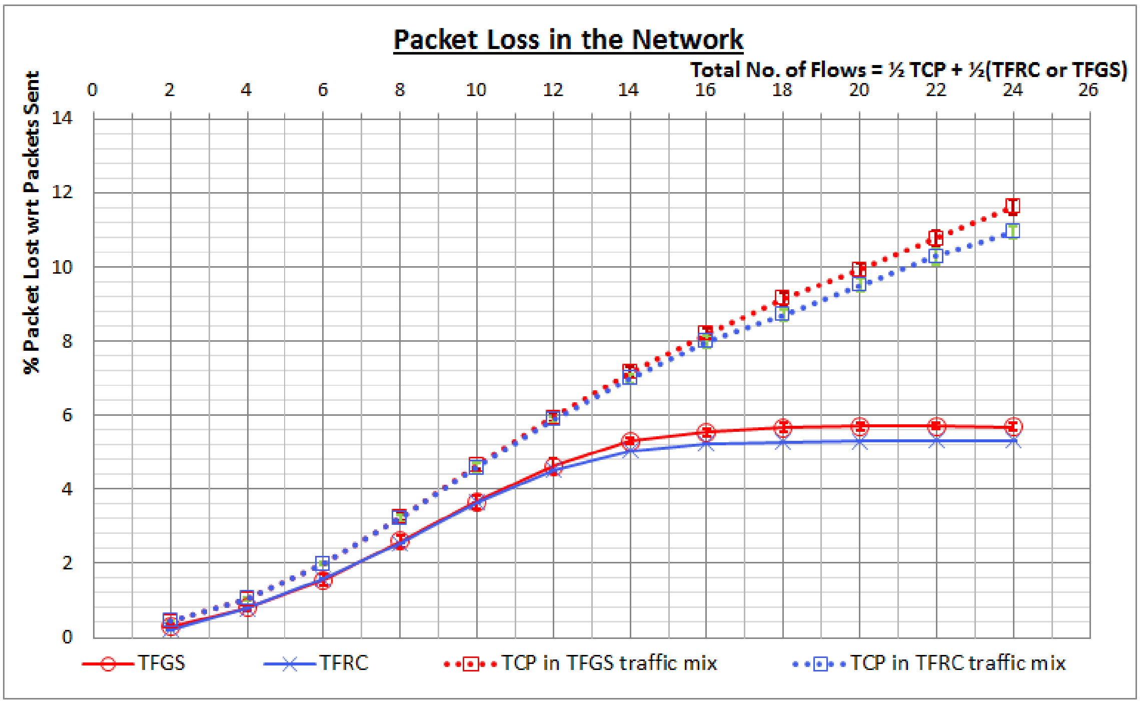


Figure 7.41, Network Loss

7.2.2.2. Quality of Experience (QoE)

Figure 7.42 illustrates the end-to-end QoE the voice flows will achieve when they operate over either TFGS or TFRC CCM. The figure shows that with 10 flows when using the TFRC CCM (half of them TCP and the remaining half voice) only 83 % of the 'voice application's offered load' achieves minimum quality (R=60). In contrast, for TFGS CCM, 8 voice flows equating to 133% offered load can sustain the minimum quality. This is of the same value seen in the Homogeneous Traffic mix. The TFGS CCM gives the same performance in terms of QoE when operating either in the Homogenous or Heterogeneous Traffic mix.

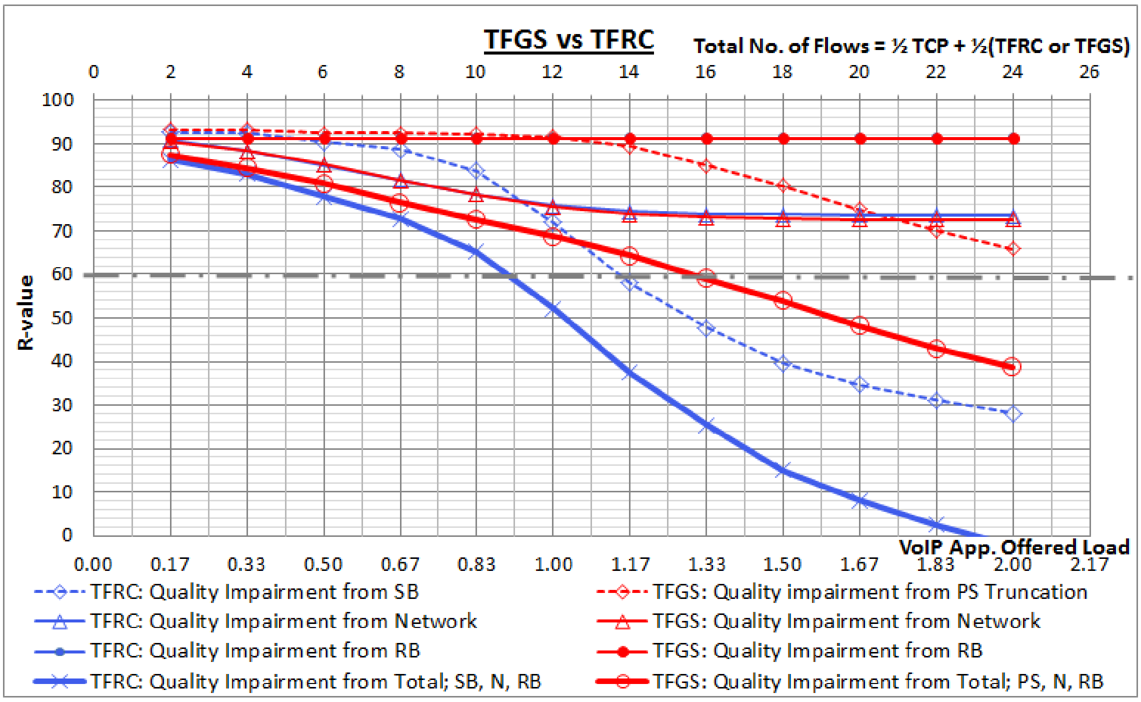


Figure 7.42, QoE of Voice flows running over either TFGS or TFRC CCM

Although the voice flows operating over either TFRC or TFGS CCM have the same Sending Rate, irrespective of whether the response to congestion is to reduce the Packet Rate, or truncate the Packet Size (see Figure 7.39). Voice flows running over TFRC experience greater quality loss impairment. This is because TFRC suffers Sender

Buffer (SB) delay and SB loss behaviour to which TFGS is immune. The SB impairment is seen at early stages of congestion, and this is because of the TCP background traffic. When the TCP traffic is using excess bandwidth, the bandwidth discovery behaviour induced by the Additive Increase Multiplicative Decrease (AIMD) can introduce packet loss in the network. With network queues configured with RED, this will randomly drop packets across all flows including TFRC flows. Therefore, these packet drops, even though small in number will cause small reductions in the available network bitrate that the 'TCP-friendly' equation calculates. Therefore, a reduced bitrate will reflect in a reduced PR_C, introducing SB delay and SB loss. This can be seen in Figure 7.43.

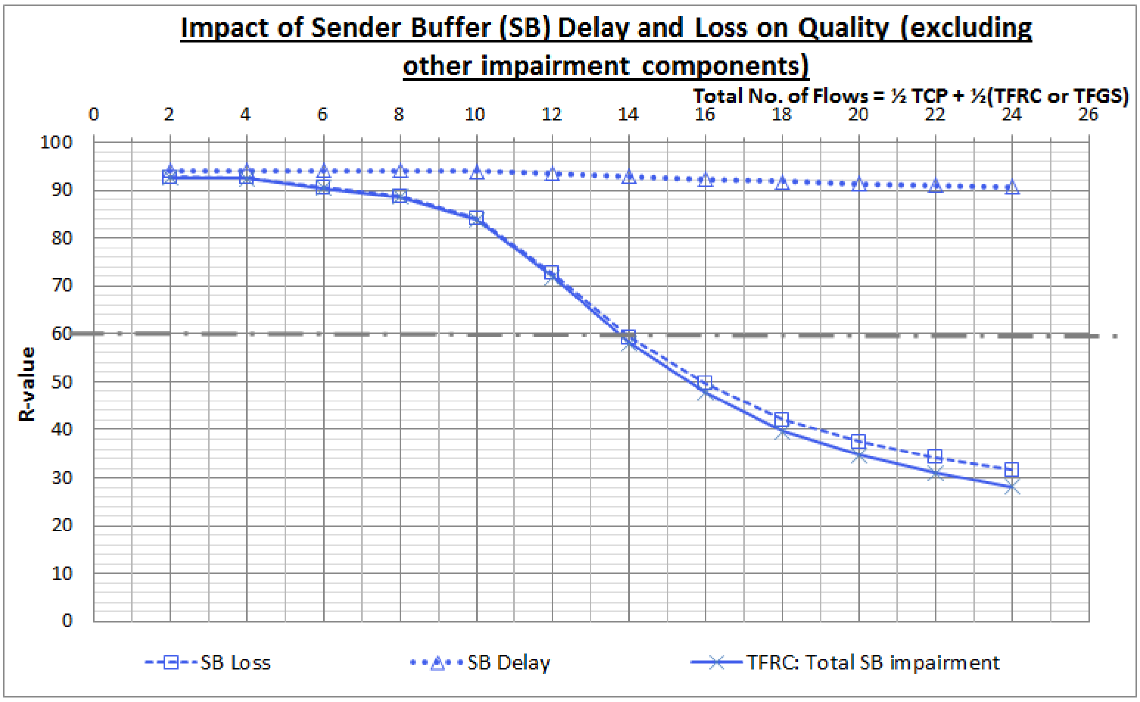


Figure 7.43, SB impairment

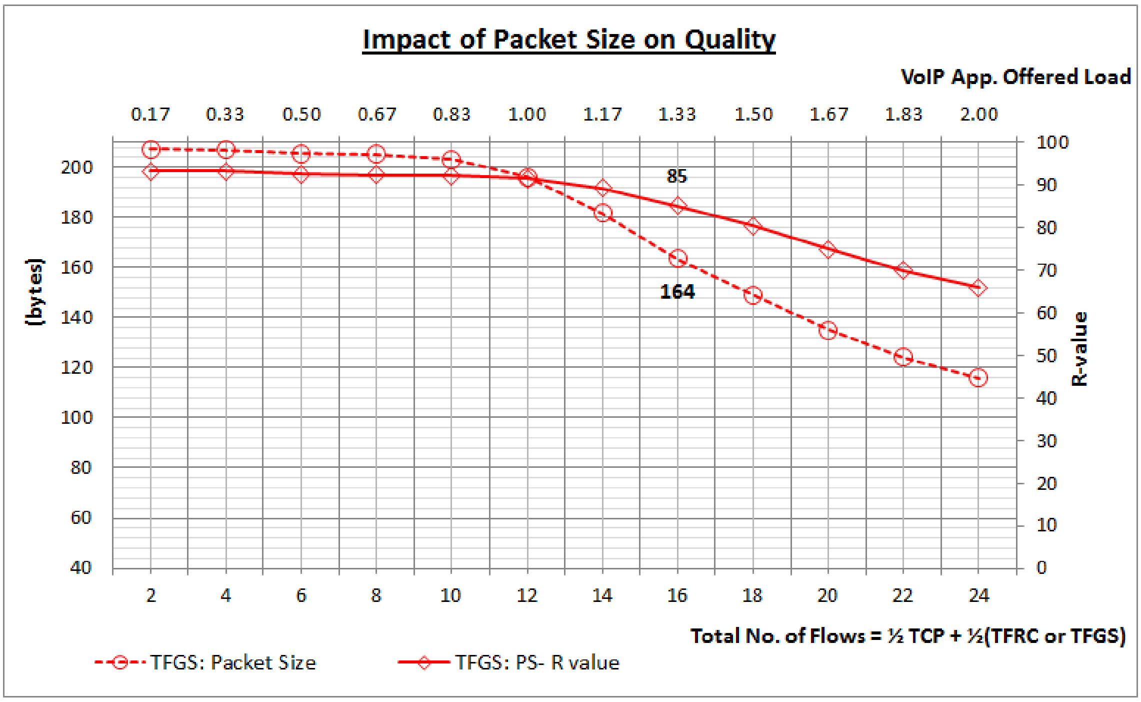


Figure 7.44, Packet Size truncation impairment

In order to illustrate comparatively the impairment that the SB and PS truncation have on the total quality, see Figure 7.45 and Figure 7.46. These figures segregates the impact of Sender, Network, and Receiver Side impairment for TFRC and TFGS CCM respectively. It can be seen that the quality impairment resulting from the SB is the main source of quality degradation, and is far more severe than that of PS impairment.

The network and receiver impairments of both the CCM are approximately equivalent (see Figure 7.42). The main difference between the two CCMs is the Sender Side impairment.

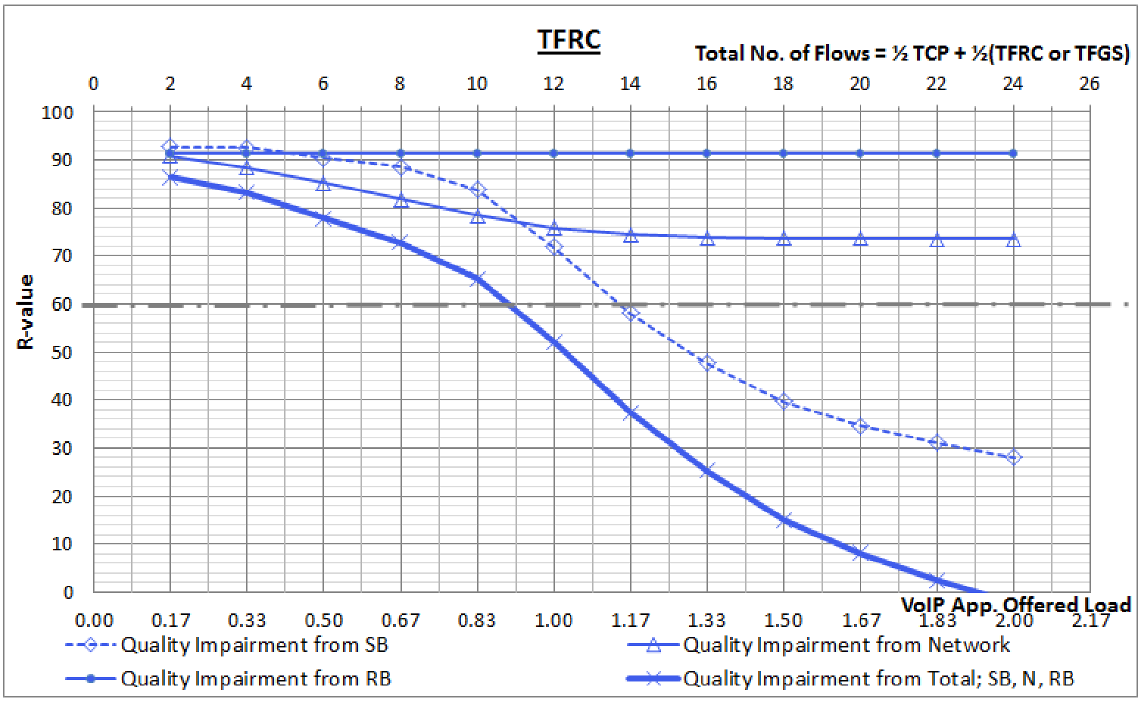


Figure 7.45, Impairment from TFRC CCM

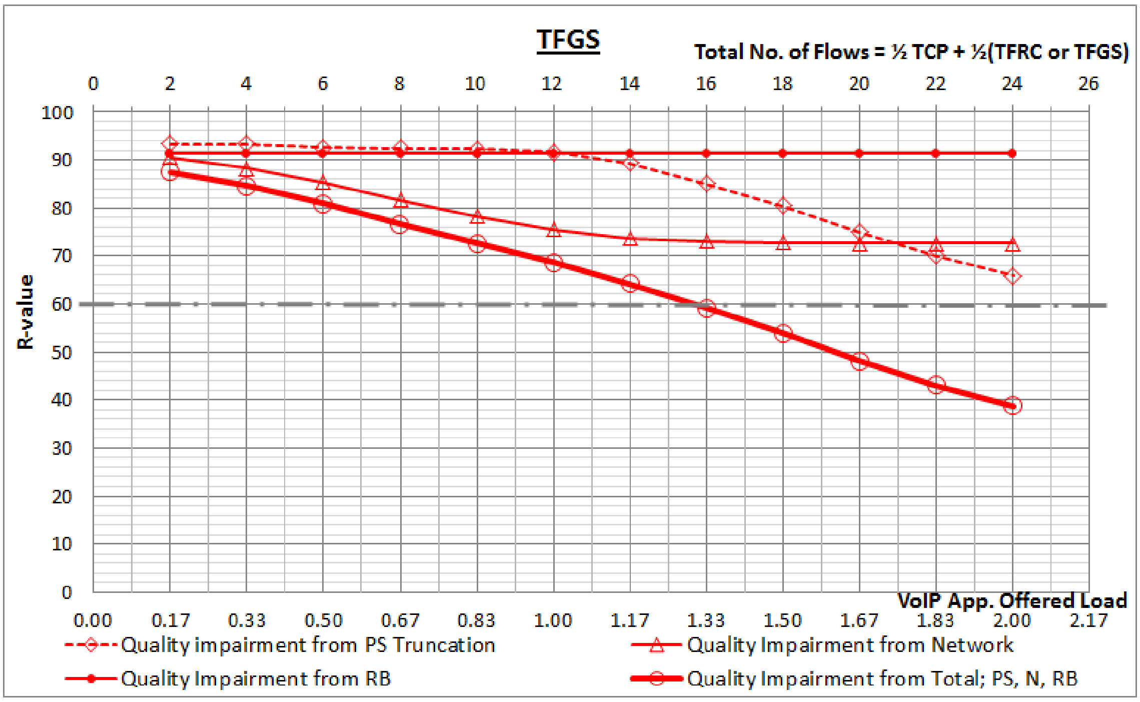


Figure 7.46, Impairment from TFGS CCM

8. Discussion, Conclusion and Future Work

The thesis has illustrated the novel Multimedia Adaptation Architecture (MAA), TFGS, for real-time interactive voice applications in best-effort IP networks, and evaluated it over a wide range of congestion scenarios. This chapter draws some concluding remarks based on the developed theory and experimental studies presented in previous chapters. Then the proposed solution is further discussed and its beneficial features are demonstrated with comparison to other existing approaches in the literature. Possible limitations are also discussed. Finally, this chapter points out areas for future work and potential research directions.

8.1. Analysis of Core Evidence

The thesis is structured in a manner to investigate and quantify the performance benefits of using the TFGS MAA. The first step was to make use of a quality measurement scheme which would enable the end user to quantify the impairment arising from packet delay, loss and byte loss for a voice call. This was achieved in chapter 6 where the E-model was used to enable the quantification of the QoE over a scale of 0 to 100. The loss and delay regression models of R. Cole [18] are used to quantify packet loss and delay. And a novel formulation is introduced which maps Packet Size truncation to an R-value.

Using this quality measurement scheme the functionality of the CCM is quantified where impairment is arising from Sender Buffer: loss and delay (for TFRC CCM) or Byte Loss (for TFGS CCM).

1. The impact of Packet Size truncation on quality highlights that when voice flows use the TFGS CCM, the minimum quality (R=60) can be sustained at a bitrate of 42 kbps. In contrast, the impact on quality from Packet Rate reduction when the flows use TFRC CCM is such that a bandwidth of 73 kbps is required. This is mainly caused by with Sender Buffer loss. For example a loss greater than 14% results from a Packet Rate reduction from 50 pps to 43 pps, and therefore the minimum quality (R=60) cannot be sustained. In contrast, with TFGS, a Packet Size truncation of 60% (i.e. a reduction of 60% of the Packet Size equivalent to 98 bytes lost, from a total Packet Size of 168 Bytes to 67 bytes excluding headers) can be tolerated before the quality becomes unintelligible. Hence, the rate of quality degradation is lower, and voice flows using the TFGS CCM can operate at low bandwidth requirements whilst maintaining their minimum quality.

2. The second performance investigation was a simulation study, where all the impairments were explored a) Sender side loss and delay from the Sender Buffer or Byte Loss from Packet Size truncation, b) Network loss and delay, c) Receiver Side loss and delay occurring from Playout Buffer. Using the quality measurement scheme all these impairments were quantified.

   The simulation results show that an offered load of 133% in terms of voice flows running over the TFGS CCM can be supported at end-to-end, whilst maintaining their minimum quality level (R=60). This is true for both Homogeneous and Heterogeneous traffic mixes.

   In contrast, for the TFRC CCM an offered load of no more than 100% for Homogenous traffic mix and 83% for Heterogeneous traffic mix can be supported, before quality levels drop below the minimum requirement.

Both the CCMs are TCP-friendly and use a ‘fair-share’ of network resources. Yet the TFGS CCM performs better in terms of QoE solely because the Packet Size truncation is far less drastic than Packet Rate reduction, which induces loss and delay at the Sender Buffer. This benefit is more prevalent in the Heterogeneous traffic mix (i.e. voice flows competing against TCP), where TCP keeps the network fully utilised (i.e. the network is always in a state of congestion), even when the voice applications are not generating equivalent ‘fair-share’ of input load compared to TCP traffic.

3. Using the TFGS MAA this will be of great benefit to Network Providers because it is able to carry more VoIP flows in the state of congestion or at low bandwidth requirements. It means that the infrastructure and hardware that is in place currently can still be used to support a range of applications. Furthermore, because the Congestion Control Mechanism (CCM) is incorporated within the Multimedia Adaptation Architecture, this is able to provide QoS to Multimedia Applications without introducing additional network overheads. The current network paradigm is kept unchanged. The strength of this approach is in its simplicity and, in turn, its scalability. Thus, it can be considered a scalable solution for QoS guarantees which can be deployed for Multimedia Applications without significant hurdles.

4. Using a Congestion Control Mechanism this enables ‘per flow’ resource management. And by using a ‘TCP-friendly’ based Congestion Control Mechanism, TFGS is able to achieve equal and ‘fair-share’ of network resources whilst competing with responsive flows such as TCP. This means other (non-multimedia) traffic flows do not suffer disproportionately. The simulation results conclude that with TFGS minimum end-to-end QoE can be maintained for voice flows over a wider range of congestion environments, whilst adhering

to the TCP-friendly bitrate. The minimum quality is equivalent to the call quality experienced when using a standard landline telephone.

5. Although the simulation study conducted in this thesis is over one bottleneck link, the user can expect the same behaviour over N links. This is because the 'TCP-friendly' equation is used, which takes into account loss and delay across the end-to-end path.

6. Although the proposed solution, TFGS MAA, is proved to be a scalable and efficient solution for IP QoS in converged networks, there still exist some limitations. Frame Size truncation will cause frame quality degradation although the rate at which it degrades is much lower than that caused by Packet Rate reduction through Sender Buffer loss and delay. Nevertheless, a Frame Size truncation greater than 60% (i.e. a Frame Size of 67 Bytes) will reflect a quality below the minimum requirements of R=60.

7. The mismatch of the Packet Rate of the CCM and the Frame Rate of the multimedia encoder is the main source of problems in the TFRC MAA. It causes buffering of frames and loss if the buffer becomes full. Otherwise, when the Packet Rate is equal to, or higher, than the Frame Rate the TFRC Multimedia Adaptation Architecture performance is equivalent to that of the TFGS MAA.

8. In the simulation study the Sender Buffer was not simulated rather its behaviour was estimated; this is because a recommended buffer configuration is not available from its authors. For example randomly, tail or head dropping of packets would have a different impact on the SB delay for each packet. Plus the method of dropping chosen would have a different impact on the loss impairment. A tail or head dropping method would result in consecutive losses leading to increased silence periods at the receiving end, which can significantly

degrade the QoE, leaving the end-user considerably dissatisfied with the service.

The SB delay was measured based on the inter-packet-gap-spacing (IPGS) at the sender. If the inter-packet-gap-spacing (IPGS) is equal to the Frame Interval (FI) of the encoder, the Sender Buffer is considered to be empty. Else if the IPGS is larger the Sender Buffer is considered to be full.

The SB loss was measured in terms of a ratio of the number of frames generated by the encoder and the number of packets sent by the CCM. This gave a precise measure of loss.

The method of calculation used to approximate SB delay and SB loss is in favour for the Sender Buffer loss, as it ignores consecutive losses. The current impact that loss has on QoE is much greater than that of delay. For example a Packet Rate reduction of 7 pps (from a Packet Rate of 50 to 43 pps) results in a loss ratio of 0.14 equivalent to a quality impairment of ΔR=33. Whereas, for the delay the Packet Rate results in a Sender Buffer delay of 90 ms equivalent to a quality impairment of ΔR=1.

9. Using the novel TFGS MAA the frames are sent as soon as they are generated, i.e. the IPGS is equivalent to the fixed Frame Interval (FI) of the encoder. As a result the delay and loss of multimedia frames at the sender side are completely removed. This reduces the total end-to-end delay and loss experienced by the multimedia stream. It improves the interactivity of the multimedia stream, which is vital for real-time voice applications.

   The fixed IPGS results in packets arriving within tight IPGS at the receiver. A majority of the packets arrive within an $IPGS_R$ of 50 ms in the homogeneous traffic scenario. And this is the case for a majority of the number of flow multiplexed in the network. Therefore, with the TFGS MAA the Playout Buffer (PB) can be of a smaller size, i.e. 50 ms rather than 80 ms. The PB can be safely configured with fixed size buffers with no need of adaptive PB configurations in place. This reduces complexity at the receiving end.

With the TFGS scenario one can safely reduce the Playout/Receiver Buffer (RB) size by 30 ms, without compromising on the Receiver Buffer loss significantly. This will reduce the overall end-to-end delay improving the interactivity and QoE for the end user. For example in the Homogenous traffic mix scenario, a flow multiplex where the offered load is less than and equal to 1, flows can see an increase in QoE from its current value of at least ΔR=+1.44 and more. A flow multiplex where the offered load is greater than 1, flows can see an increase in QoE up to ΔR=+3.22.

In the case of TFRC, packets are sent at varying $IPGS_S$ (because the response to congestion is to vary the Packet Rate which is inversely proportional to the IPGS) and therefore, arrive at varying $IPGS_R$ at the receiver. The degree of variation in the $IPGS_S$ is proportional to the amount of congestion (i.e. the number of flows multiplexed) in the network. Because different $IPGS_R$ variations are seen according to the number of flows multiplexed, this leads to different PB size requirements. This creates the need for adaptive Playout Buffers. This increases the complexity at the receiver and the possibility of the need for larger PB sizes.

However, the increase in QoE is not sufficient enough to push the total quality value above R=60 when the flows are operating under congestion conditions i.e. when the offered load is greater than 1.

In conclusion the TFGS MAA benefits from (*a*) using smaller PB sizes, which results in an increase in QoE of the flows including those flows where the offered load is greater than 1, and (*b*) fixed PB configurations are adequate. There is no need for adaptive Playout buffering, which would increase complexity at the receiving end. (c) This increase in QoE is although small but can illustrate some improvement in QoE at current network capacities.

10. The range of benefits offered from using the TFGS MAA need to be translated into an immediate wide scale deployment, so both the applications and network can benefit. This will preserve the Internet and reduce the risk of 'congestion collapse' as was seen in 1988 [29]. The growth in use and expectations of the Internet are high. The number of businesses that solely make profit because of the Internet are put at risk if applications using the Internet do not respond to congestion. It is in the wider interest that multimedia applications put TFGS MAA into practice, to eliminate this risk.

    Currently Skype has seen to be widely used VoIP service among users around the globe, with online users at any one time to be over 42.2 million [69]. A research paper published referred to as "An Experimental Investigation of the Congestion Control Used by Skype VoIP" made some strong concluding remarks of Skype Congestion Control [70]:

    "We have found that Skype flows are somewhat elastic, i.e. they employ some sort of congestion control when sharing the bandwidth with unresponsive flows, but are inelastic in the presence of classic TCP responsive flows, which provoke extreme unfair use of the available bandwidth in this case. Finally, we have found that when more Skype calls are established on the same link, they are not able to adapt their sending rate to correctly match the available bandwidth, which would confirm the risk of network congestion collapse."

8.2. Conclusions

Next generation communication networks will be based on the IP paradigm and will incorporate multiple network core and access technologies. In recent years, real-time applications, like VoIP have developed tremendously and there is an increasing demand for delivering services with Quality of Service (QoS) requirements over a shared TCP/IP network infrastructure. This network infrastructure was originally designed for data communication. This means adaptations of applications are needed if the current best-effort TCP/IP architecture is to be used to support a wide range of services (voice, video and data). The underlying assumption is that all types of traffic should expect to have equal access to the network resources and hence, to share them fairly [1].

The Internet Engineering Task Force (IETF) suggests that all applications carried over the TCP/IP network should incorporate end-to-end Congestion Control Mechanisms (CCM).

Real-time interactive multimedia applications in particular impose stringent QoS requirements (i.e. low delay, loss, and jitter) on the underlying IP networks, to ensure timely delivery of multimedia frames.

For real-time services large delays can mean that a large proportion of delivered packets arrive 'too-late' in time to be used by the application. These packets are effectively lost and this translates into dramatic quality degradation for VoIP services.

Recent developments in Congestion Control Mechanisms such as 'TCP-Friendly Rate Control' (TFRC) have been able to achieve 'fair-share' of network resource when competing with responsive flows such as TCP, but little attention has been paid to the end-to-end QoS requirements for applications using such Congestion Control Mechanisms, particularly in the context of real-time interactive applications.

This thesis developed a novel Multimedia Adaptation Architecture (MAA) known as 'TCP friendly rate control – Fine Grain Scalable' (TFGS). This MAA is able to maintain an isochronous service by sending frames as soon as they are generated, i.e. by maintaining the Packet Rate of the Congestion Control Mechanism (CCM) at a level equivalent to the fixed Frame Rate of the Multimedia Encoder. This eliminates delay of frames at the Sender Buffer (SB) and loss of frames when the Sender Buffer becomes full, (as experienced with the TFRC MAA) where a difference between the two rates arises in the presence of congestion. When maintaining a fixed Packet Rate, the inter-packet-gap-spacing (IPGS) of the arriving packets are of a tight range. This reduces the demand of large Playout Buffer (PB) sizes, thus reducing the PB delay, and overall end-to-end delay. The elimination of SB delay, SB loss and the reduction in PB delay all contribute to a significant increase in QoE for the voice flows.

The response to congestion is to truncate the multimedia frames, which is requested by the Packet Size Truncation (PST) of the TFGS CCM. The effective bitrate reduction is exactly the same as of the TFRC CCM, however it is done in 'bytes per unit time' rather than 'packets per unit time'. This method of response maintains the same 'fair-share' of network resources as seen in the TFRC CCM when competing with responsive flows (such as TCP).

By exploiting the flexibility of Fine Grain Scalable (FGS) encoders where the quality of a frame can be adapted (truncated) after encoding, the Multimedia Adaptation Manager (MAM) of TFGS is able to adapt the stream instantaneously 'on-the-fly', without needing to re-encode the frame. The MAM takes full advantage of this functionality by truncating the frame as requested by the PST function to 'byte-level' precision, achieving a true match to network supply, and better Quality of Service for the end user. This contrasts with the quantized levels of quality granularity when using Coarse Grain encoding. Using FGS encoders, each frame can be truncated exactly to the size requested by the PST function without needing to notify either the application or the encoder. Additionally, the decoder (at the receiver end) can fully recover the data from a truncated frame, without requiring notification from the encoder in advance.

Although the investigation study in this thesis was focused on voice services, this novel TFGS MAA is also applicable to video services. FGS video encoding techniques are available [43] therefore, the same MAA can be used, where video frames can be truncated to adjust to varying network bitrates. Again the video service will aim to achieve high QoE at the cost of frame quality degradation.

8.3. Future Work

8.3.1. Quality Acknowledgment

Quantifying the quality performance of a voice connection in terms of an R-value enables the user to measure the perceived effect on quality from a) Packet loss, and delay impairment, and b) Byte Loss from Packet Size truncation.

Most multimedia applications use the Real-time Transport Protocol (RTP) alongside the "RTP Control Protocol" (RTCP). Recent updates to the RTCP are available. These are referred to as RTCP extended reports (RTCP-XR) [71]. RTCP-XR enables the receiver to tag the R-value measurement onto the Acknowledgement (ACK) packets destined for the sender. This can enable the sender to monitor how well a voice connection is performing at periodic time intervals (here at every received ACK packet). The sender can therefore decide whether to continue with the current state of quality or terminate the call. Section 8.3.2.2 goes into detail about adding end-to-end Admission Control onto the TFGS MAA. However, experimentation is needed to demonstrate how best to terminate a connection (whether on the levels of Packet Size truncation or measure on the perceived, R-value, quality).

8.3.2. Admission Control

Looking at the results achieved for either of the two CCMs, one can conclude that after a certain threshold, all the flows operating over the network will suffer and none of them will achieve their minimum acceptable quality.

One can argue that such a scenario will lead to congestion collapse caused by stale packets where the network is busy forwarding packets which no longer are of good use to the end-user. This may be due to a number of reasons such as a) packets received have arrived 'too-late' or b) are in insufficient in numbers because of a low Packet Rate at the sender (in the case of TFRC CCM) or high loss rate in the network, c) extremely small packets sizes (in the case of TFGS CCM, PST truncation function) sent, due to limited available network bitrate in presence of network congestion.

Furthermore, one can also argue that precious network resources have been wasted in supporting such applications, when they could have been used by other applications.

Motivated by the above issues a novel Admission Control functionality is proposed for the future versions of TFGS Congestion Control Mechanism (CCM) by introducing two new functions (*1*) Start-up Phase (SP) and (*2*) Terminate (TR).

SP probes for sufficient network capacity to support the minimum requirement of the application in terms of bitrate. If the Congestion Control Mechanism (CCM) indicates that there is sufficient capacity, it signals to the applications to start. The TR function terminates the application if the minimum bitrate cannot be achieved within a pre-determined time. Furthermore, if a consistent low bitrate is indicated after the Start-up phase, the CCM may well terminate the connection and notify the application accordingly. Using these two functionalities (SP and TR) one can aim to support a limited number of applications over a congested network, so that they can aim to maintain their minimum acceptable quality requirement over the duration of the connection. Compared to a scenario where all applications are suffering equally and all are unable to achieve their minimum quality requirements. This will be useful both to the end-user and to the network in terms of better resource management.

With experimentation it will be possible to quantify the benefits of integrating Admission Control (i.e. the new Start-up Phase and Terminate functions of the CCM). This can be done by illustrating the number of connections that can operate over the network without dropping below the minimum QoS requirements irrespective of the congestion state of the network (compared to a scheme which does not employ Admission Control).

Figure 8.1 illustrates the interaction of the SP and TR functionality between the other components of the Multimedia Adaptation Architecture (MAA).

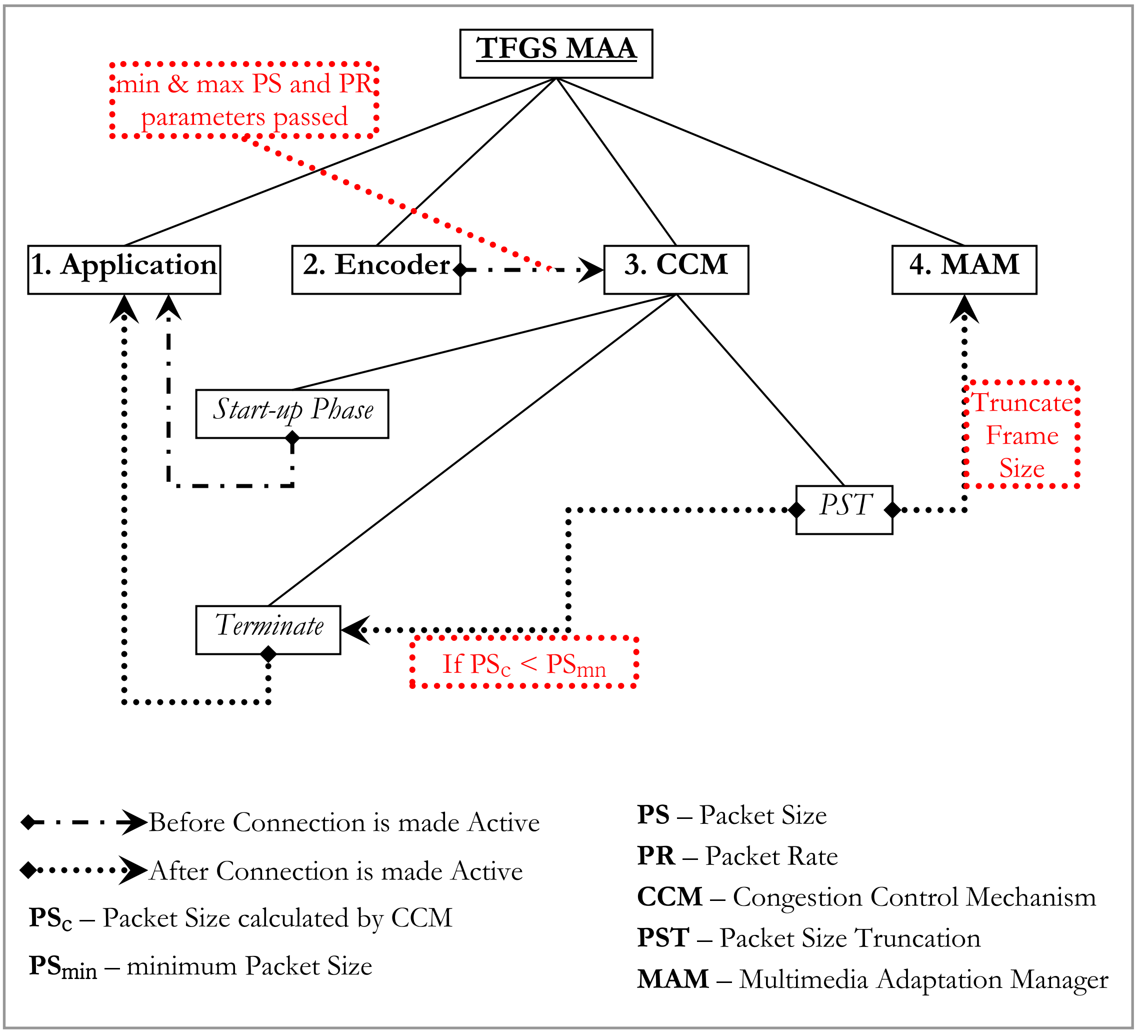


Figure 8.1, 'Components of the TFGS Multimedia Adaptation Architecture (MAA)'

These two functions; SP and TR have been adhered to by the Call Admission Control (CAC) used in the Resource ReserVation Protocol (RSVP) [72]. This occurs when routers receive reservation requests. The protocol must determine whether all the links between the source-to-destination can accommodate the traffic demands of the request. This is referred to as the admission test. If the test fails an error message is sent to the appropriate receivers.

8.3.2.1. Start-up Phase (SP)

Before a multimedia connection is made active between the end systems, the CCM will be required to probe for sufficient network capacity to support the minimum bitrate required by the application (in terms of minimum Packet Size and fixed Packet Rate).

Below is an example of how the available bitrate rate (T_{TCP}) indicated by the CCM is interpreted by the Start-up Phase function. The minimum (effective transmission) bitrate, $T_{ETR(min)}$, is a product of the minimum Packet Size, $P_{S(min)}$, and fixed Packet Rate, $P_{R(fix)}$, as shown in Equation 8.1. During the Start-up phase dummy packets are sent. These packets are maintained at the minimum Packet Size, $P_{S(min)}$, and the Start-up Phase function probes for the fixed Packet Rate, $P_{R(fix)}$.

$$T_{ETR(min)} = 8 \cdot P_{S(min)} \times P_{R(fix)}$$ *Equation 8.1*

Equation 8.2 indicates the Packet Rate (PR_C) of the CCM as a function of the available network bitrate (T_{TCP}) and minimum Packet Size, $P_{S(min)}$. Once the PR_C is either equal to or greater than the 'fixed Packet Rate' $P_{R(fix)}$ of the application, the CCM signals to the application to start sending data. This is the point where the start-up phase ends and the Packet Size Truncation (PST) function of the CCM takes over. This maintains the fixed Packet Rate, and adapts the Packet Size in response to congestion. This process is illustrated in Figure 8.2.

$$P_R = \frac{T_{TCP}}{8 \cdot P_{S(min)}}$$ Equation 8.2

If $\mathbf{P_R \geq P_{R(fix)}}$ ***, signal to application to start sending data***

The Start-up Phase function conveys useful information, in advance, to both the end-systems, about the congestion state of the network (For example, whether it is suitable to make and maintain a connection between the two users or not). If the CCM does not indicate the minimum bitrate within a pre-determined time period, the CCM terminates the connection and notifies the application accordingly.

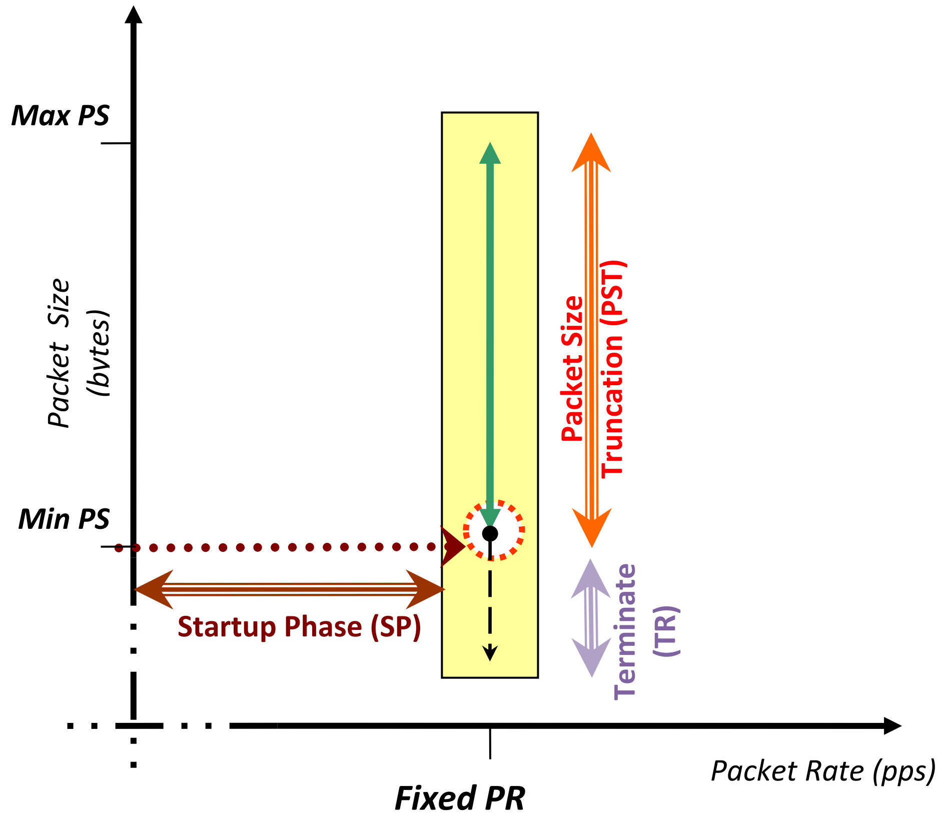
In terms of network performance, the behaviour of the Start-up Phase; maintaining a fixed minimum Packet Size and probing for the required Packet Rate, PR_{fix}, is a fair method of response to networks that are processing packets, irrespective of packet size. This improves the performance of those networks (network routers specifically) which take into account the rate at which packets arrive into the network compared to the size of the packets

8.3.2.2. Terminate

A connection will be forced to terminate immediately if (*1*) during the Start-up phase the CCM cannot indicate a Packet Rate higher than or equal to, the fixed Packet Rate of the application within a pre-determined time period, (*2*) during the connection (i.e. after the Start-up Phase) the Packet Size Truncation (PST) function computes a Packet Size lower than the minimum size as required by the encoder.

The termination of the connection would be immediate, because the calculated available network bitrate, T_{TCP}, is a result of the bitrate reducing at a smooth rate, which is achieved by the 'loss event' rate calculation. Therefore, the indicated bitrate, T_{TCP}, is not expected to increase any time soon.

The Start-up phase time period can be around 10 seconds as this is what is seen with GSM networks, which is referred to as call-setup time.



Probe for desired Packet Rate. When this condition is fulfilled signal to Multimedia Application to start sending data

Minimum Bitrate (BR_{min}) for Application.
Signal to Application to start sending Multimedia Data

Vary Packet Size in Response to Congestion

If the PST function of the TFGS CCM requests an adaptation of the packet smaller than the minimum Packet Size, then terminate the connection and notify the multimedia application.

Figure 8.2, 'TFGS CCM Configuration including SP, TR and PST functions'

9. Appendix I, Results: Homogeneous Traffic Mix

This section shows the simulation results for a homogenous traffic mix, giving a comparison of TFRC and TFGS CCM across various congestion levels. The network scenario description is defined in Table 5.3.

Simulation Study 1 - Homogenous Traffic Scenario							
Simulation Set	**Flow (Application) Type**	**Codec Type**	**Duration (secs)**	**CCM**	**No. of Flows**	**Capacity, C, (bps)**	**RED Queue Parameters (Min / Max) in packets**
A	Voice	MPEG-2	60	TFRC	X	499,200	20 / 60
B	Voice	MPEG-2	60	TFGS	X	499,200	20 / 60

from Table 5.3 *(Chapter 5)*

woTwoSS -28 with Validation, Non-cong.xlsx

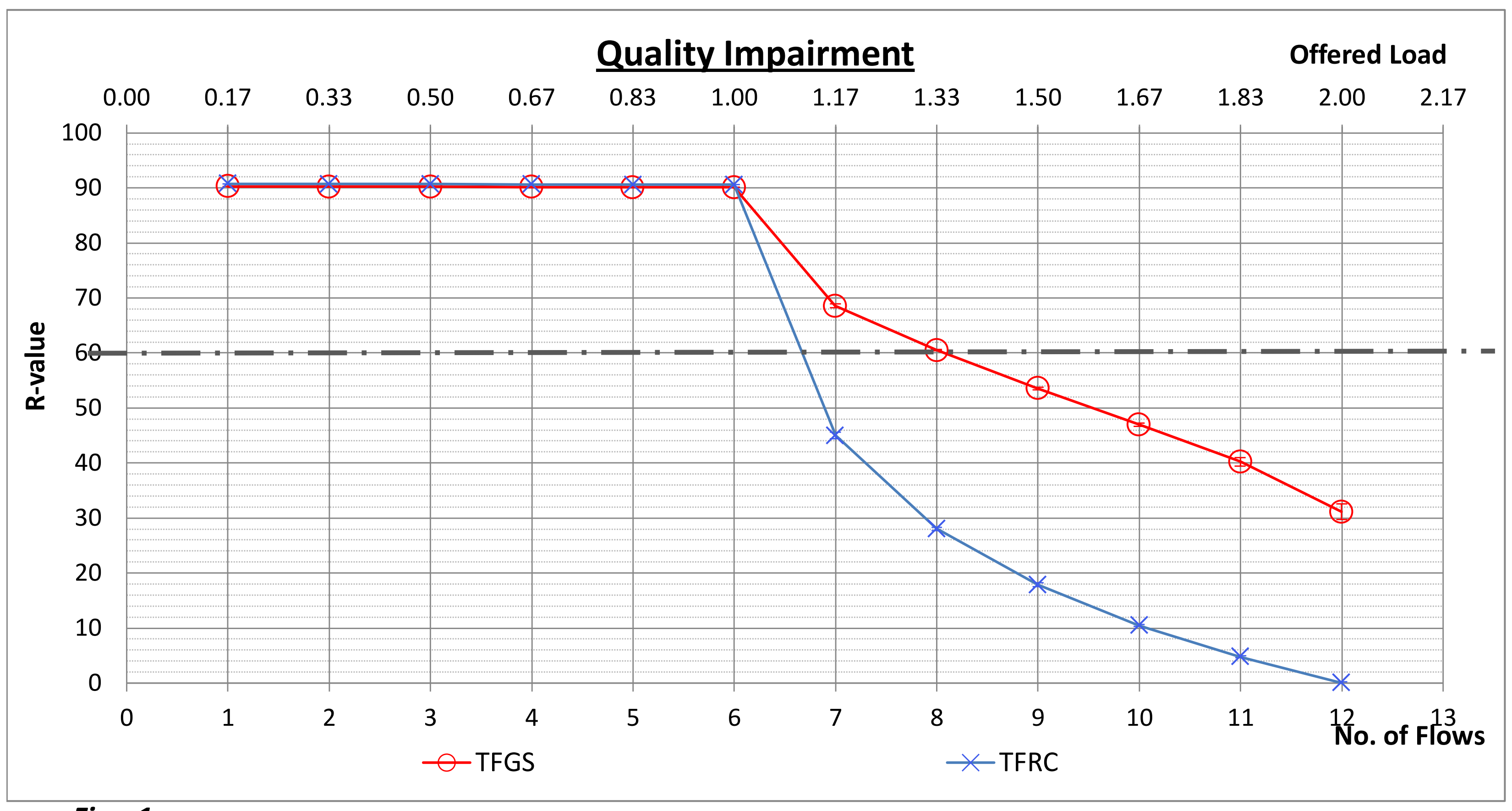


Fig. 1

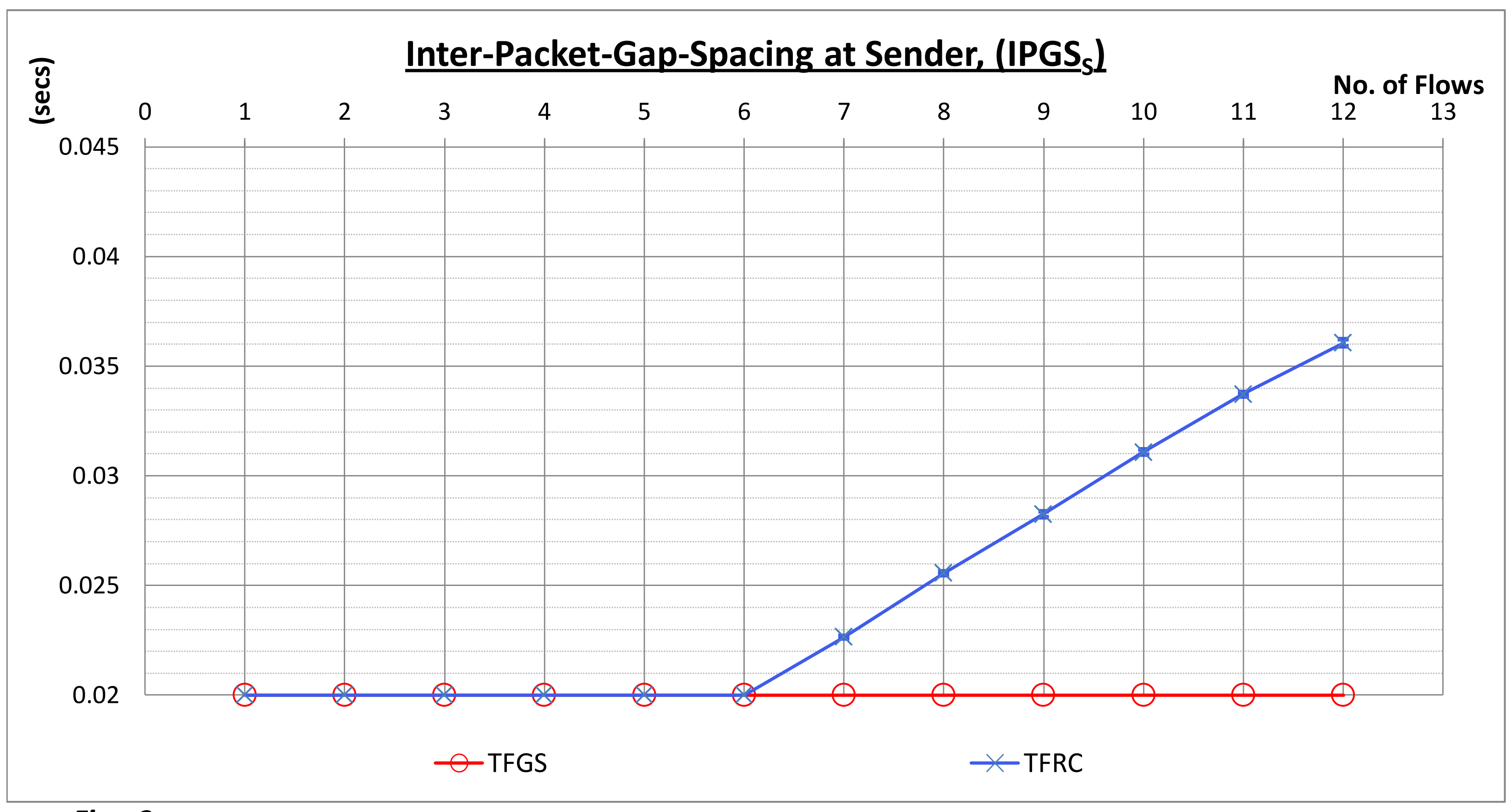


Fig. 2

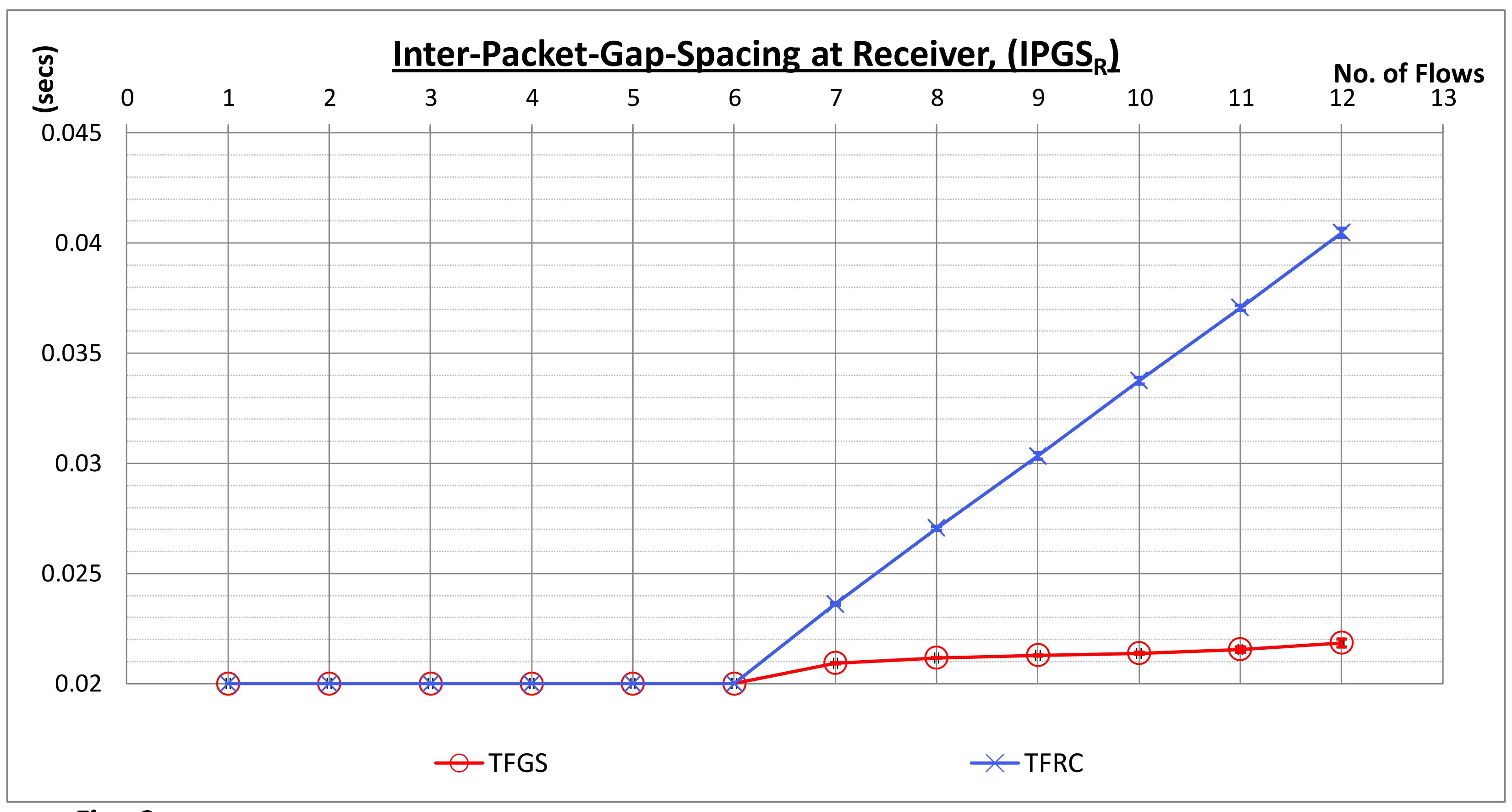


Fig. 3

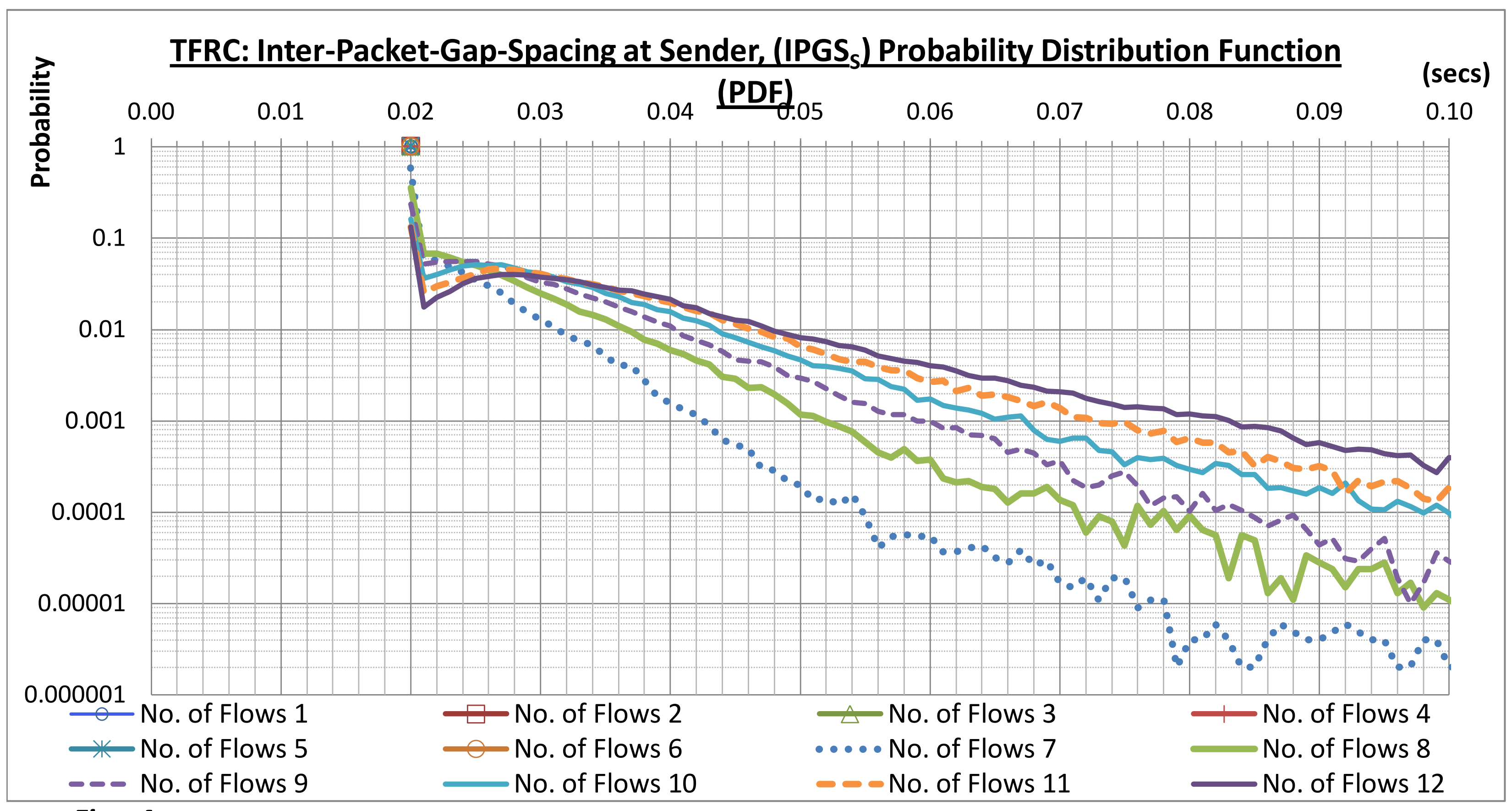


Fig. 4

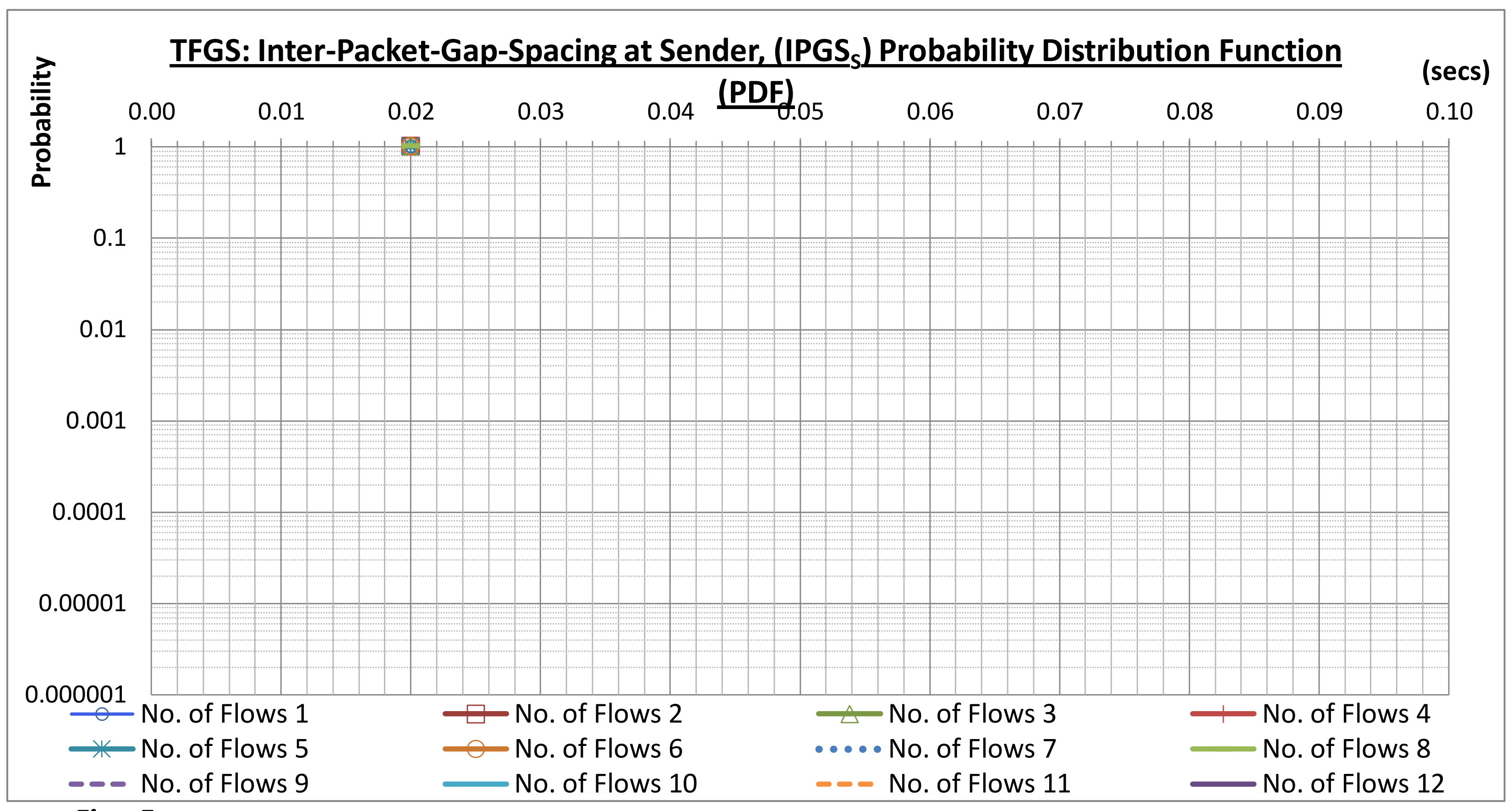


Fig. 5

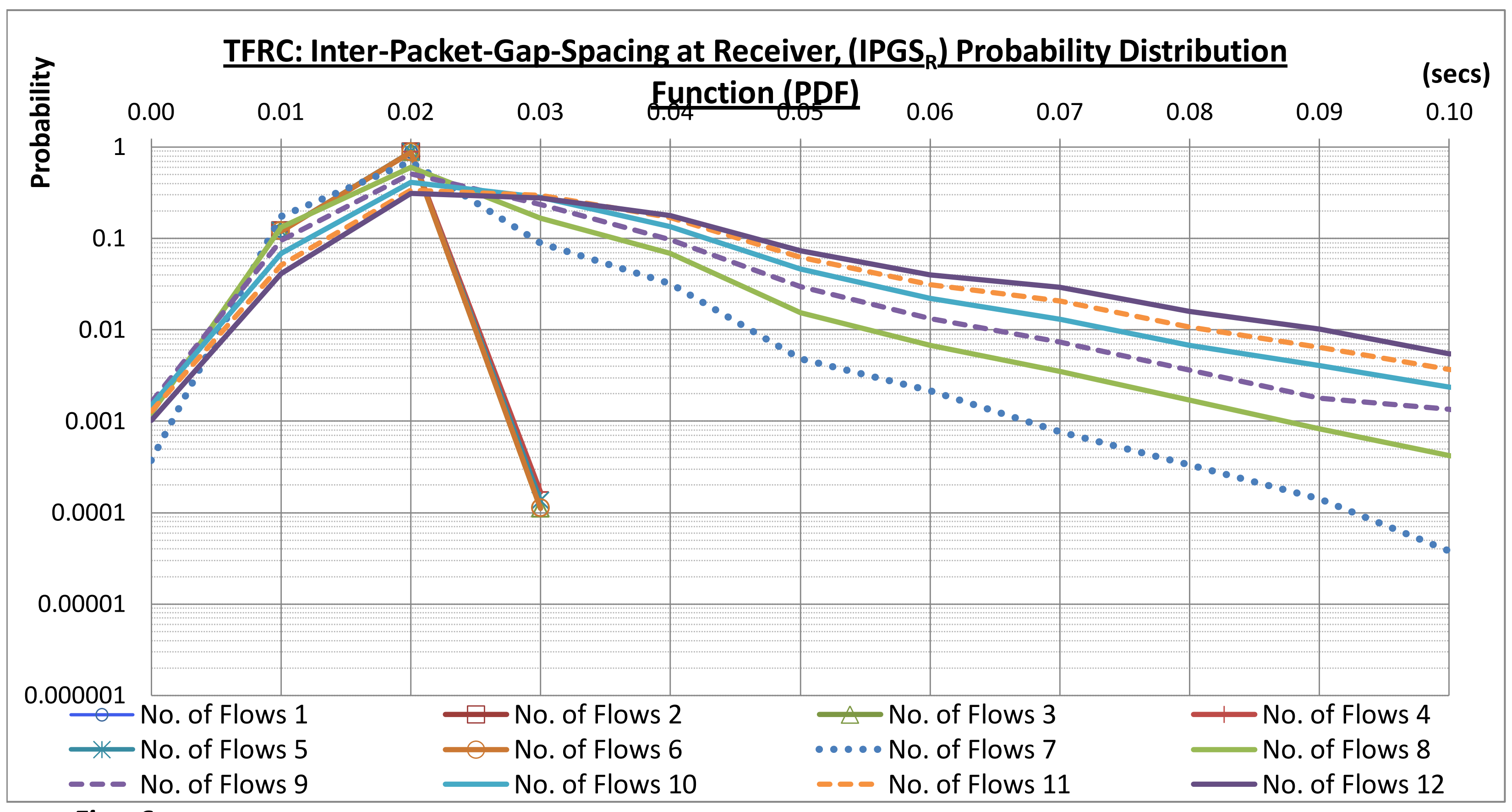


Fig. 6

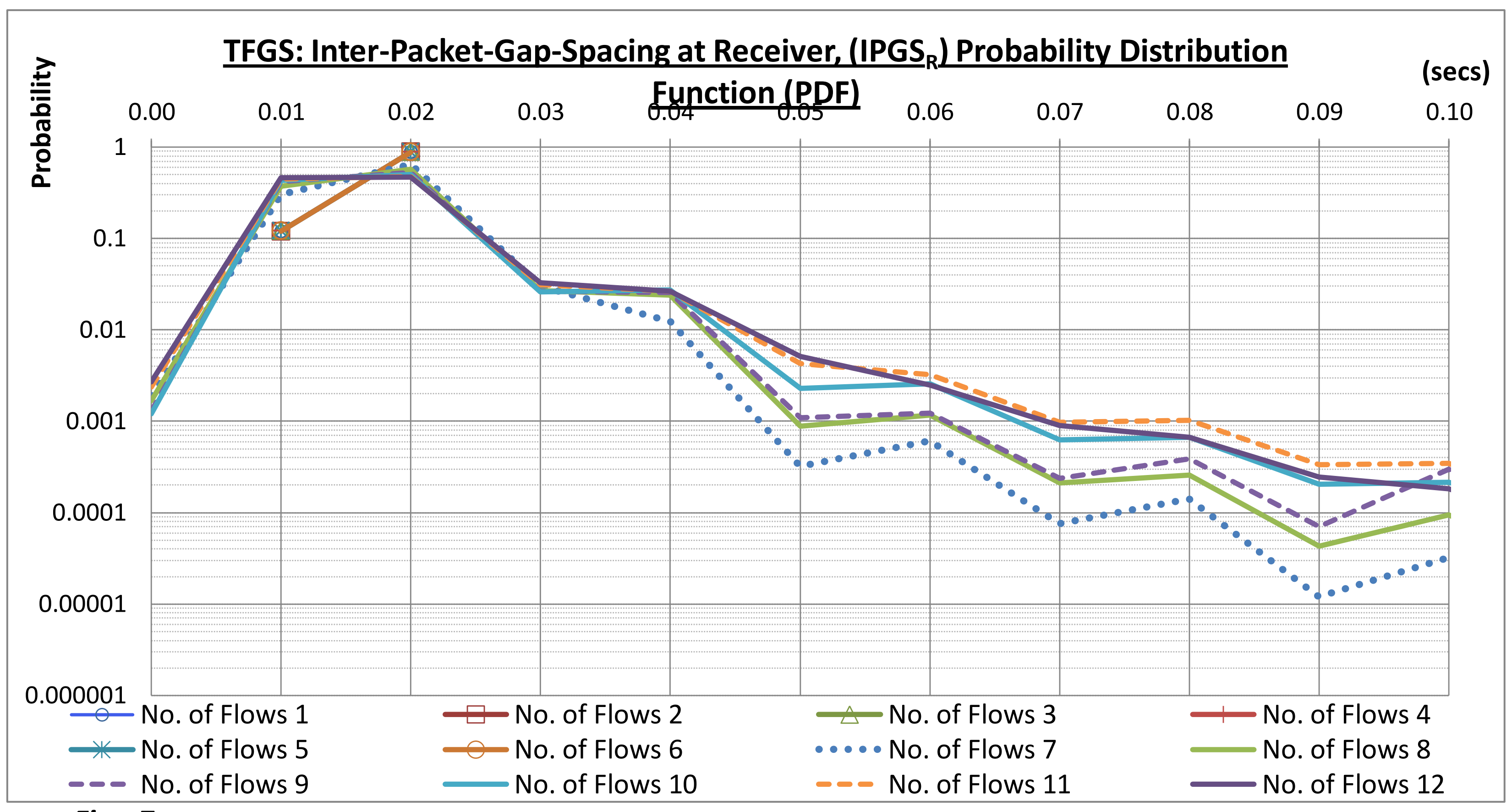


Fig. 7

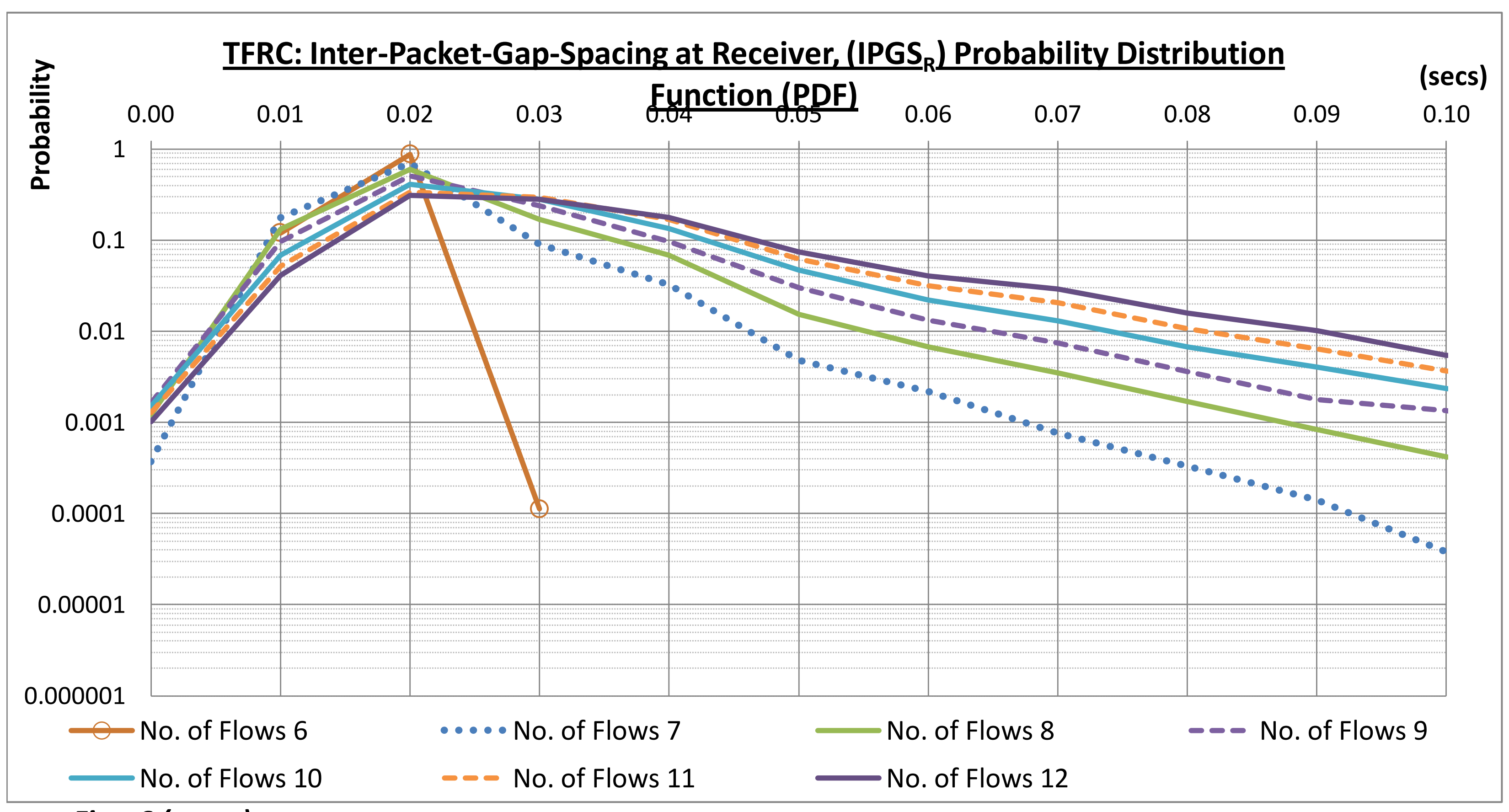


Fig. 6 (zoom)

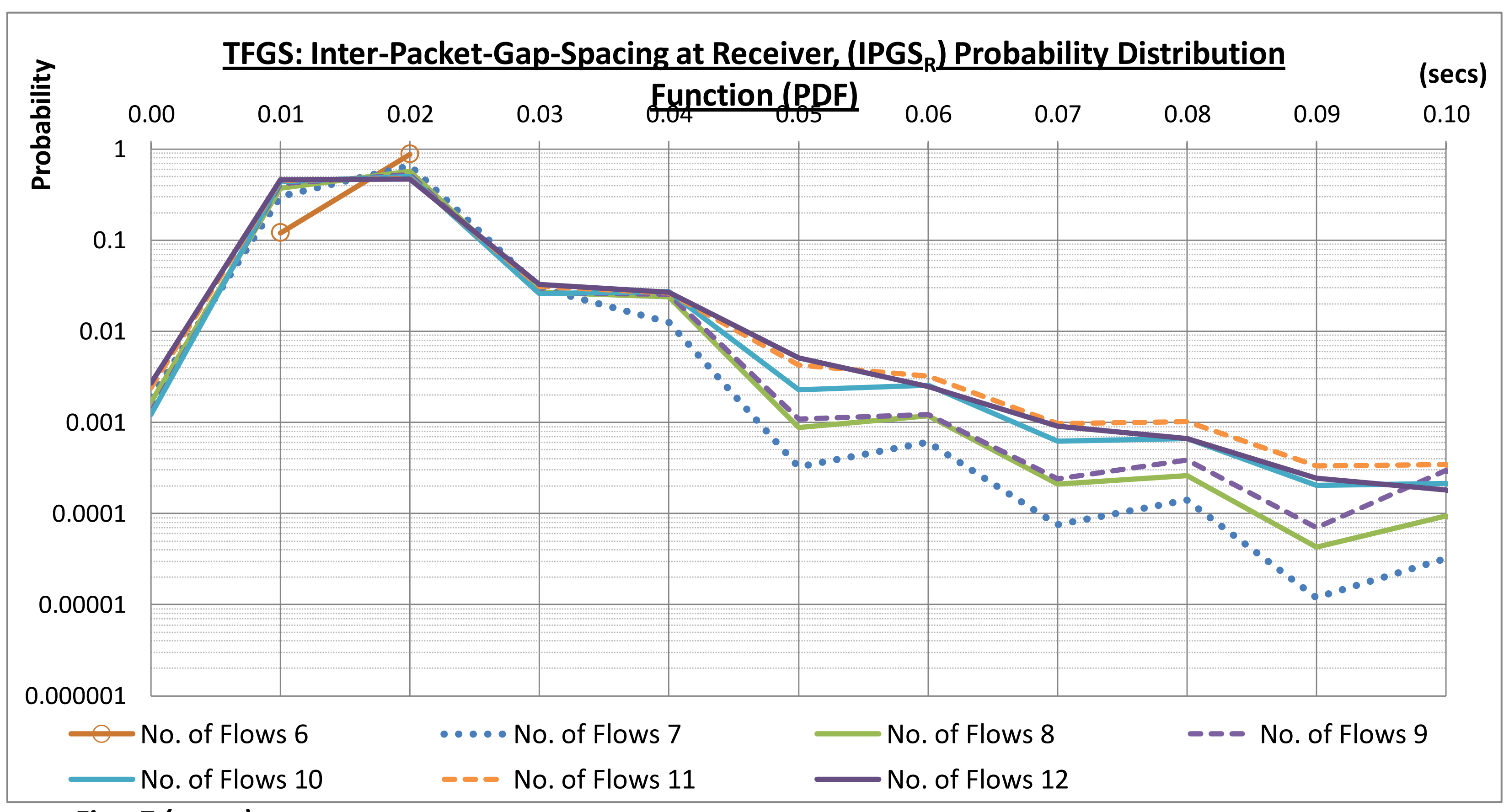


Fig. 7 (zoom)

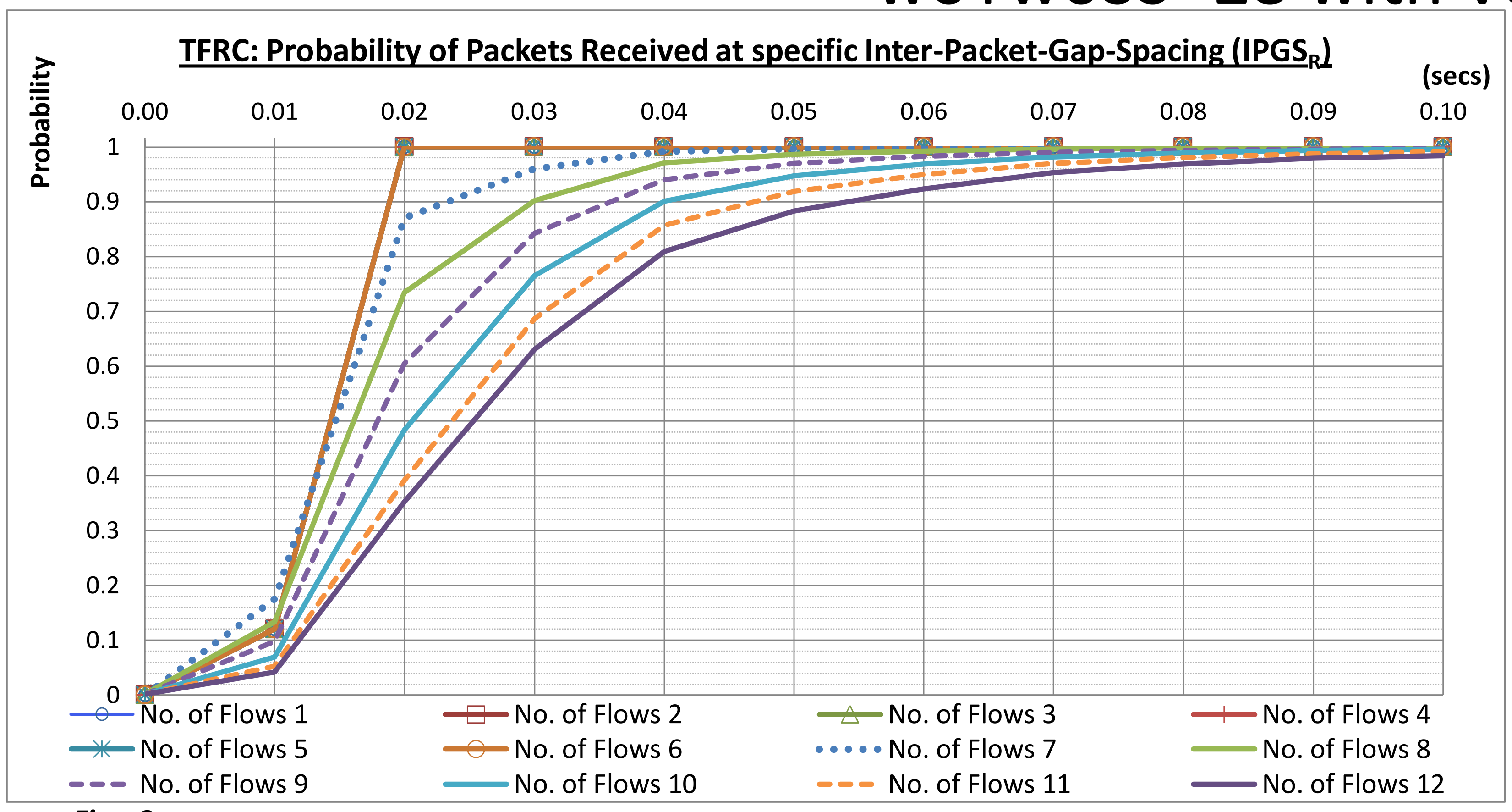


Fig. 8

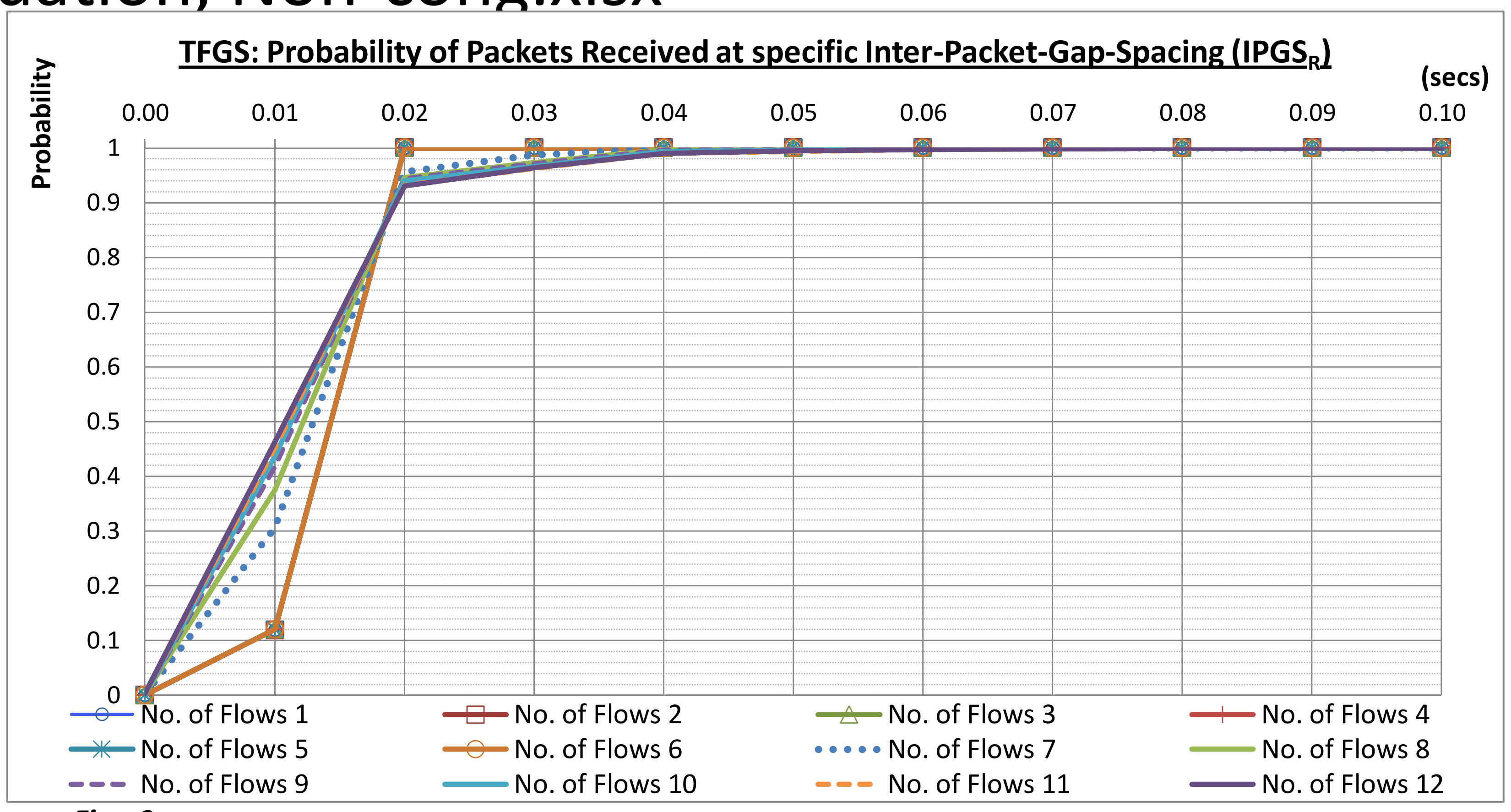


Fig. 9

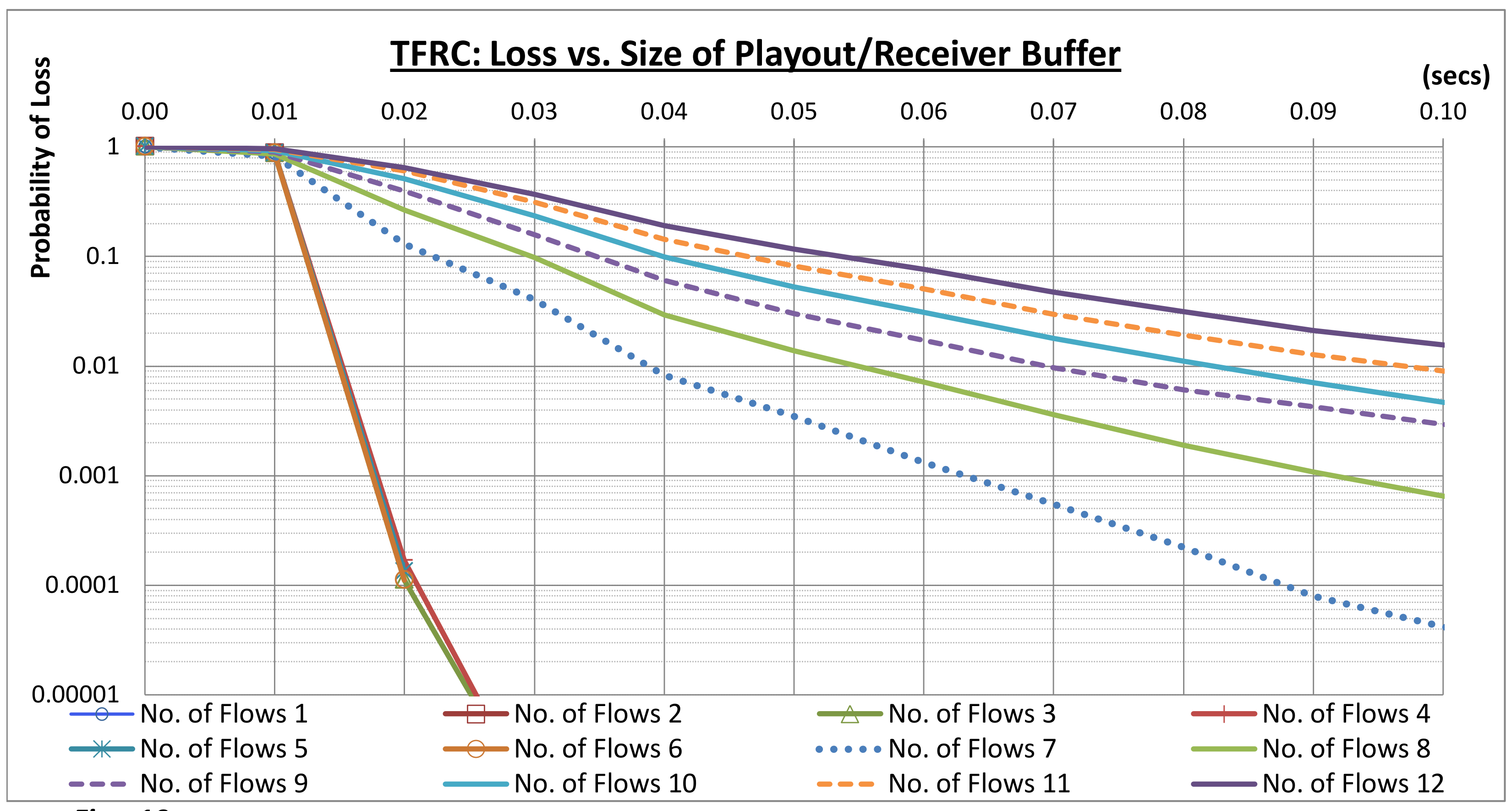


Fig. 10

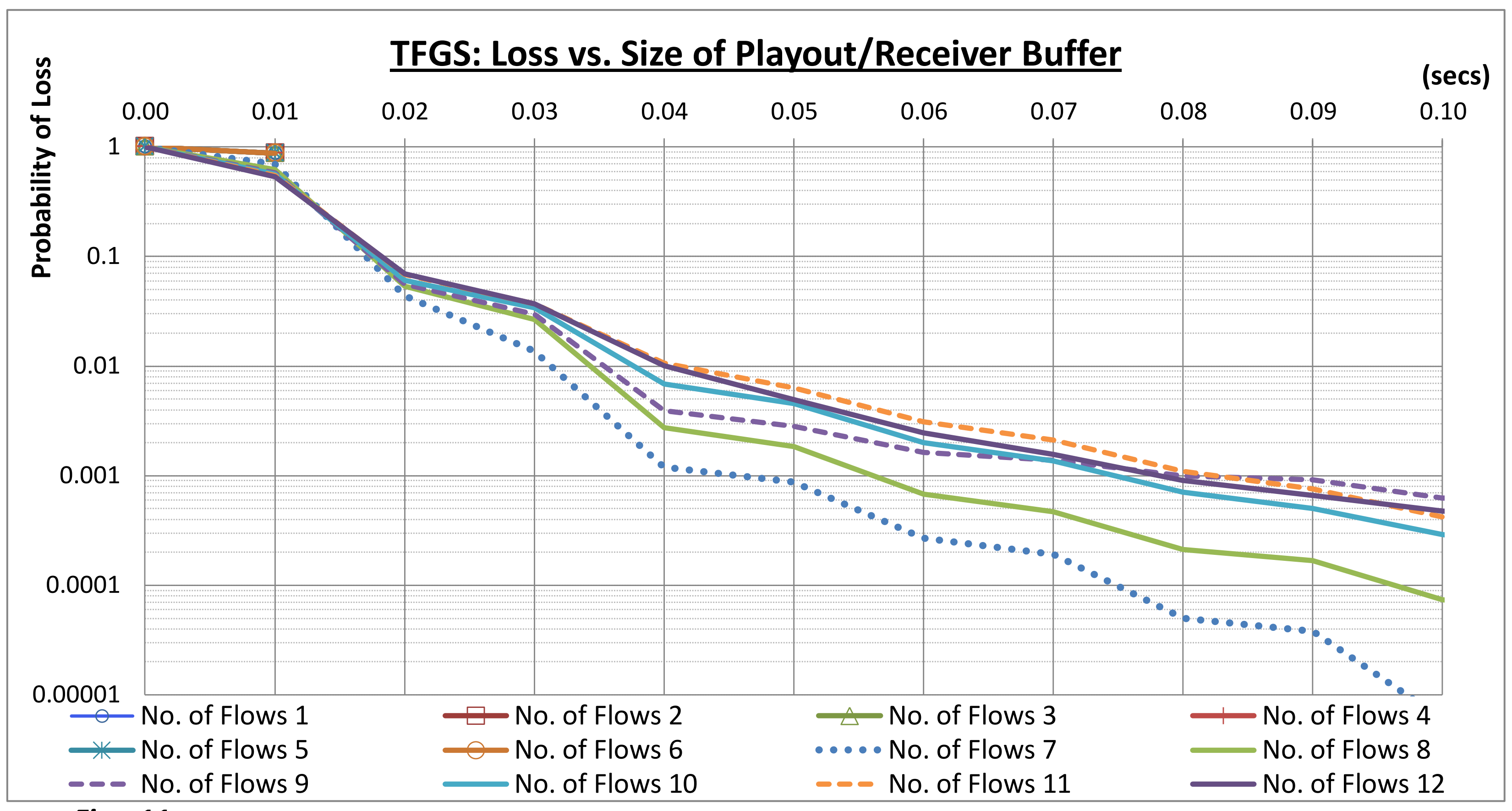


Fig. 11

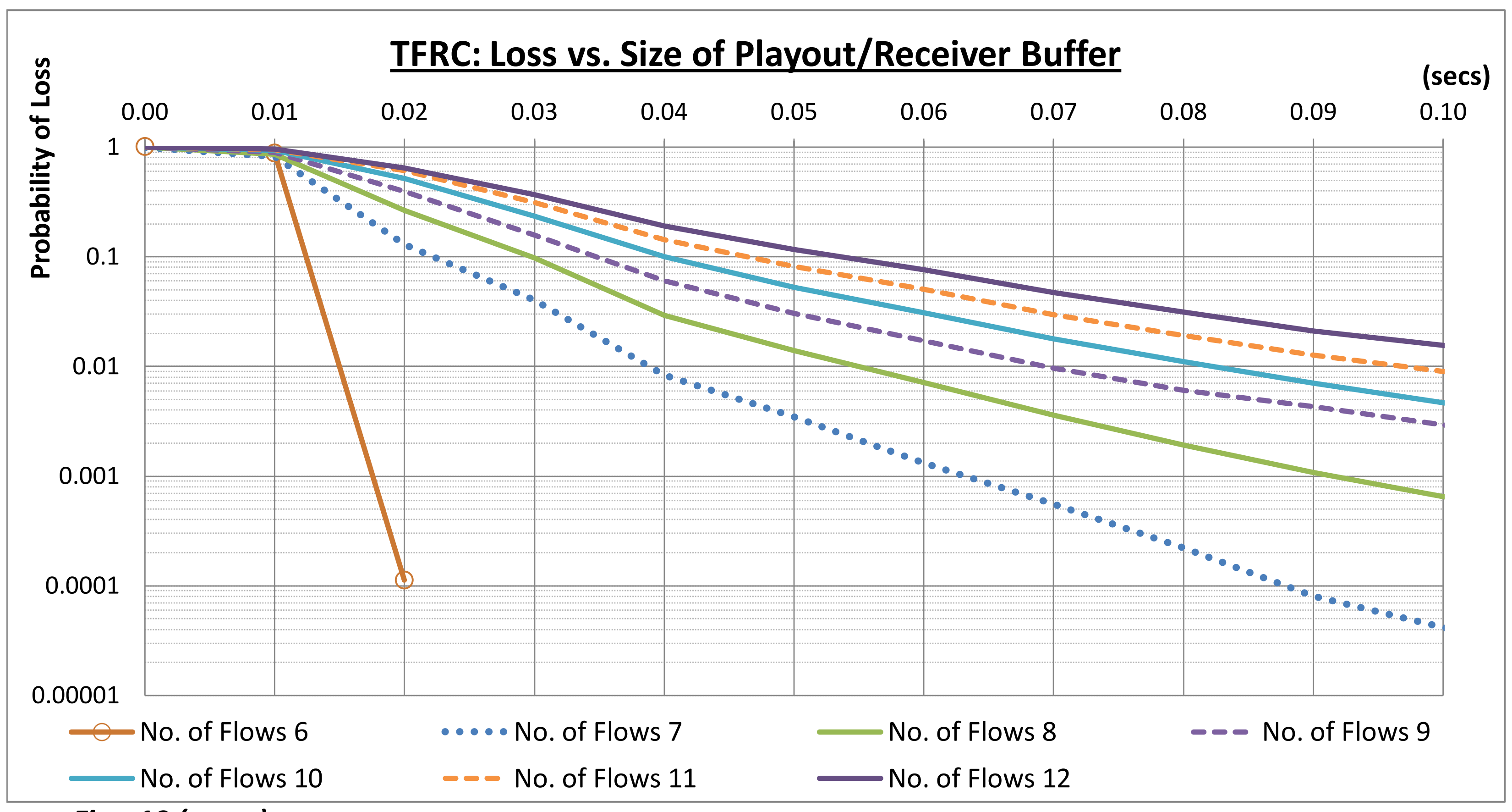


Fig. 10 *(zoom)*

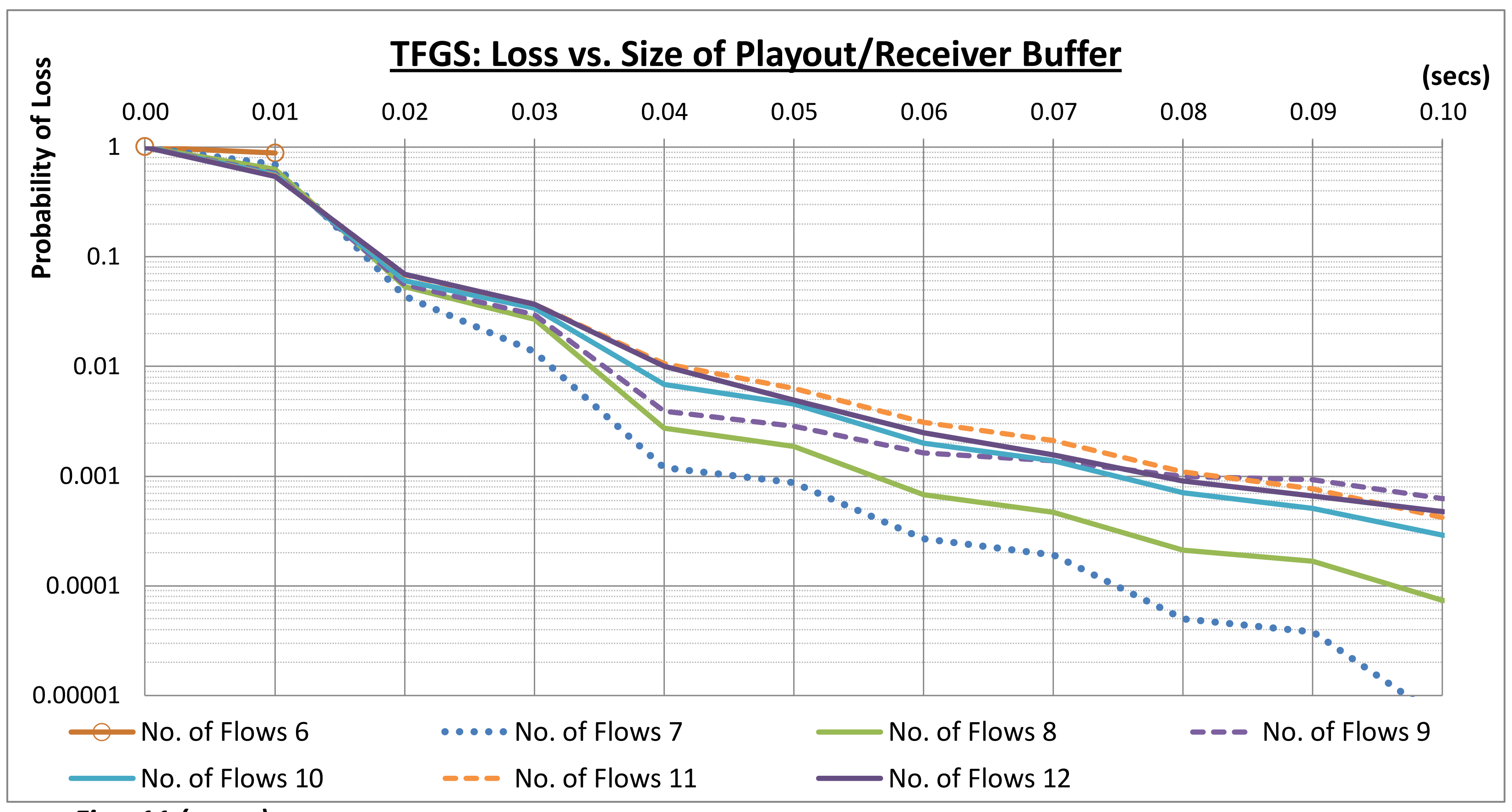


Fig. 11 *(zoom)*

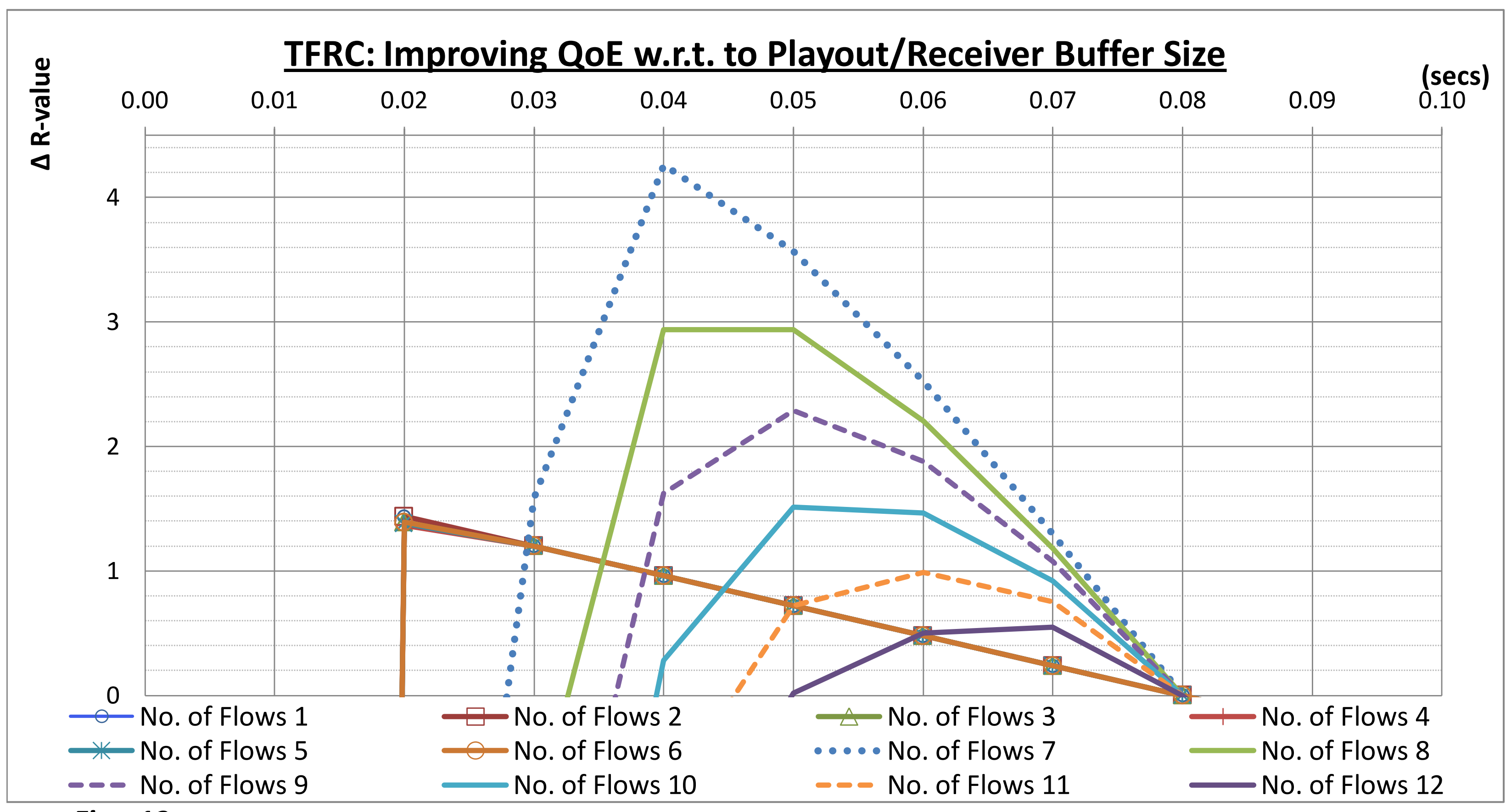


Fig. 12

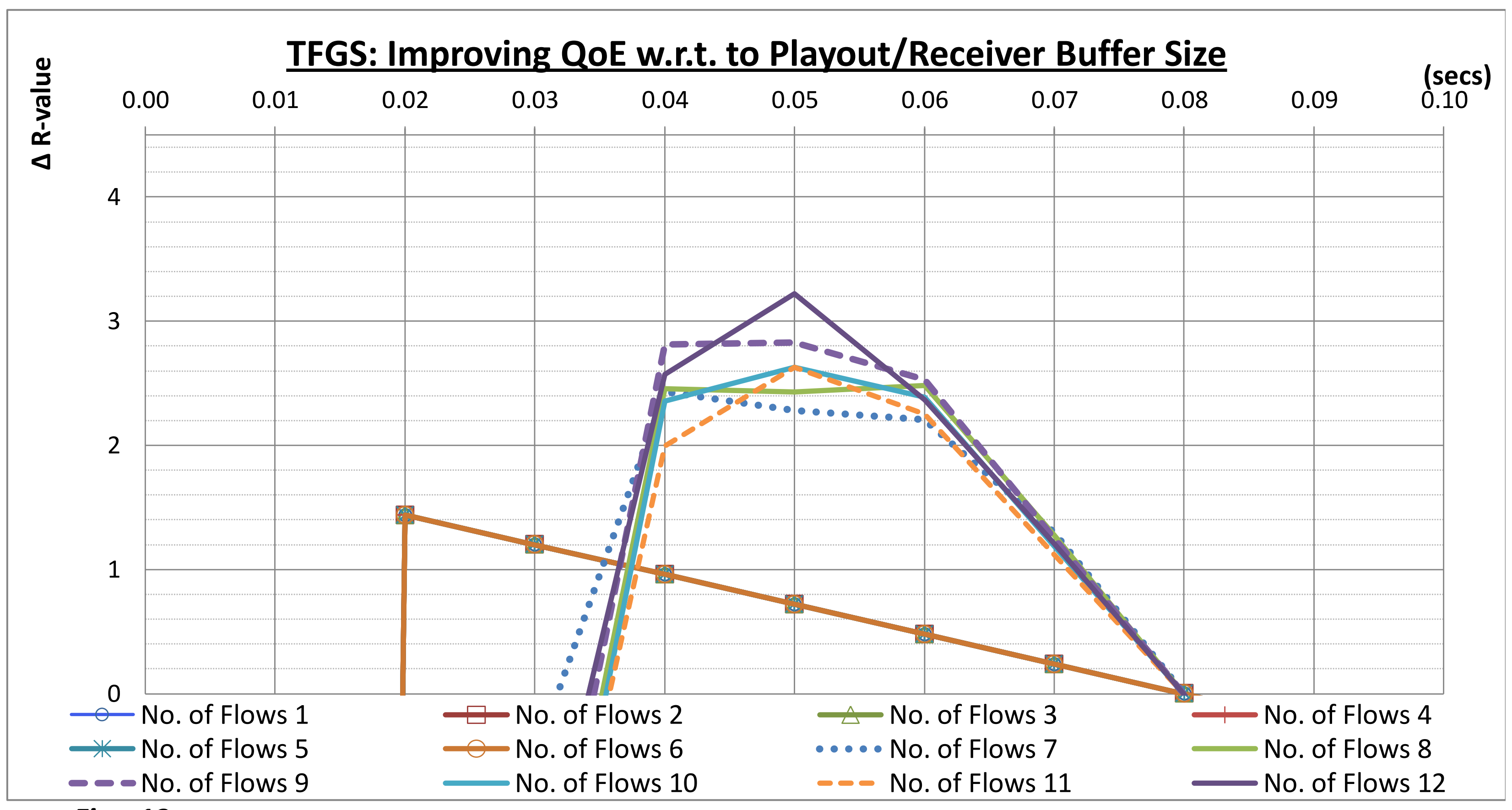


Fig. 13

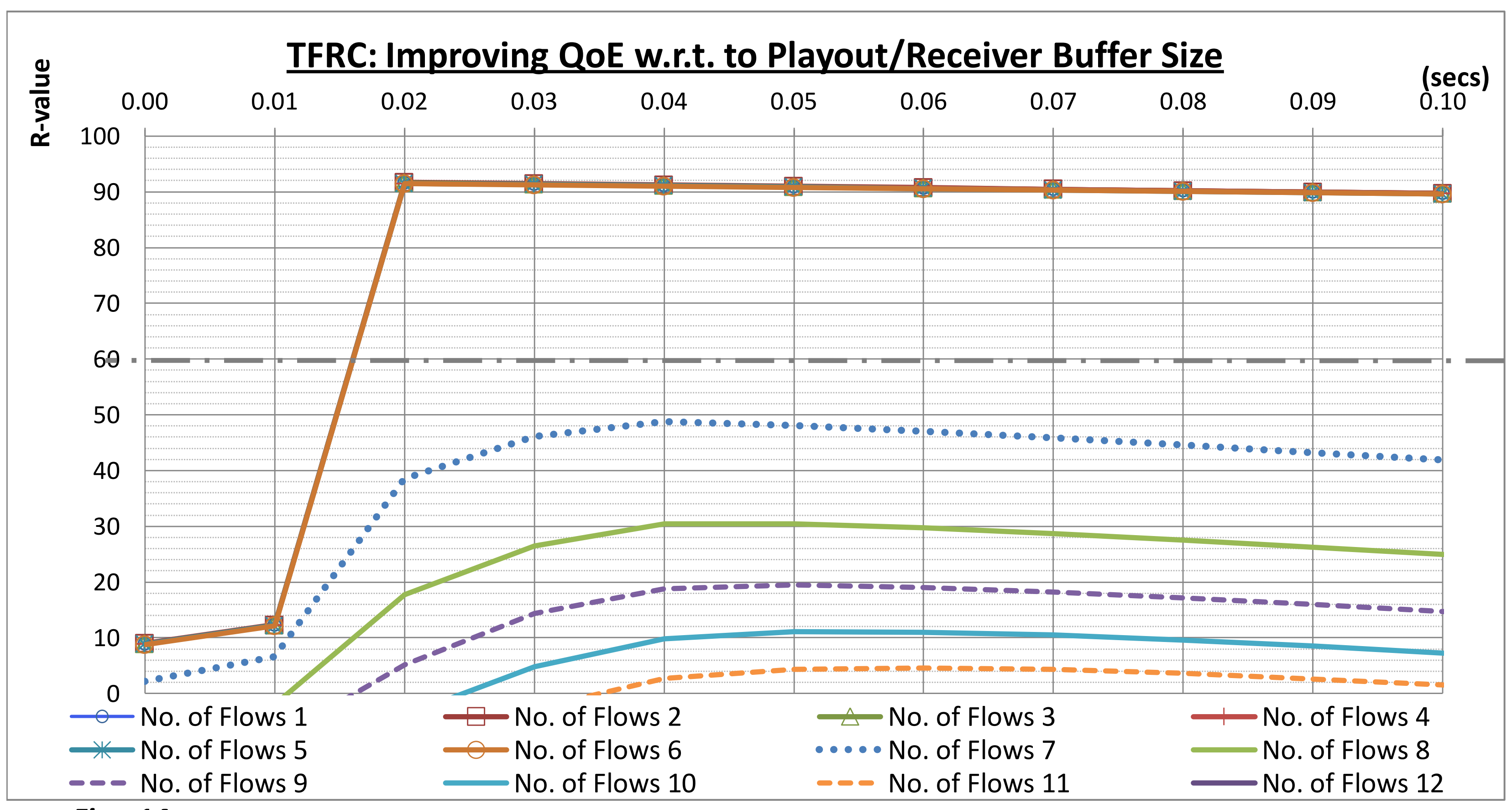


Fig. 14

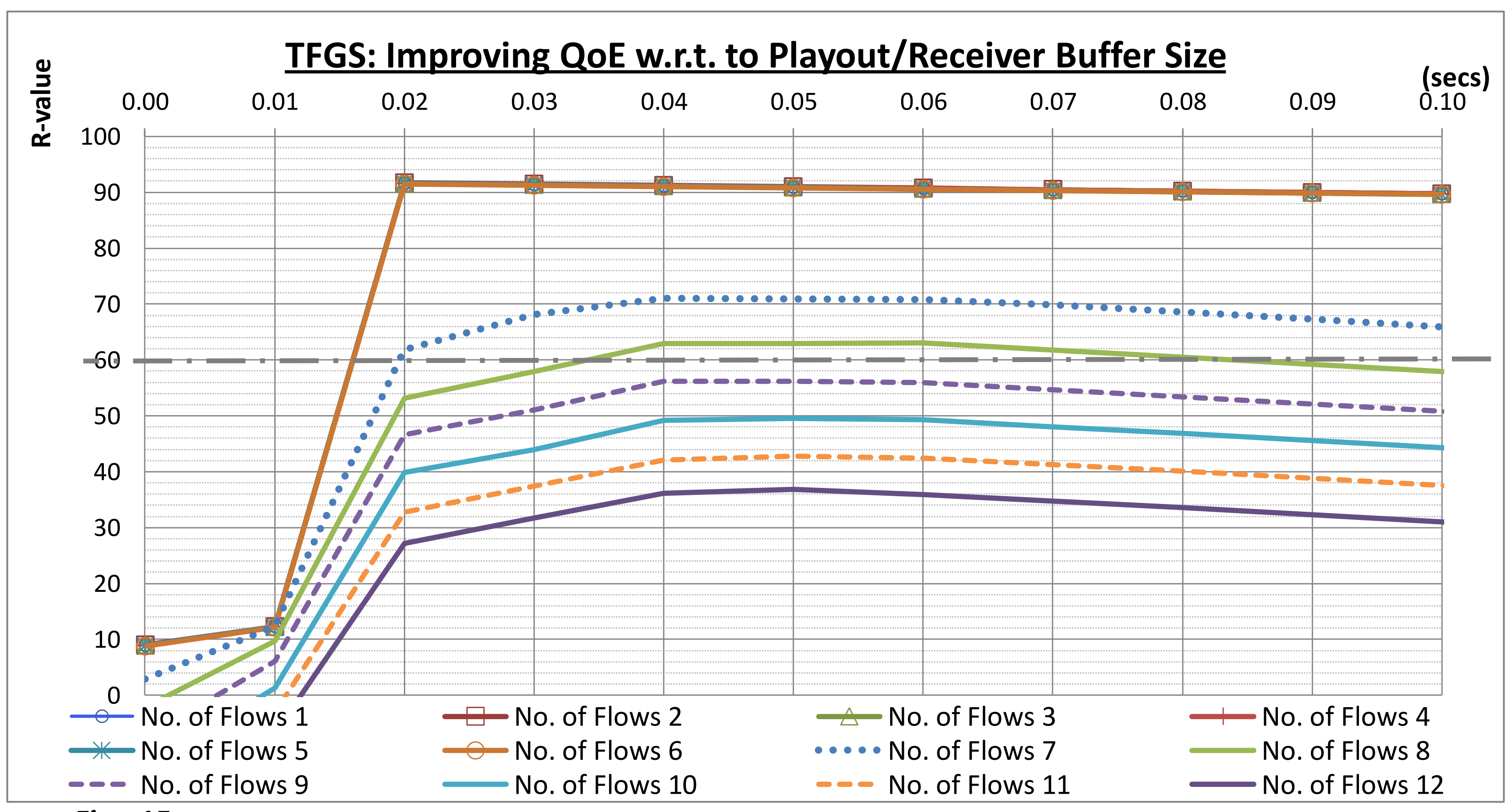


Fig. 15

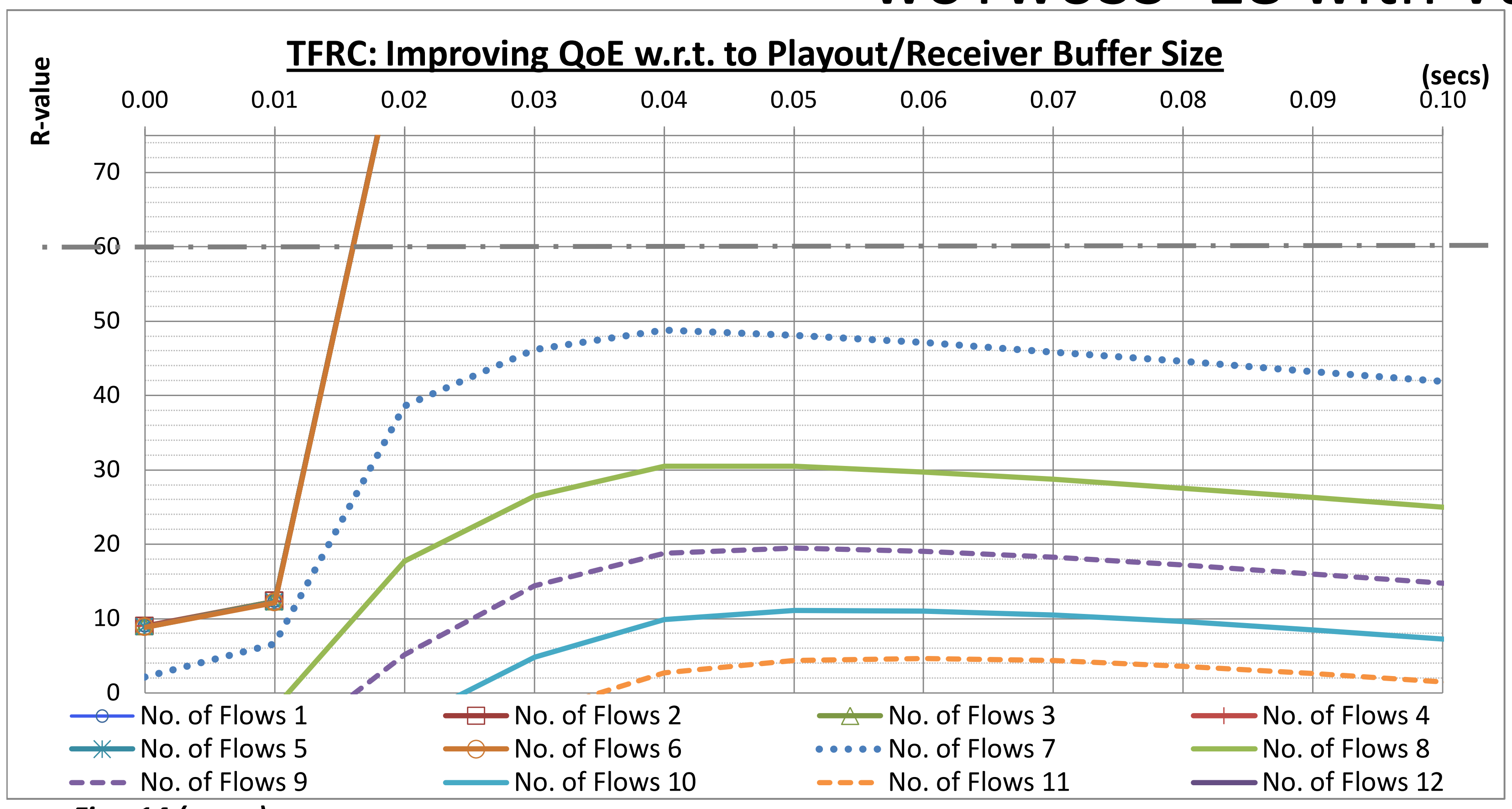


***Fig.* 14 (zoom)**

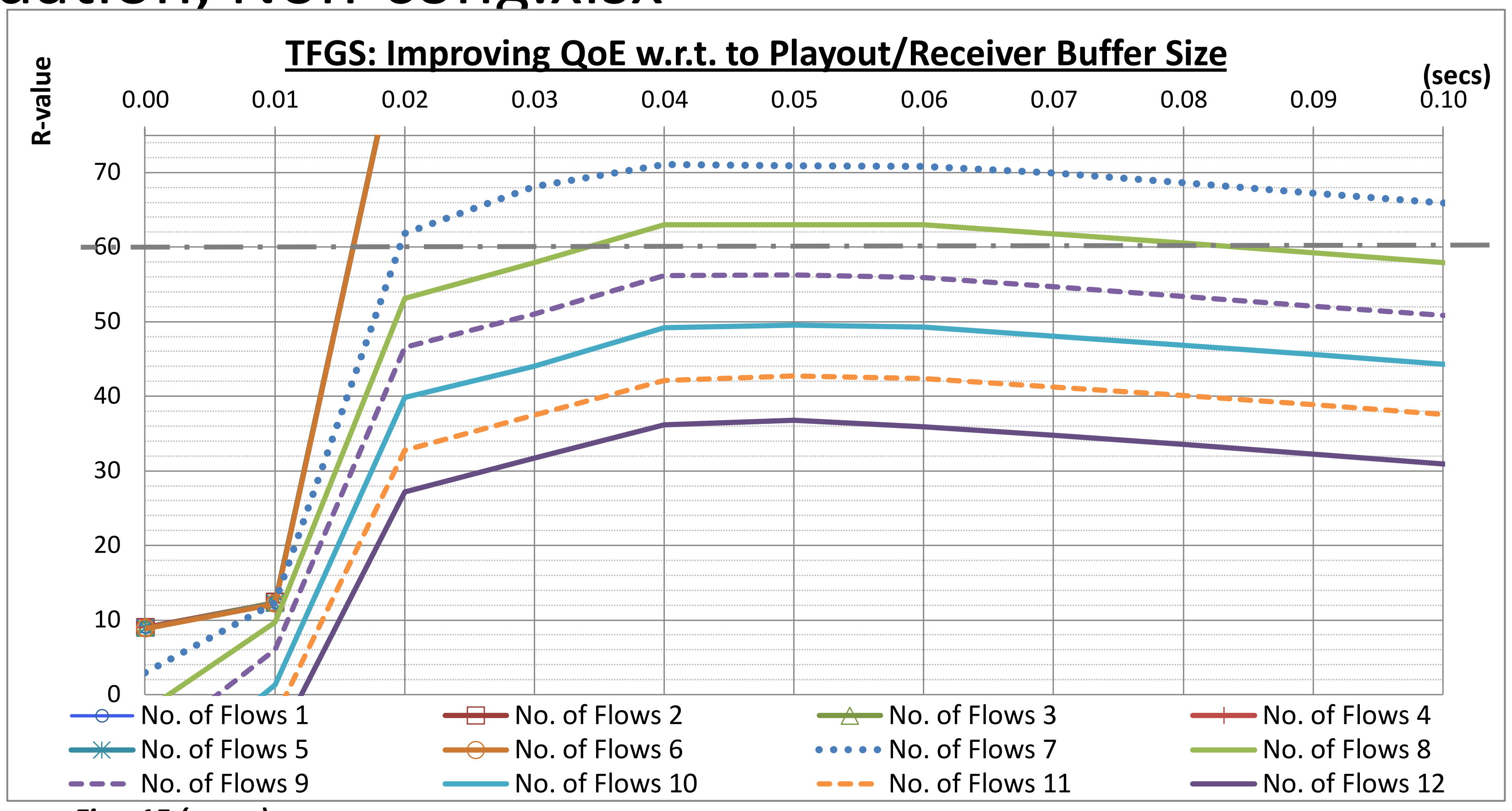


***Fig.* 15 (zoom)**

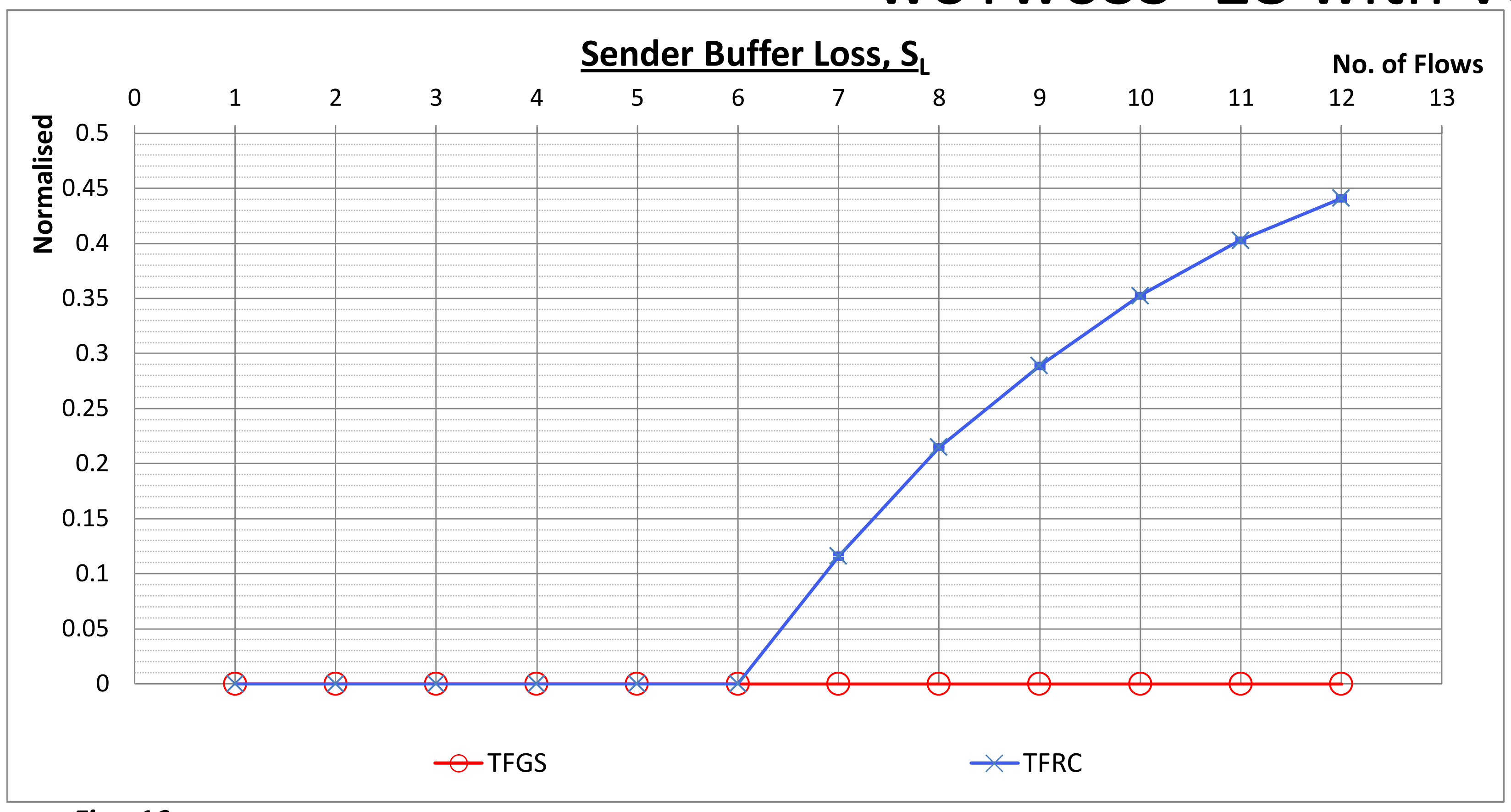


Fig. 16

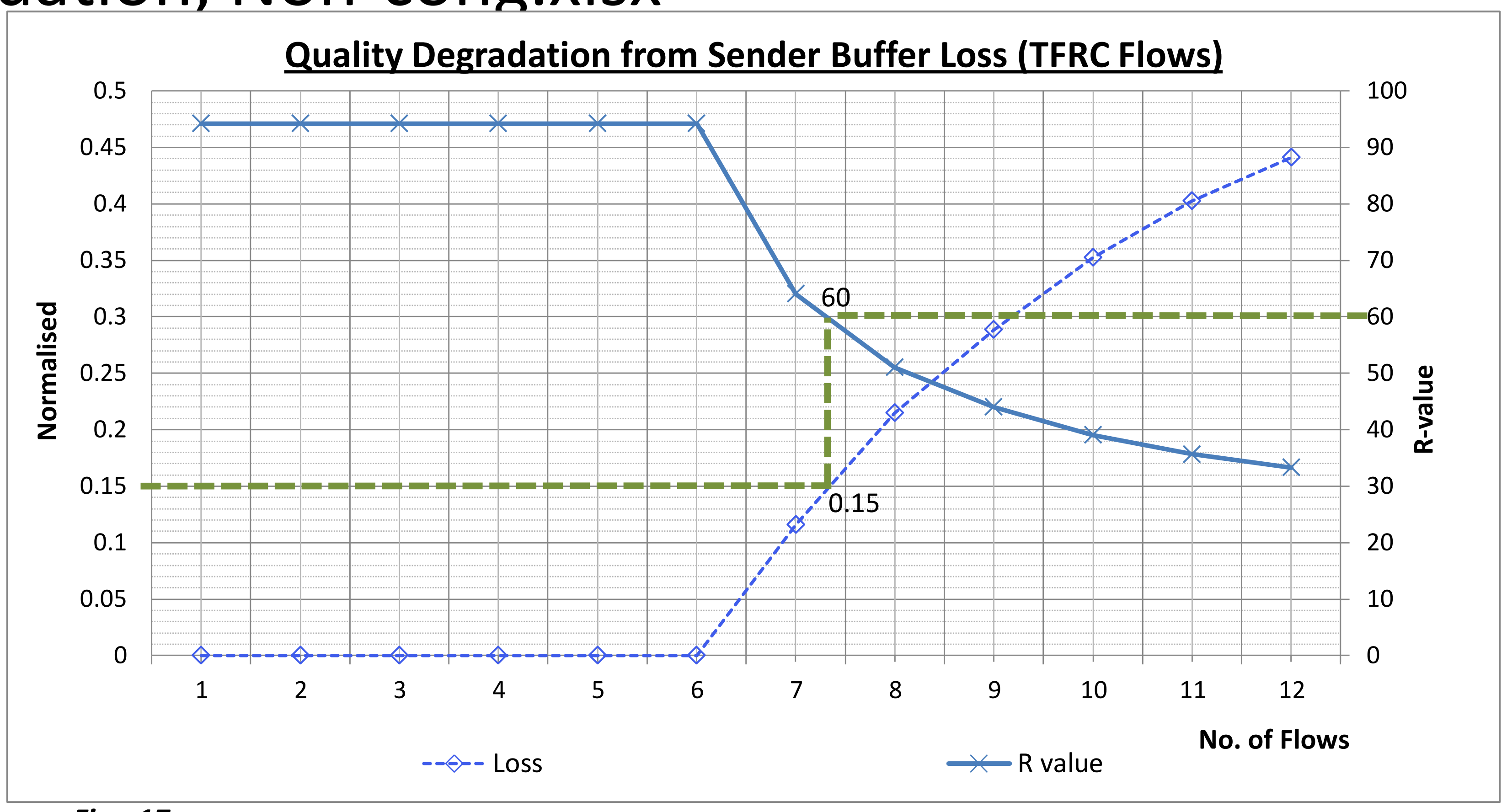


Fig. 17

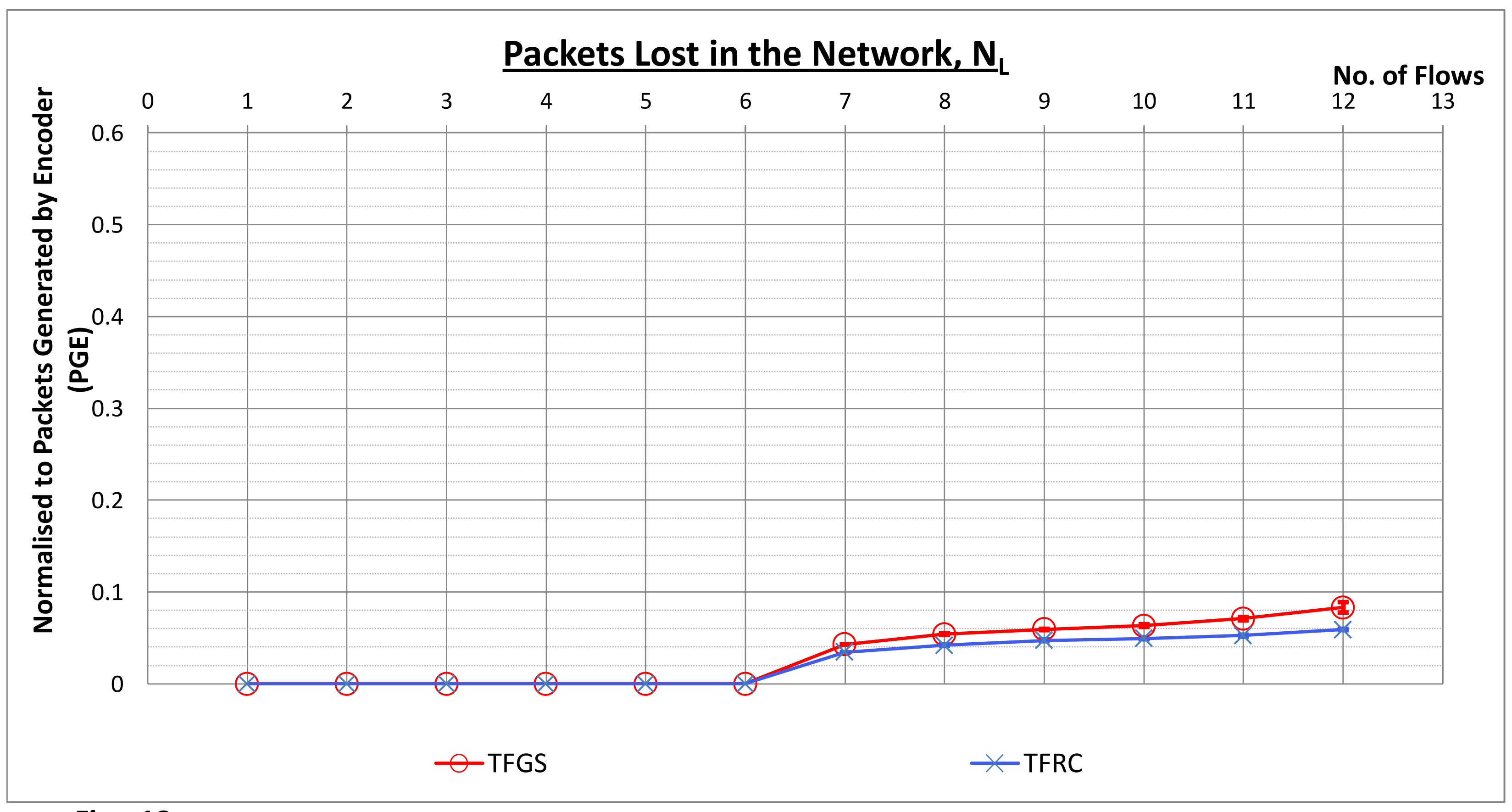


Fig. 18

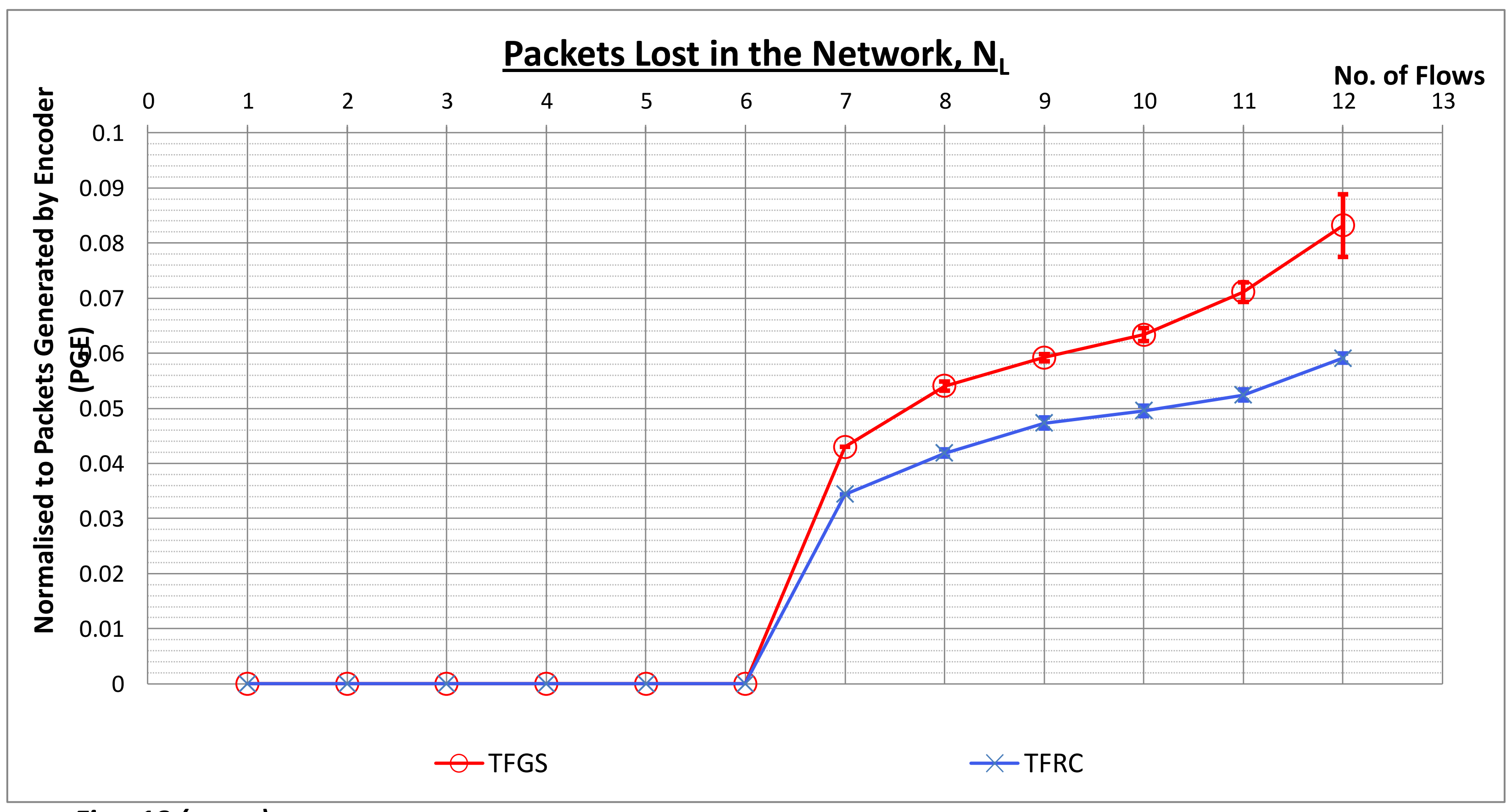


Fig. 18 (zoom)

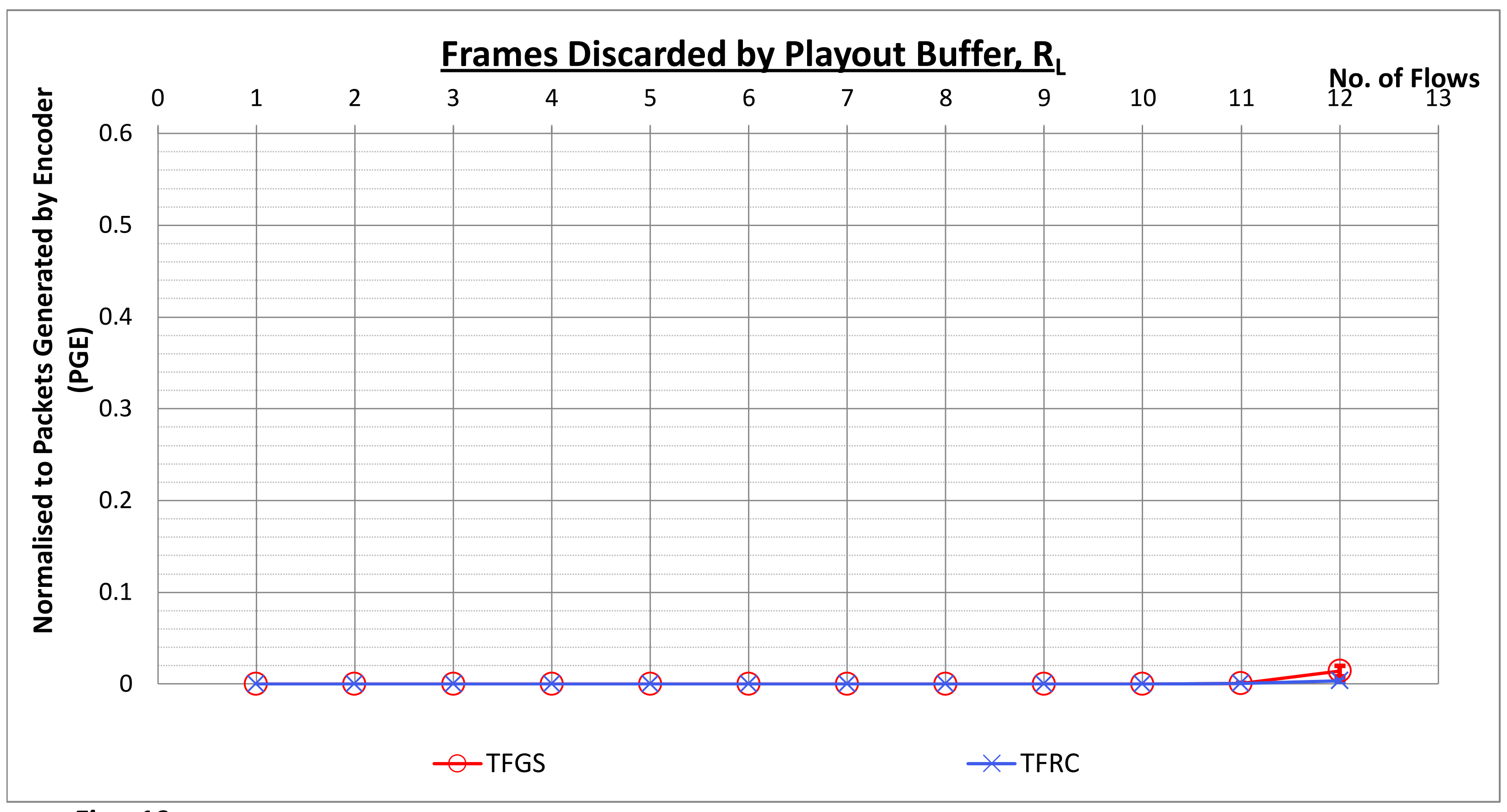


Fig. 19

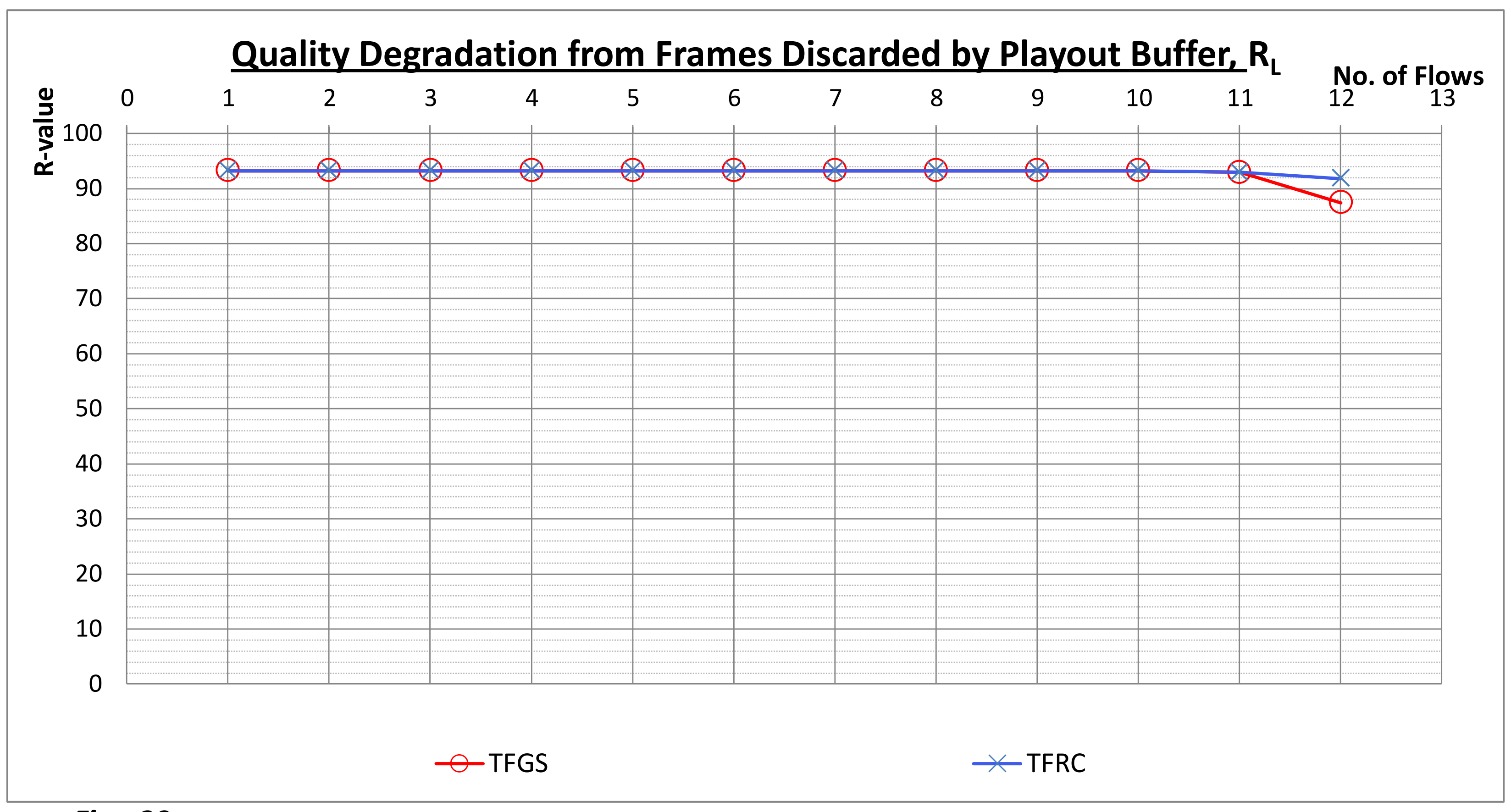


Fig. 20

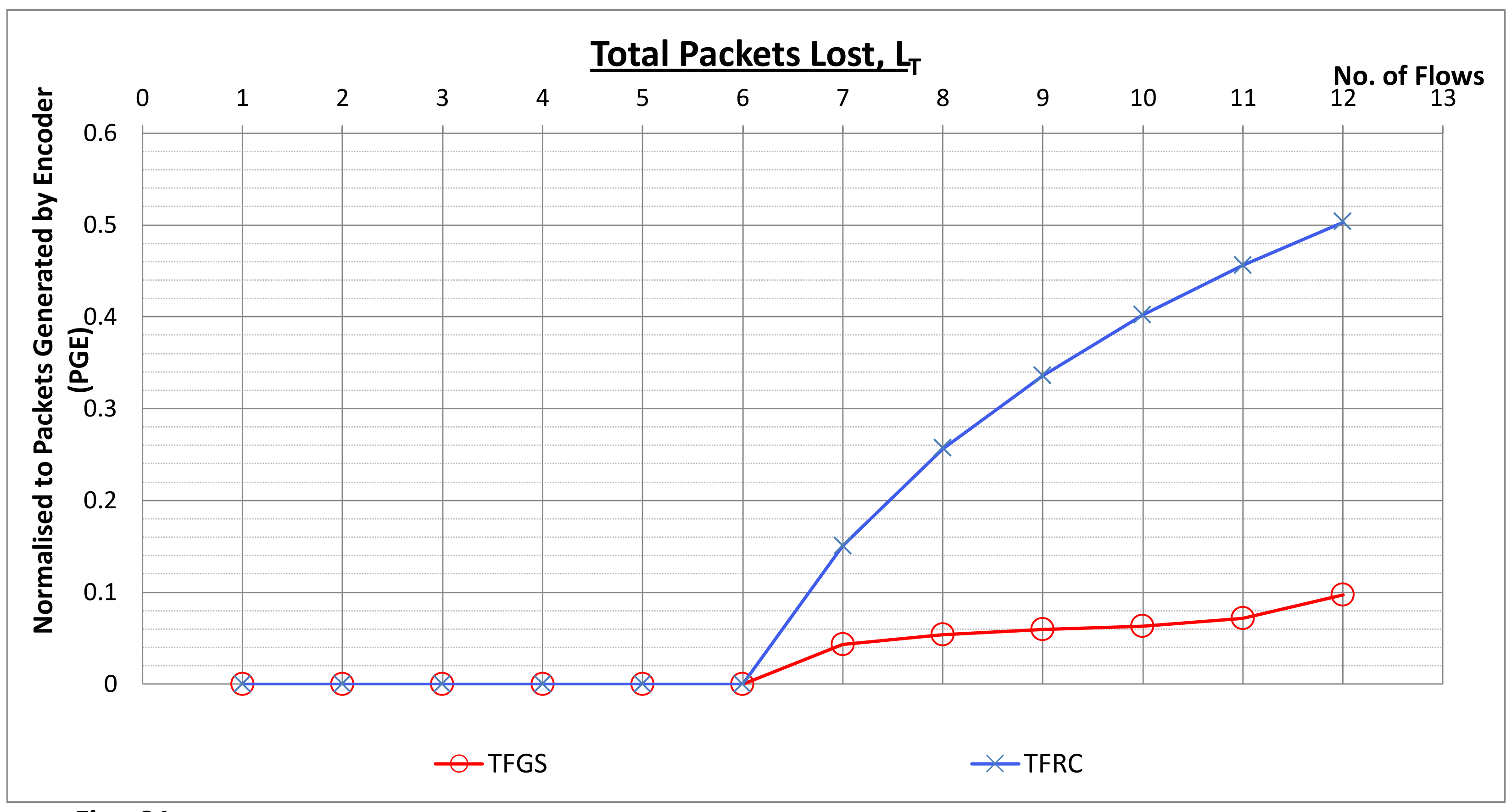


Fig. 21

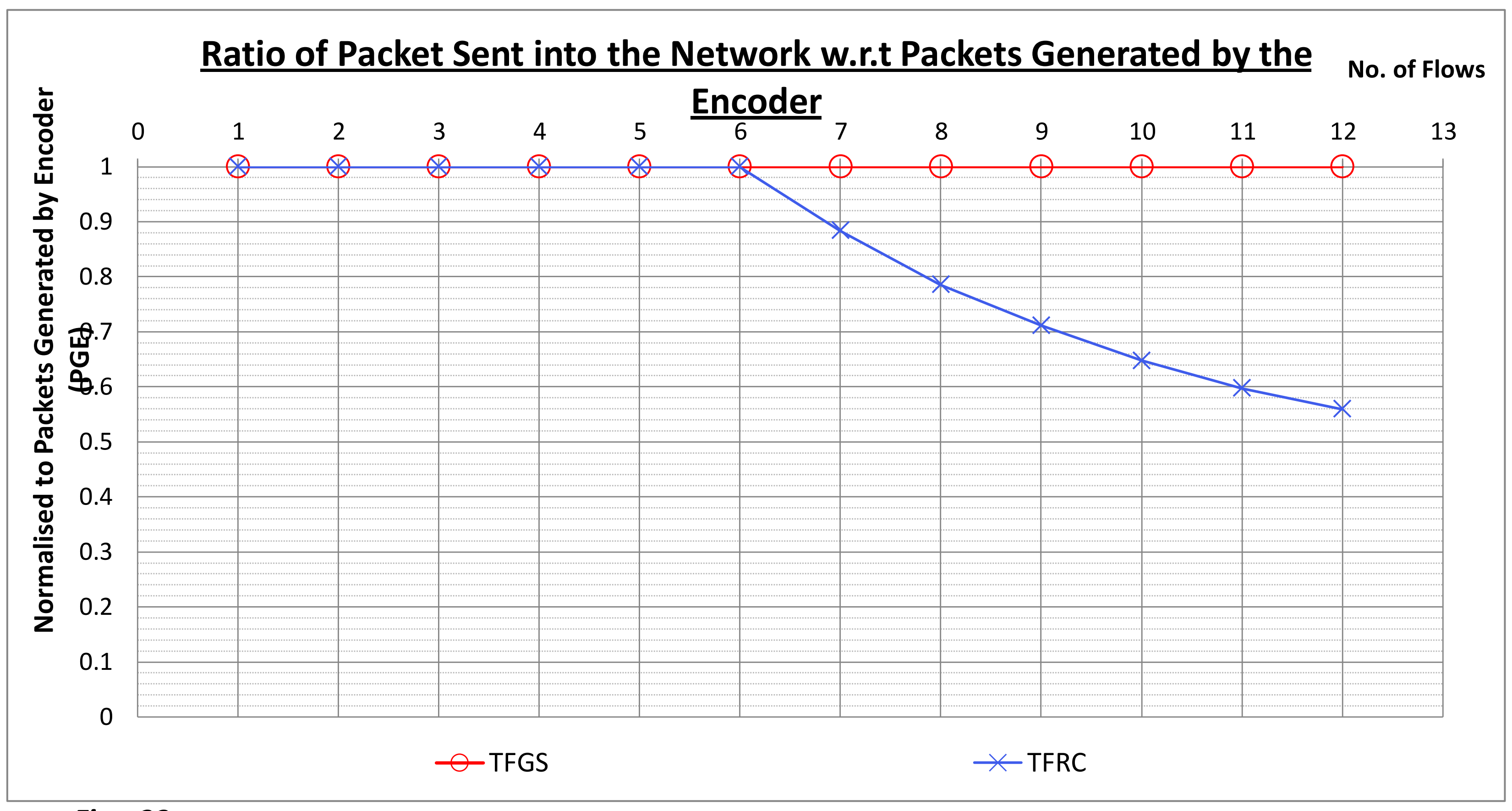


Fig. 22

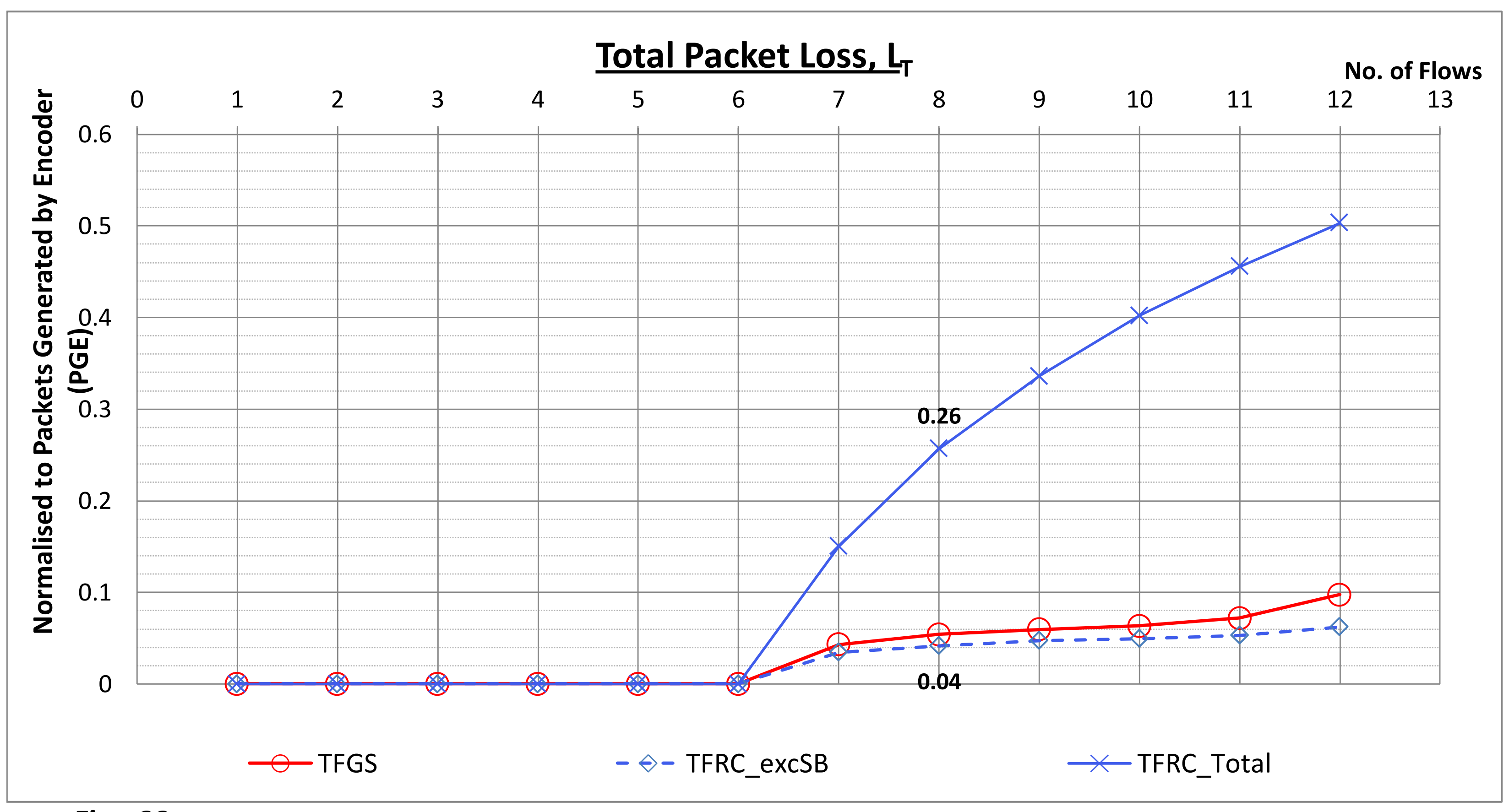


Fig. 23

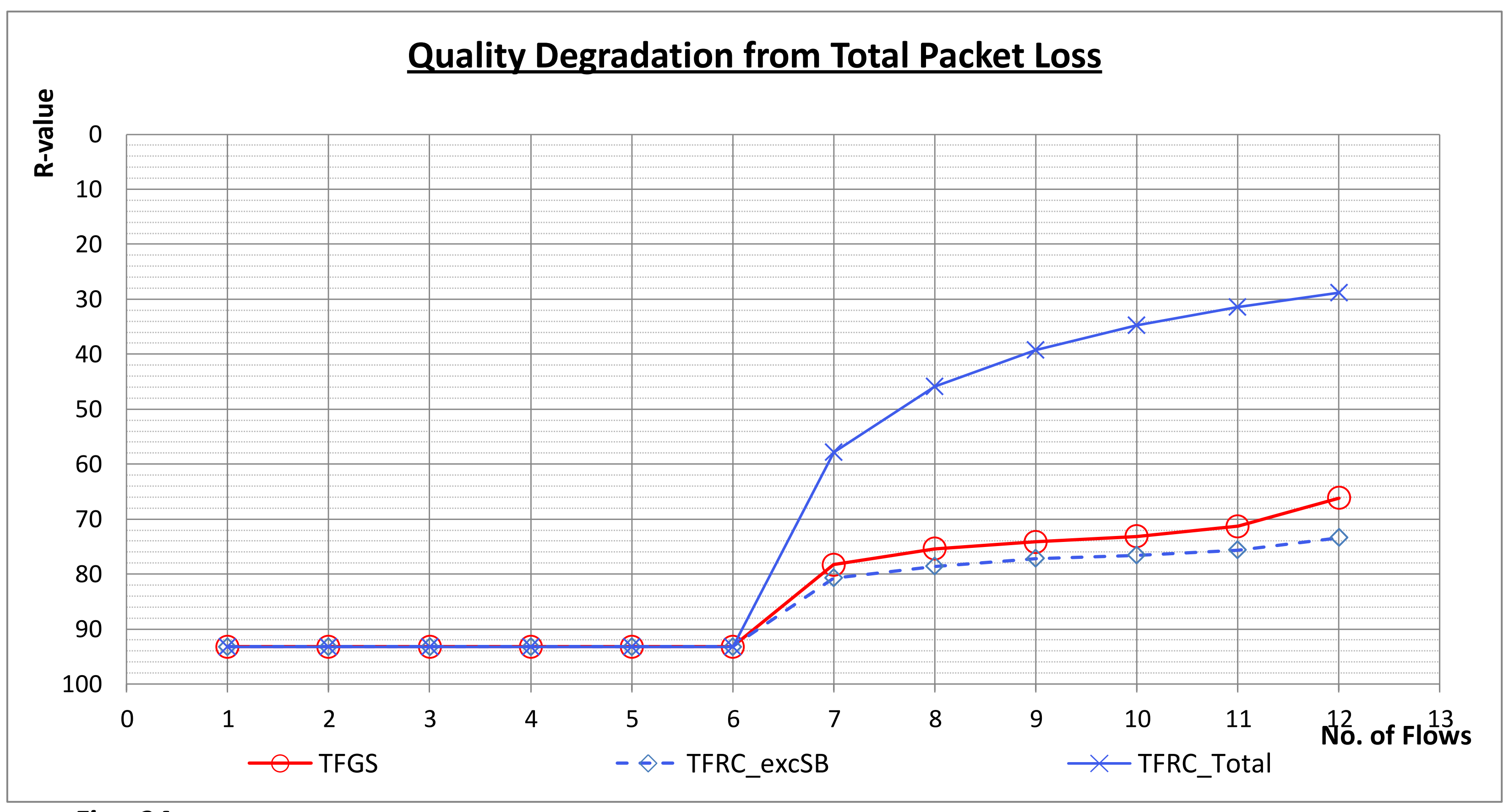


Fig. 24

woTwoSS -28 with Validation, Non-cong.xlsx

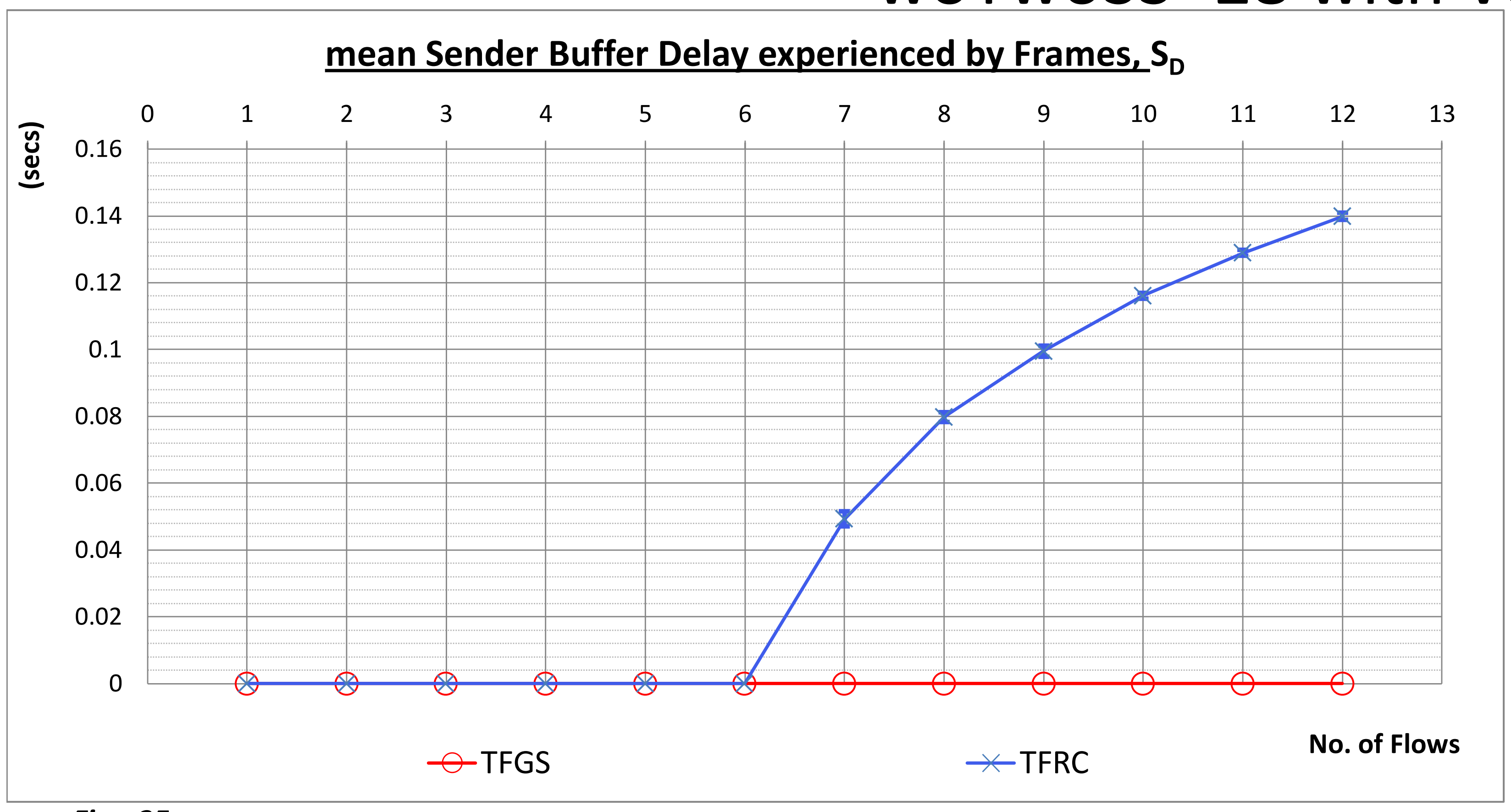


***Fig.* 25**

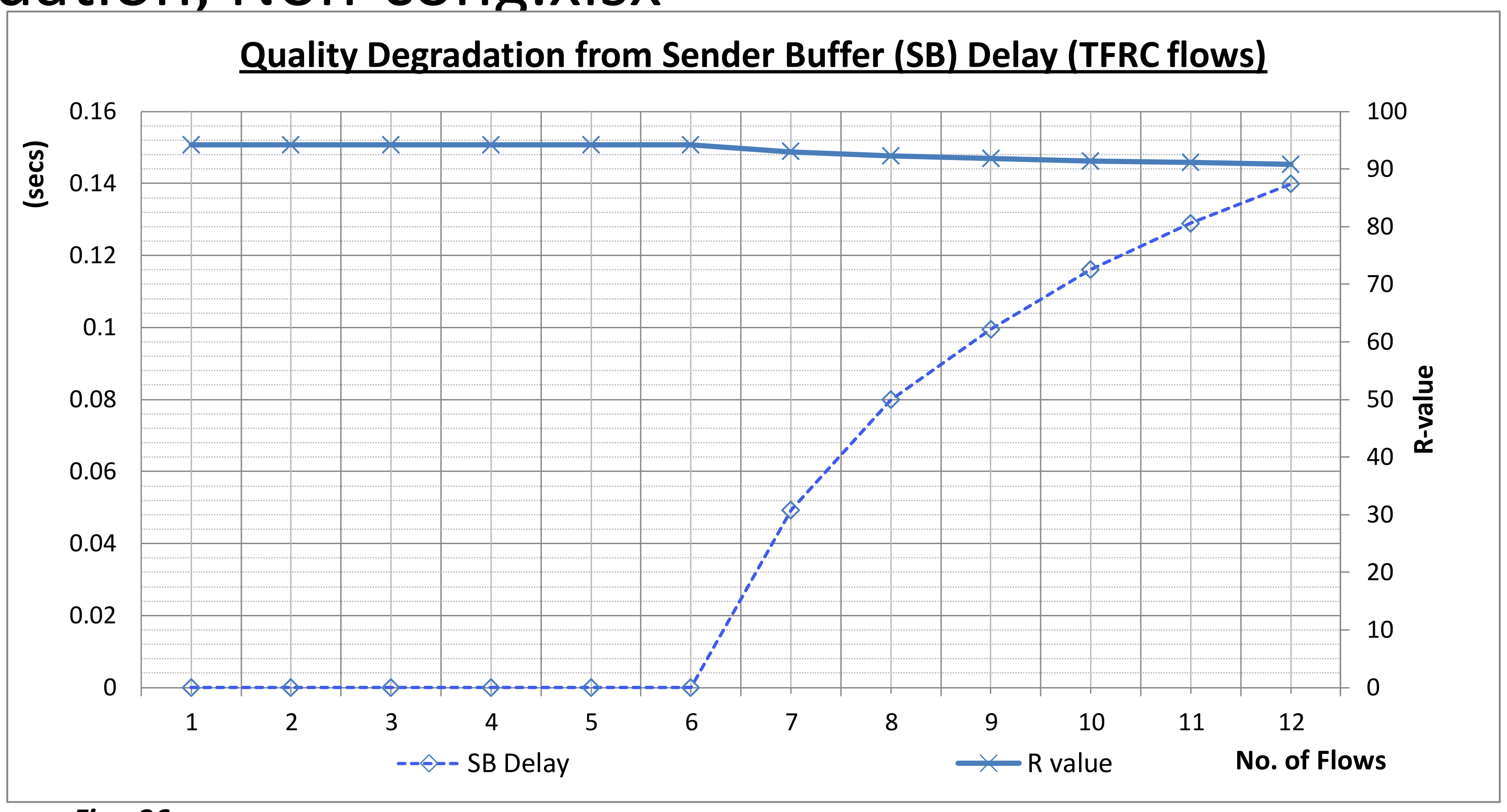


***Fig.* 26**

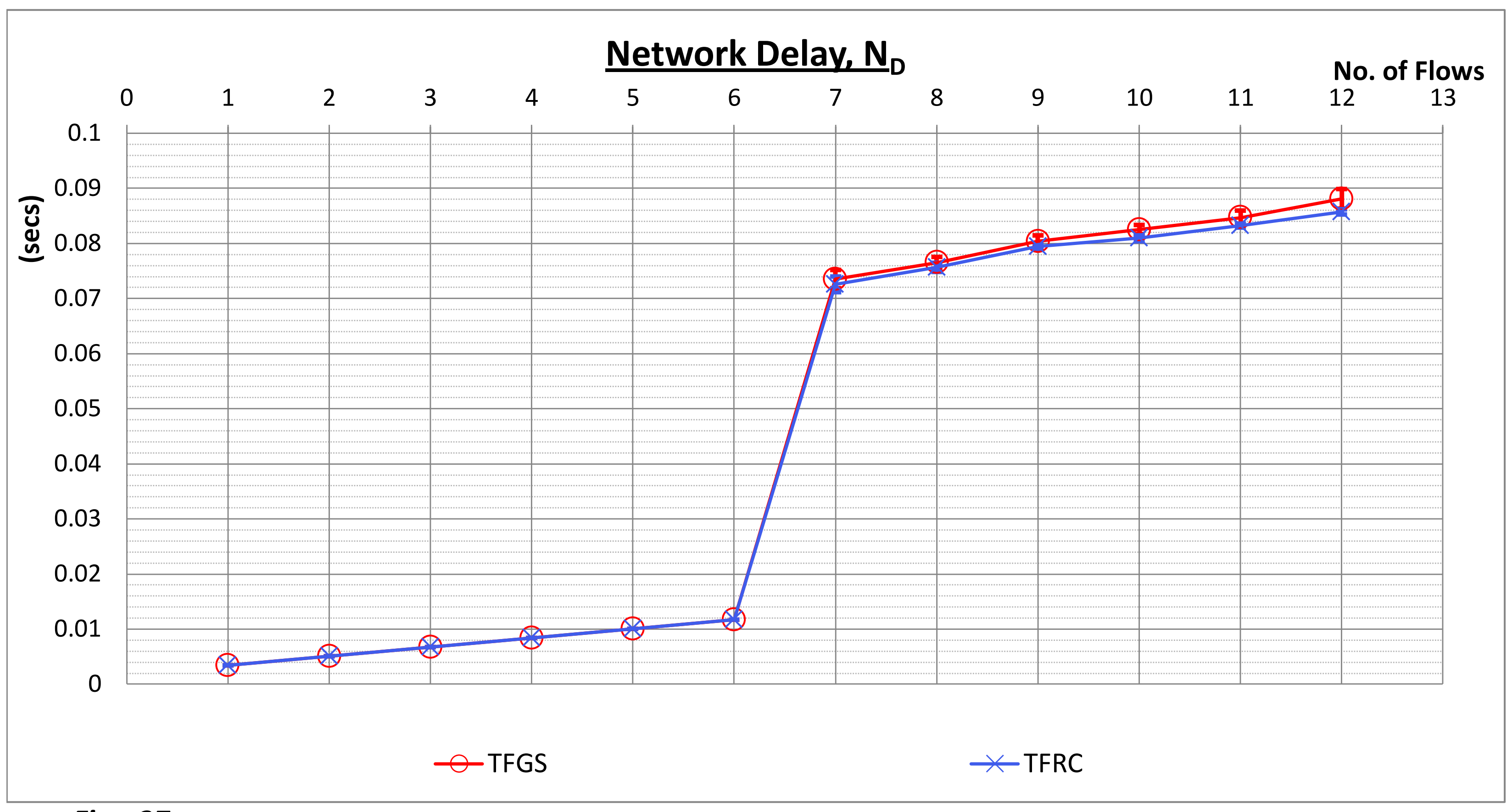


***Fig.* 27**

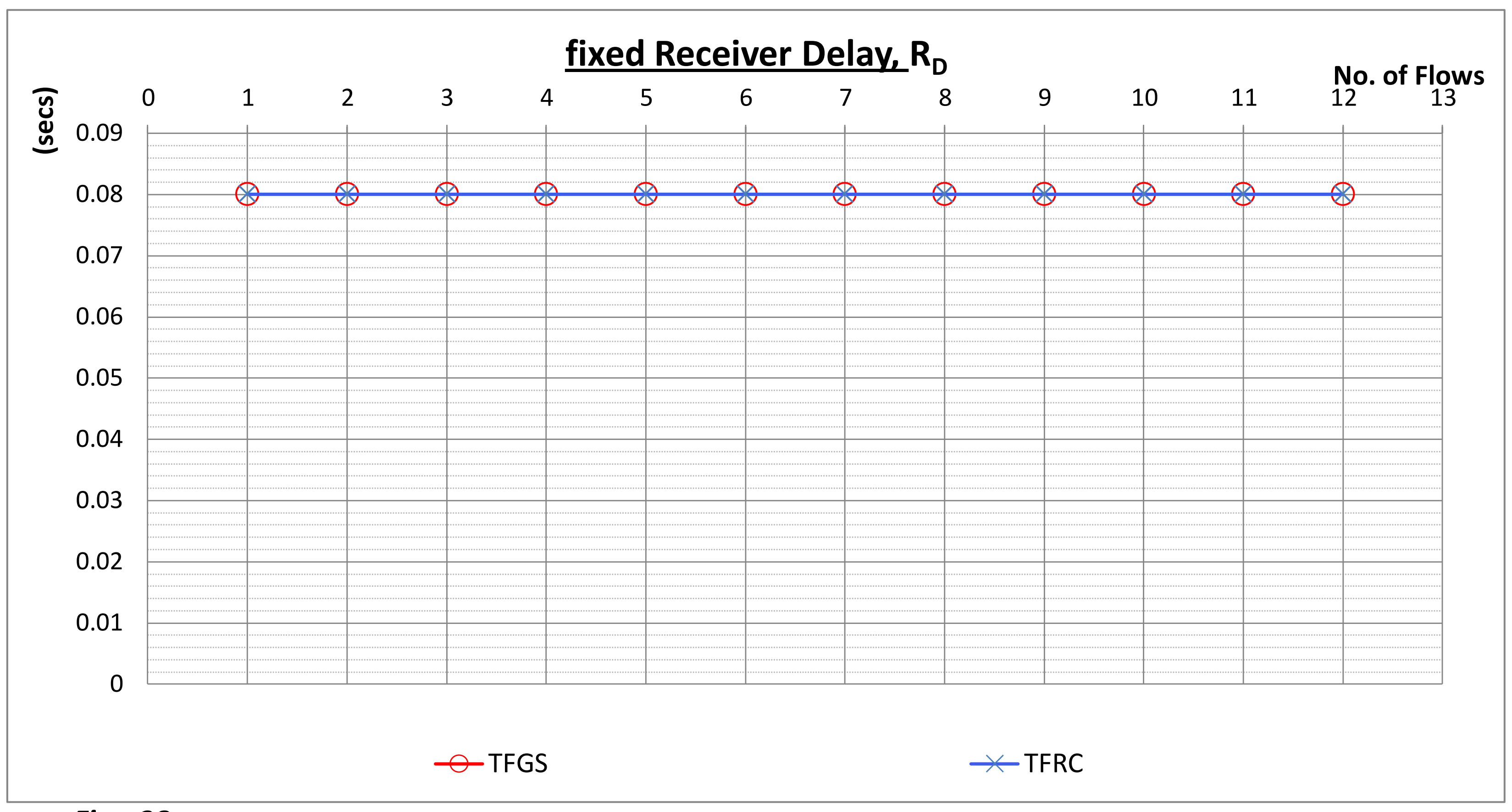


***Fig.* 28**

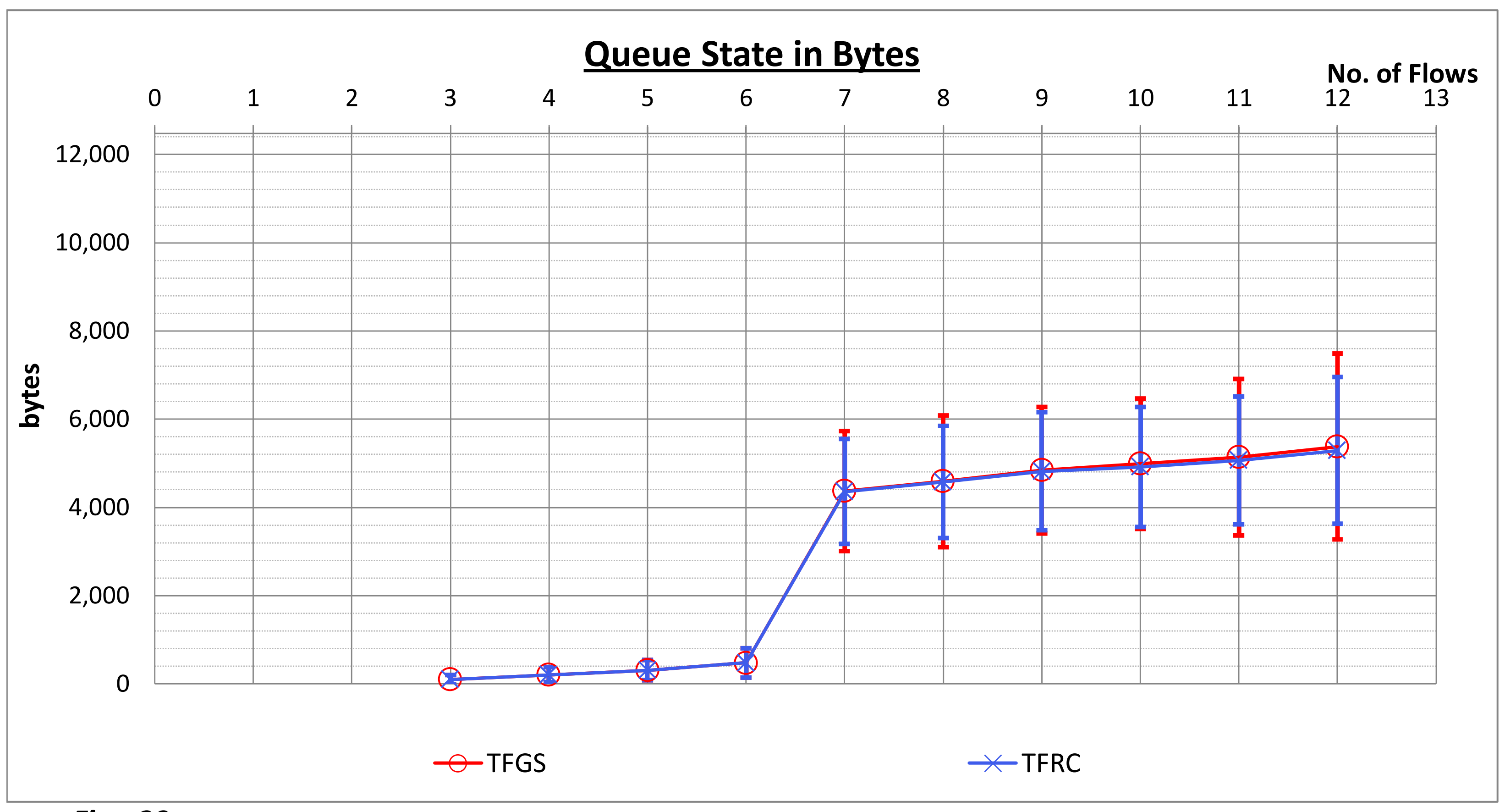


***Fig.* 29**

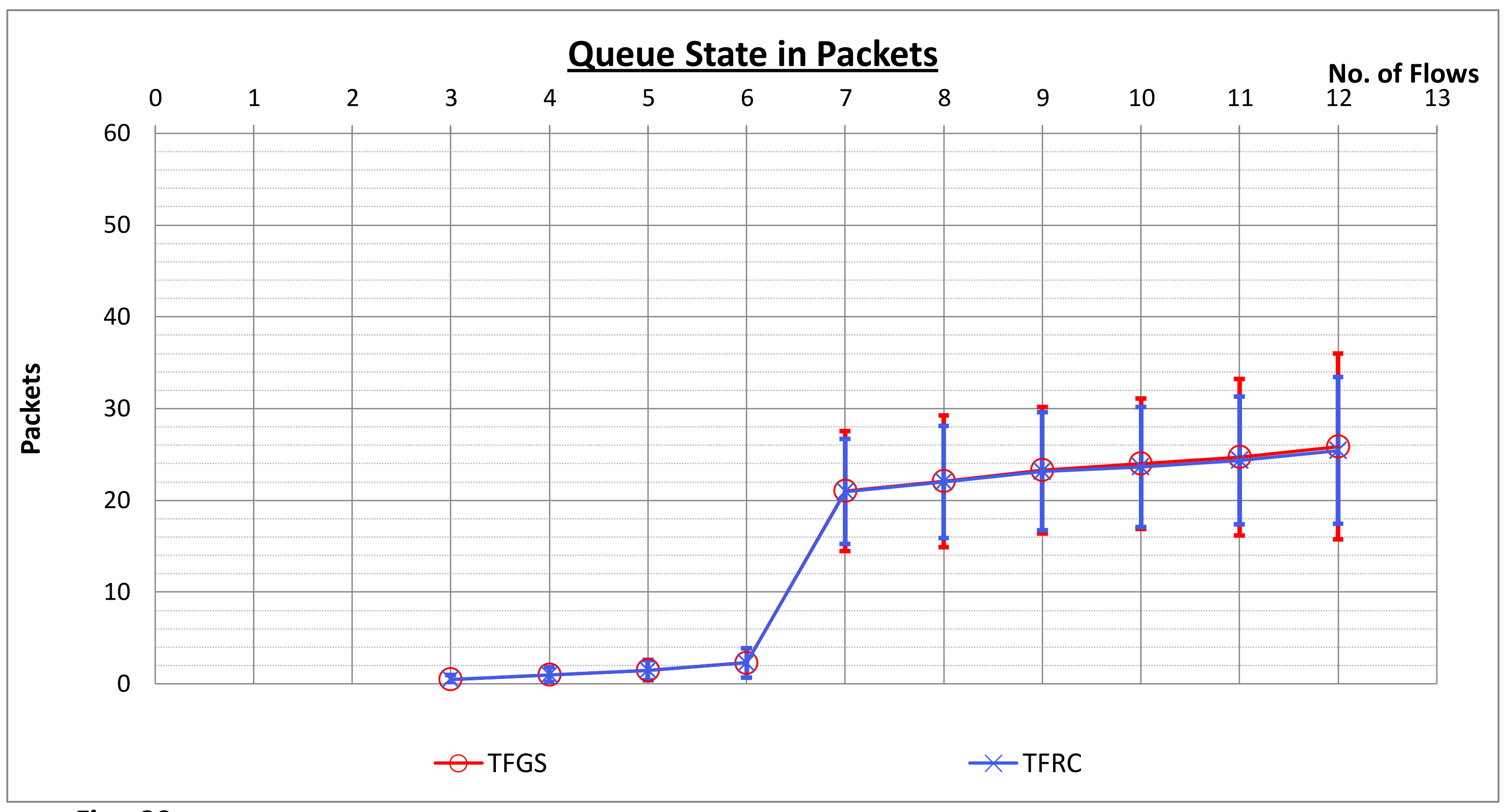


***Fig.* 30**

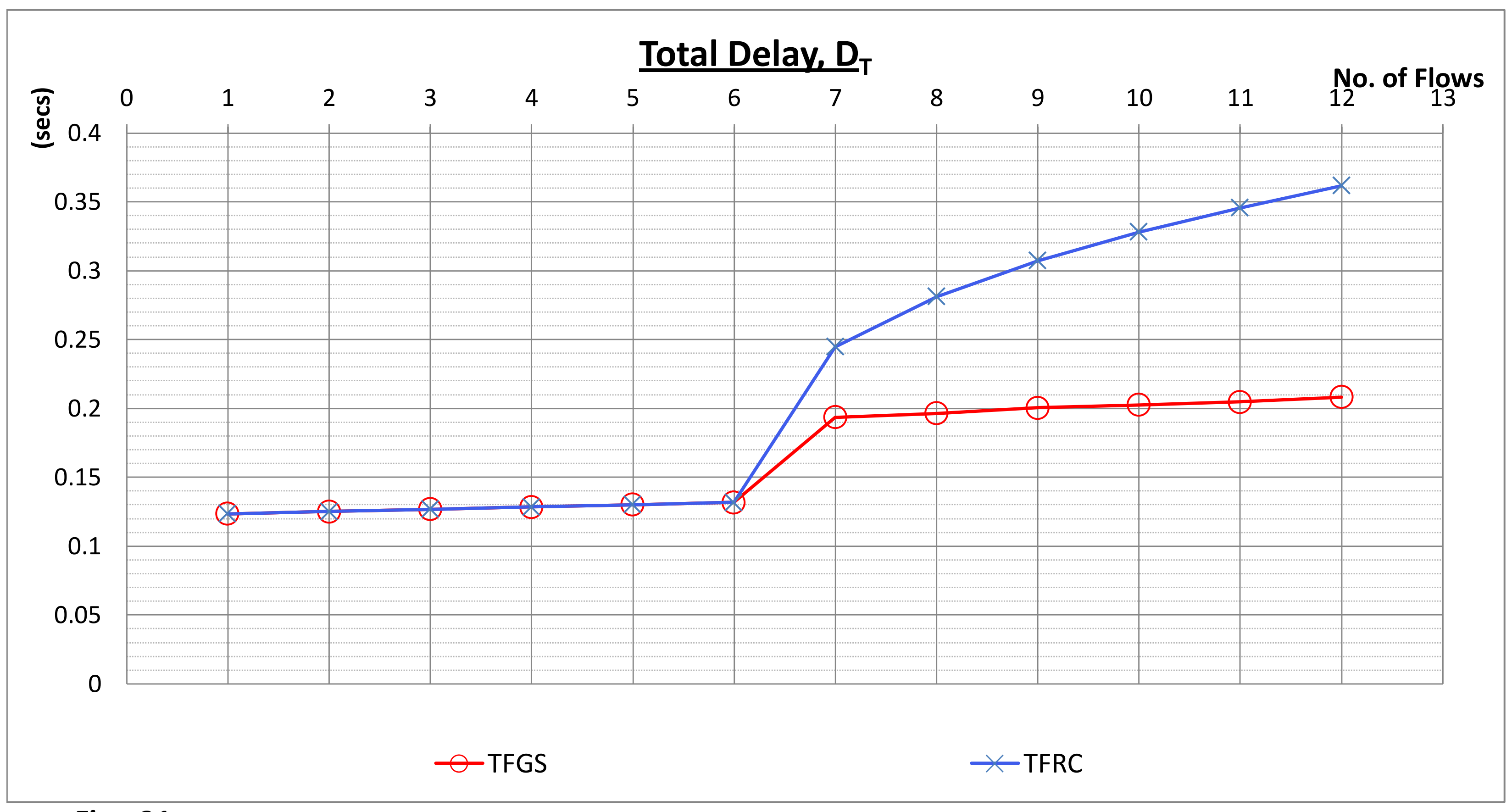


***Fig.* 31**

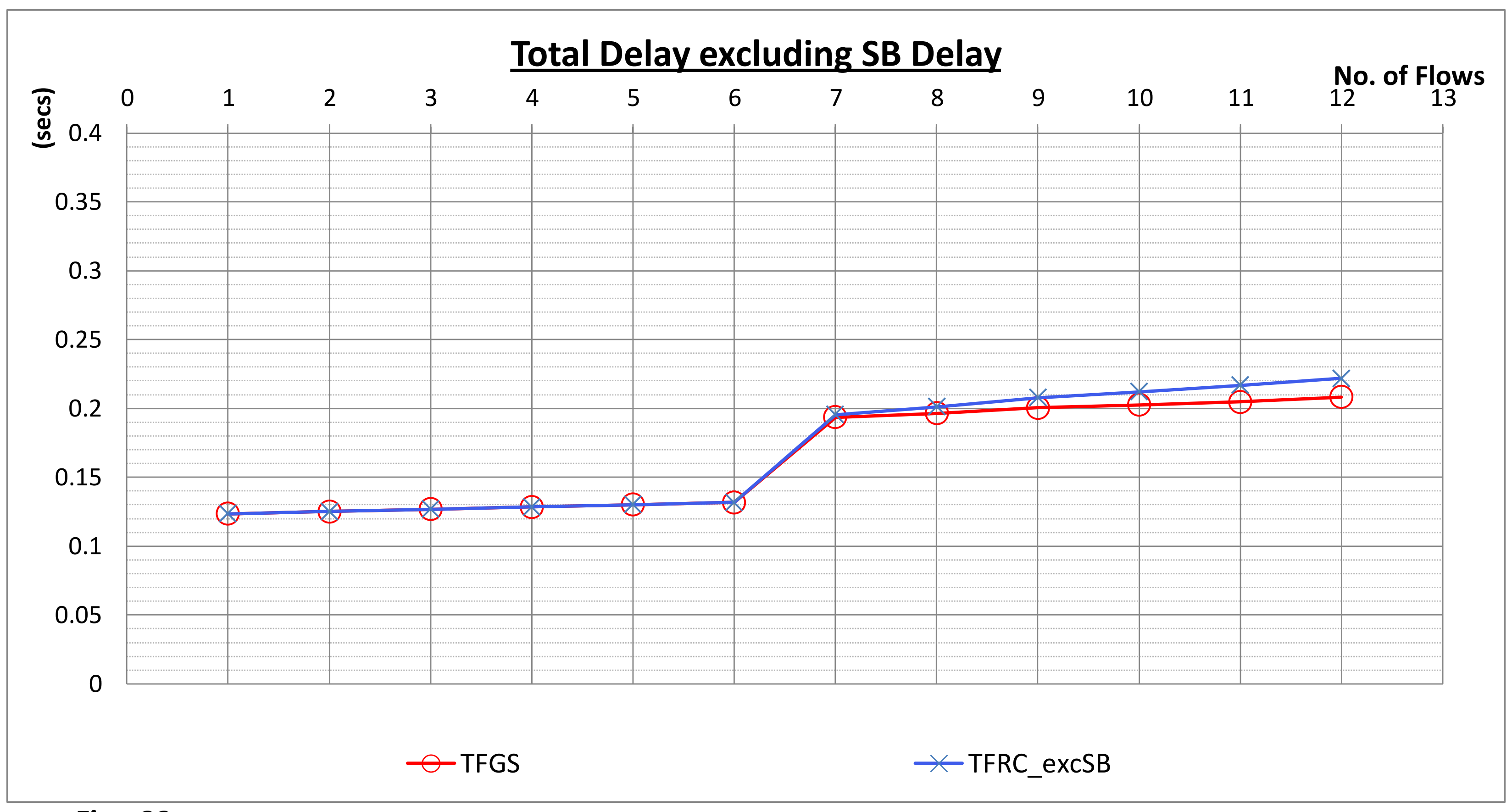


***Fig.* 32**

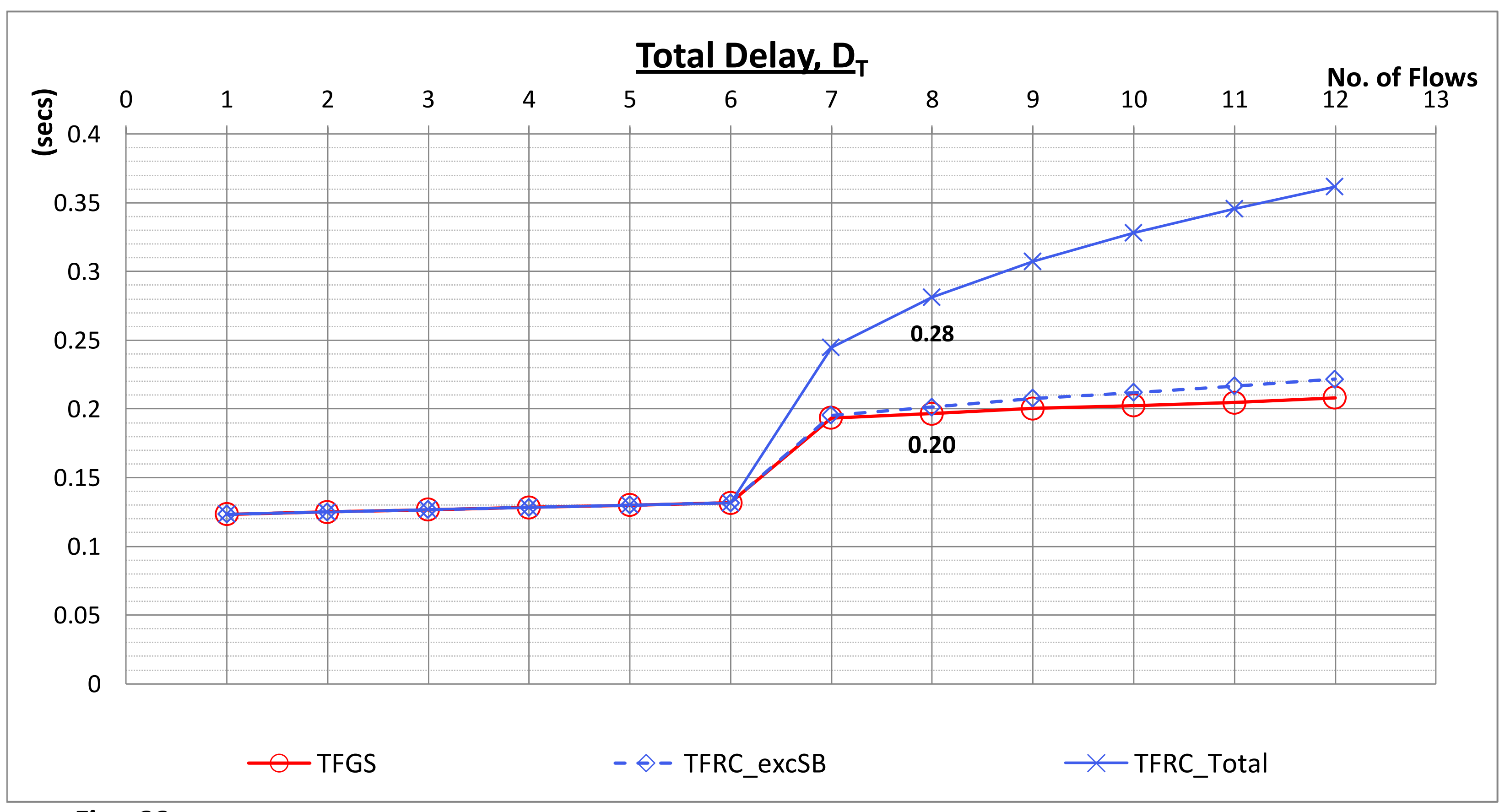


***Fig.* 33**

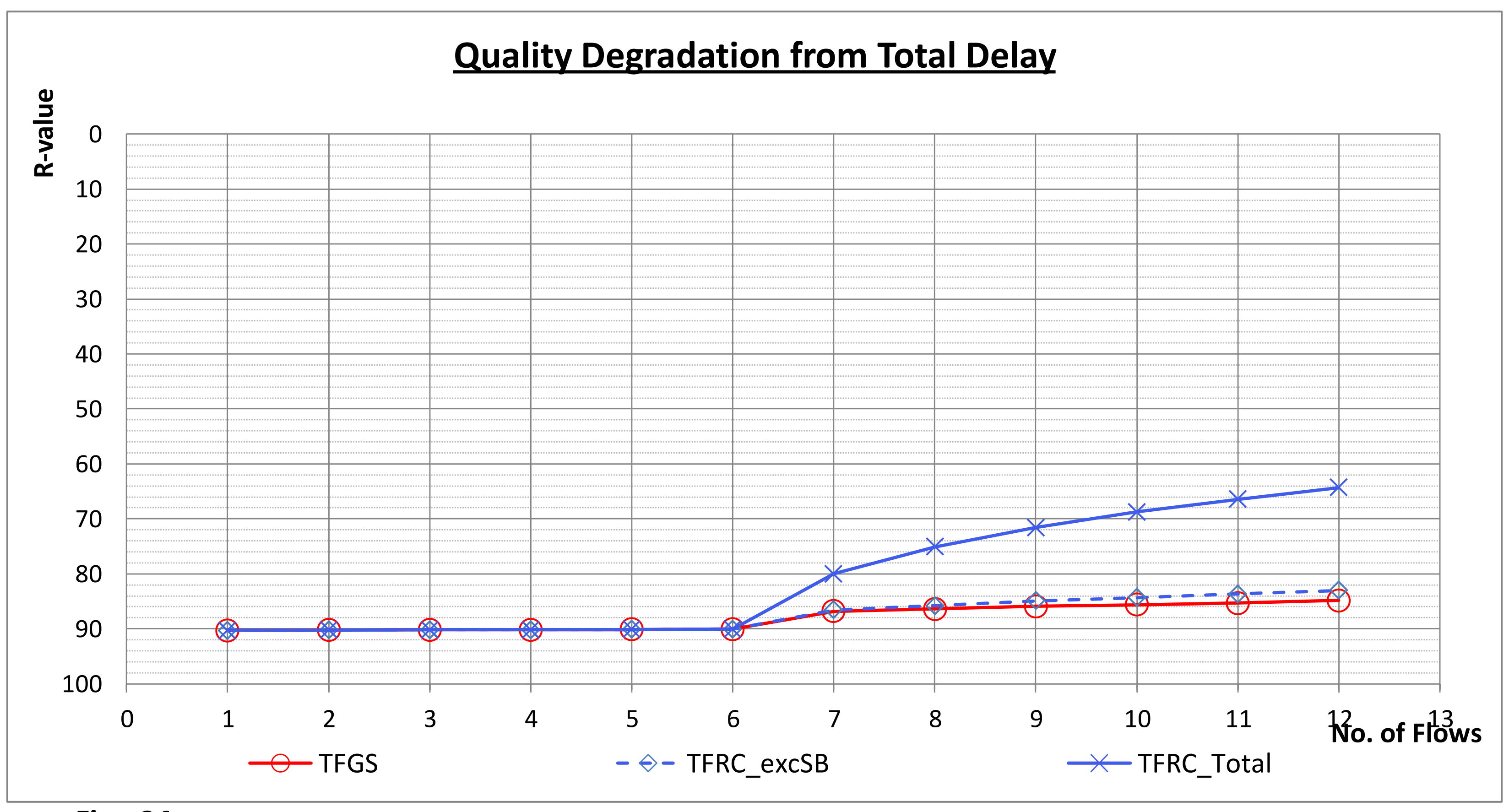


***Fig.* 34**

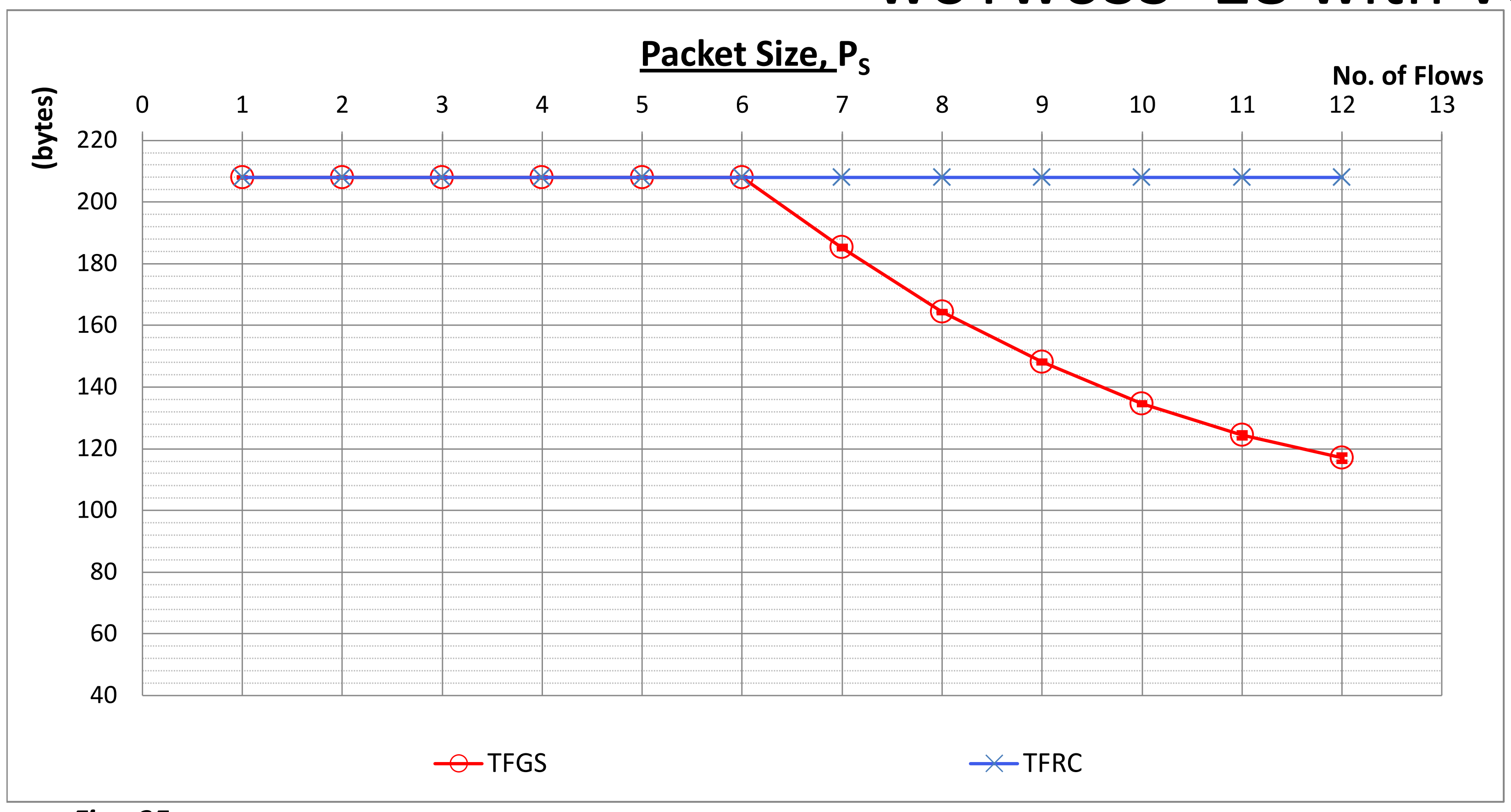


Fig. *35*

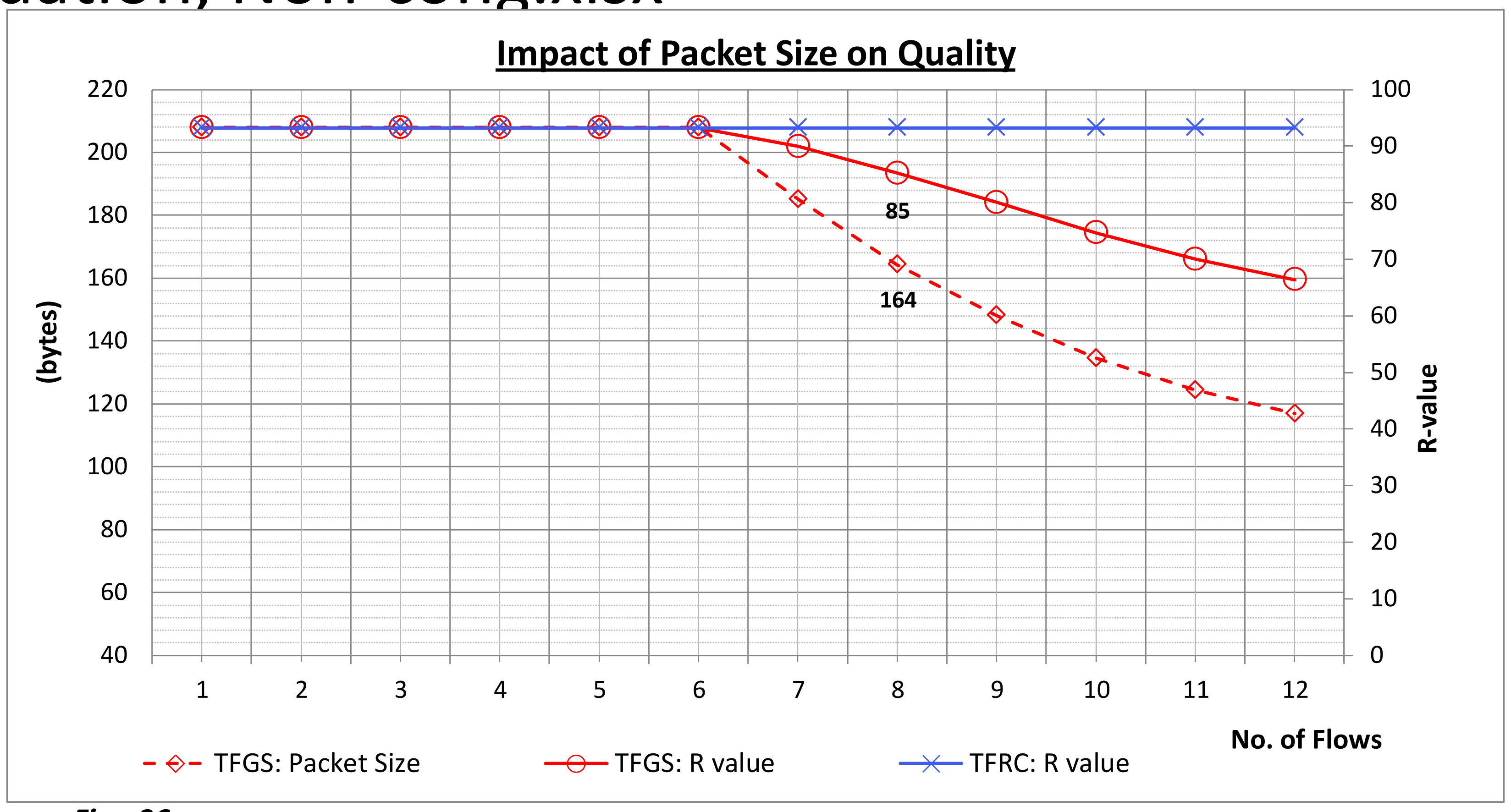


Fig. *36*

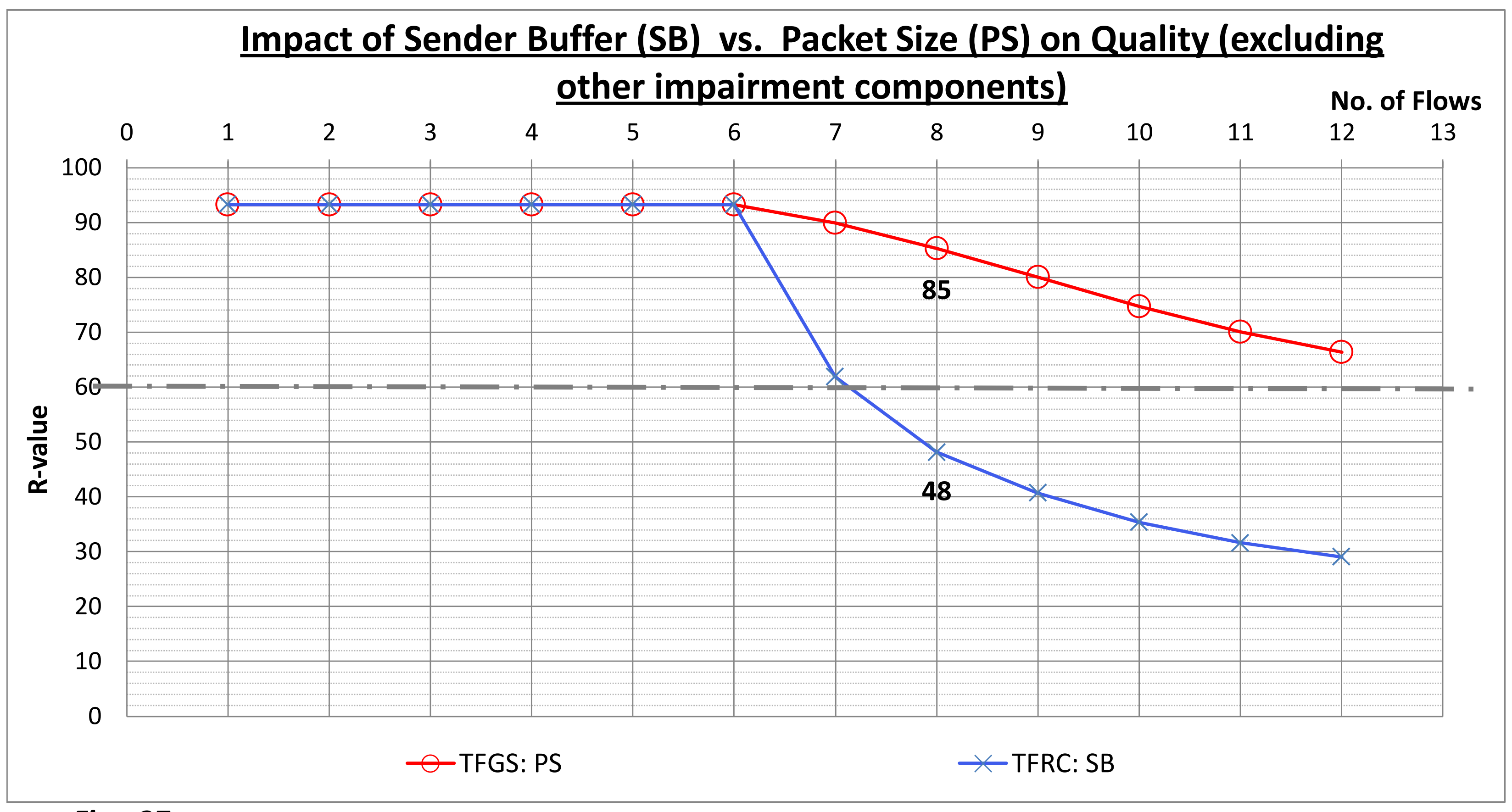


Fig. *37*

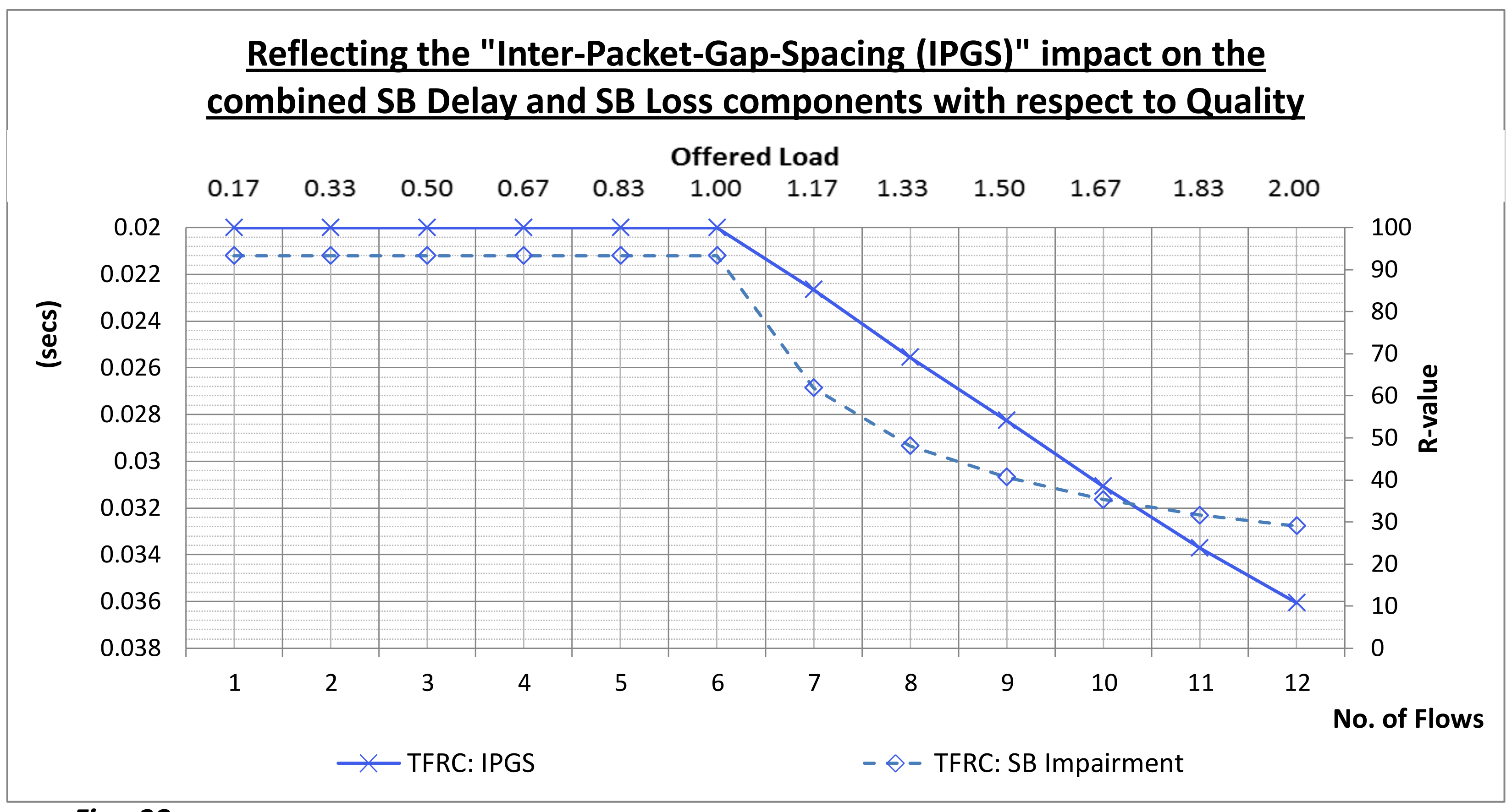


Fig. *38*

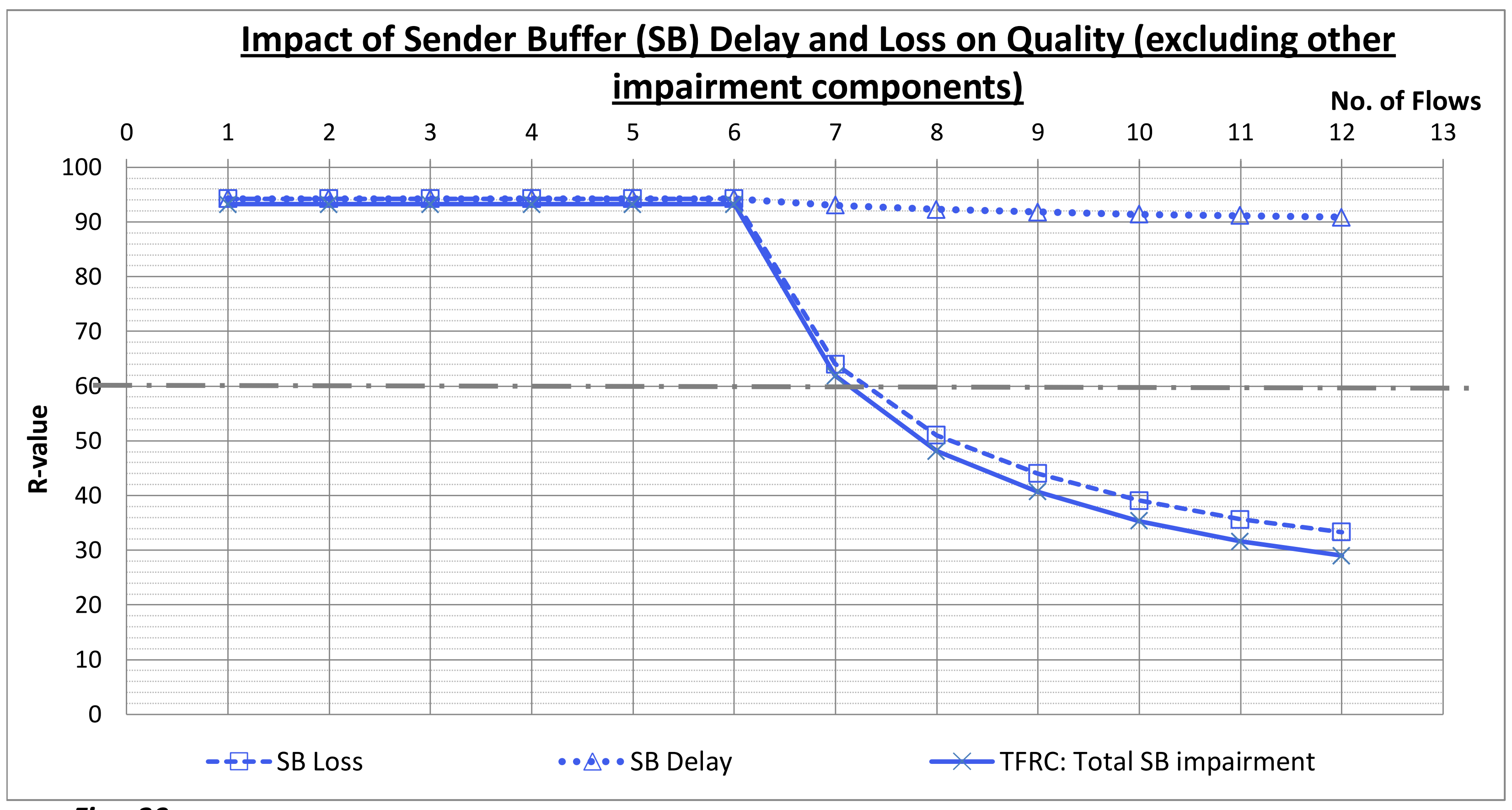


Fig. *39*

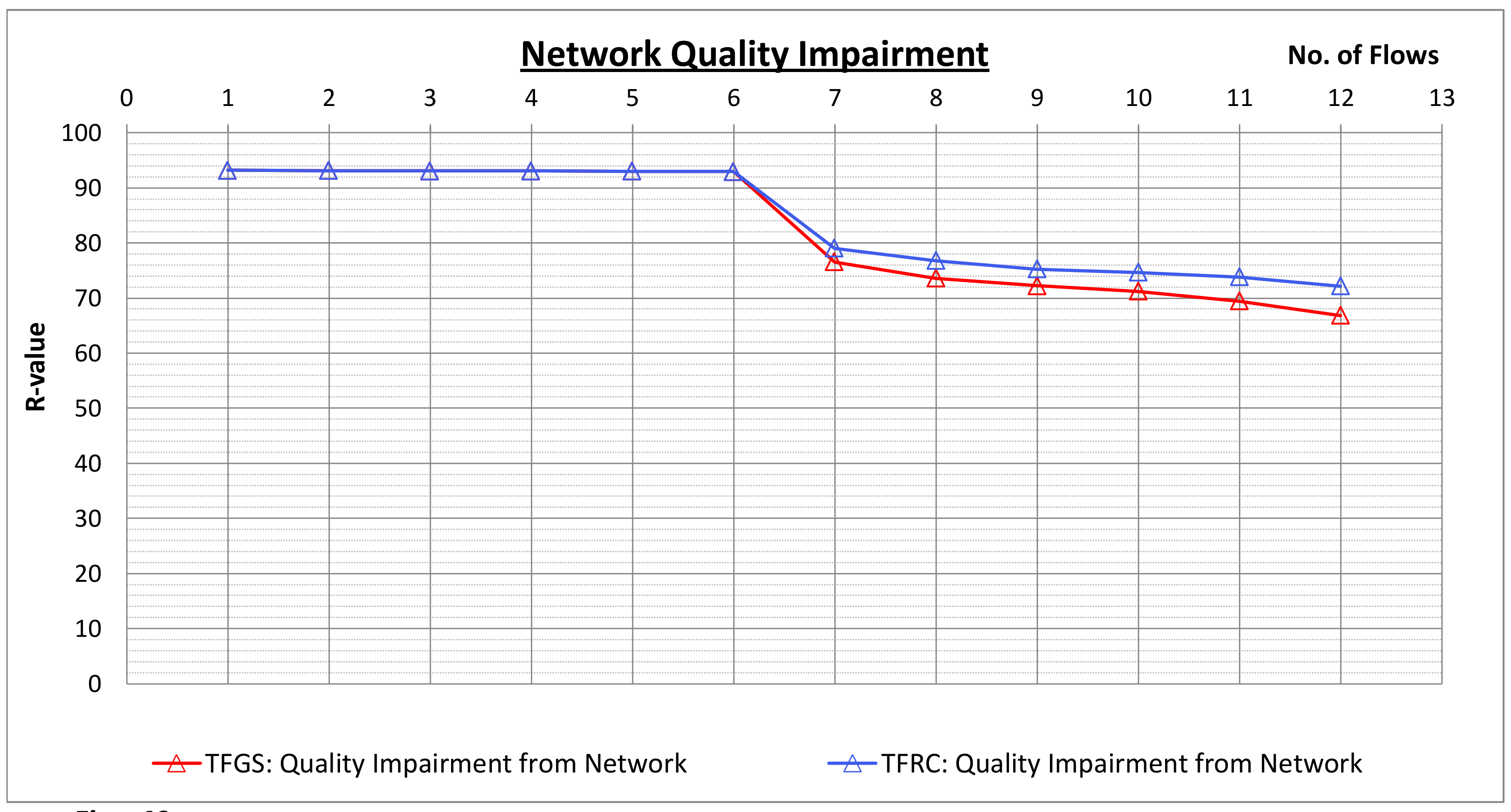


Fig. *40*

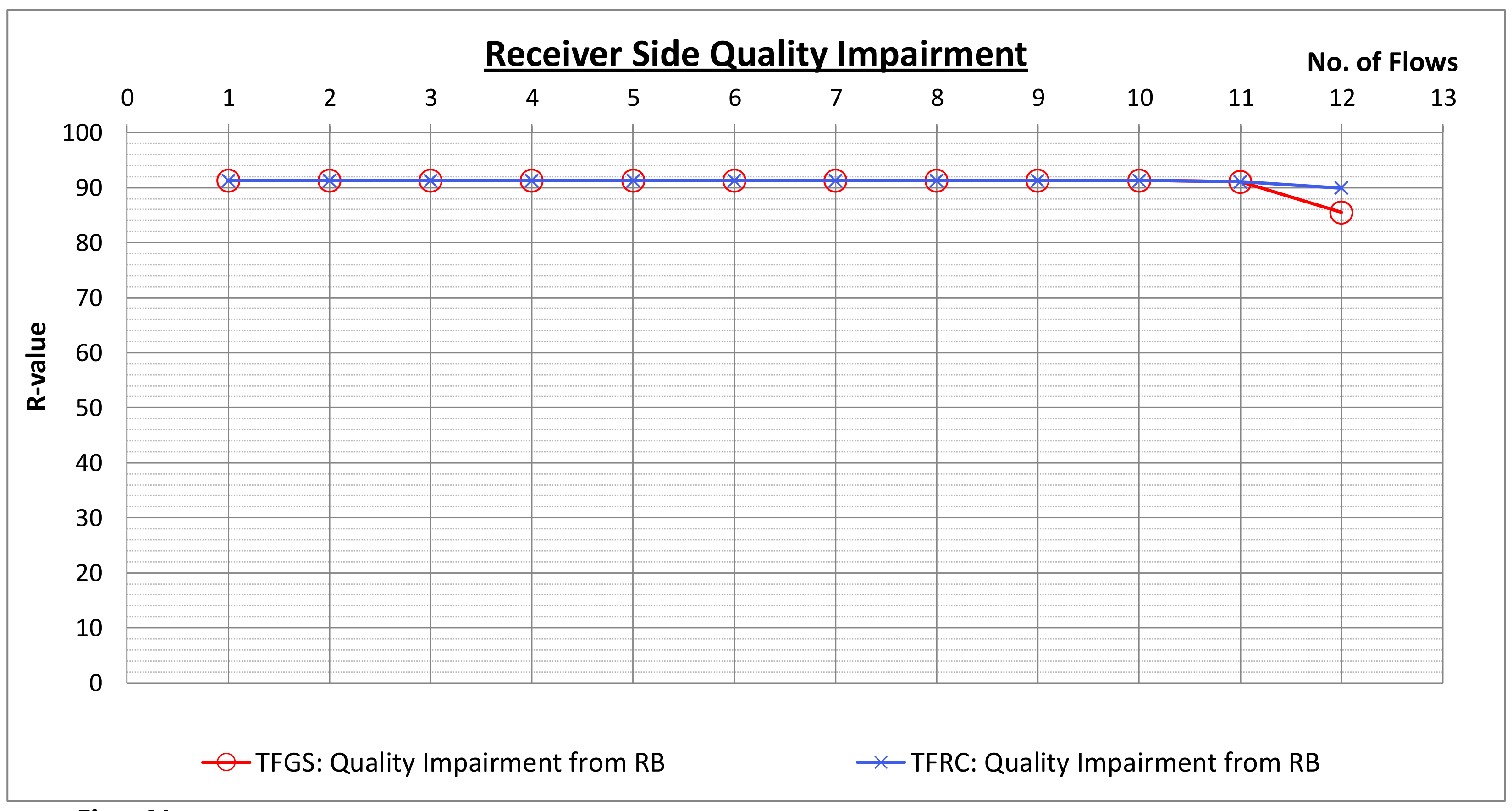


Fig. *41*

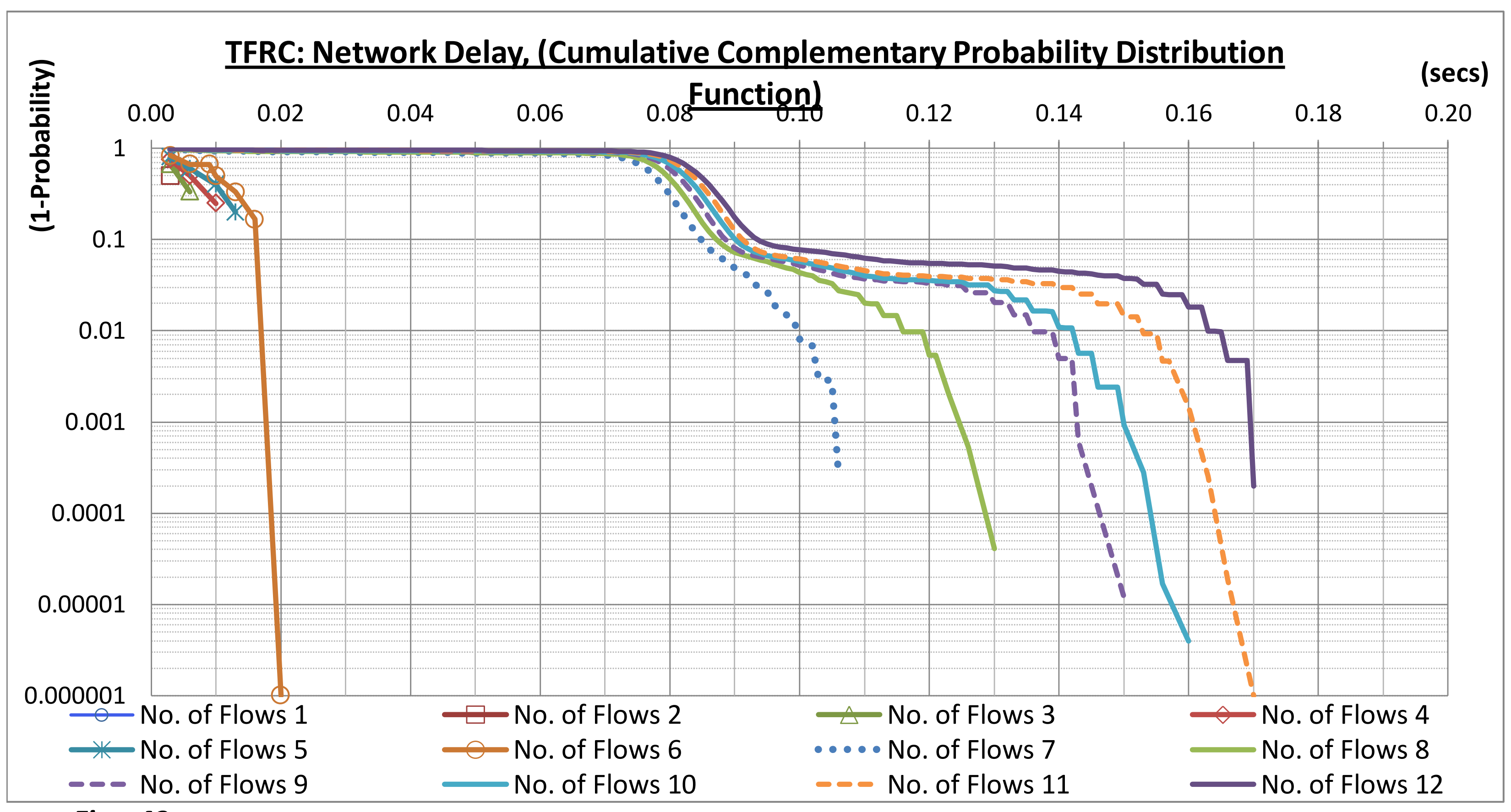


Fig. *43*

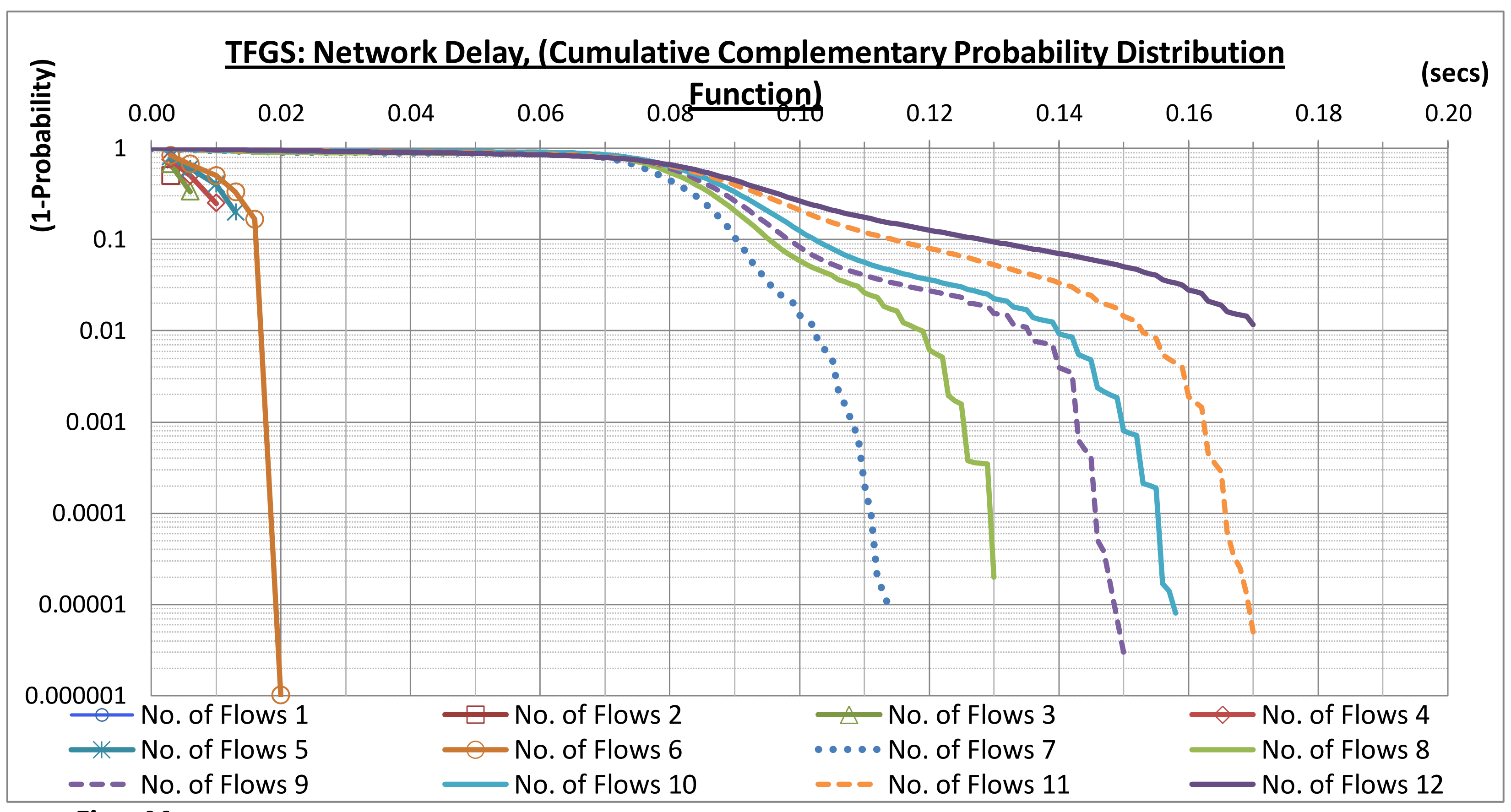


Fig. *44*

woTwoSS -28 with Validation, Non-cong.xlsx

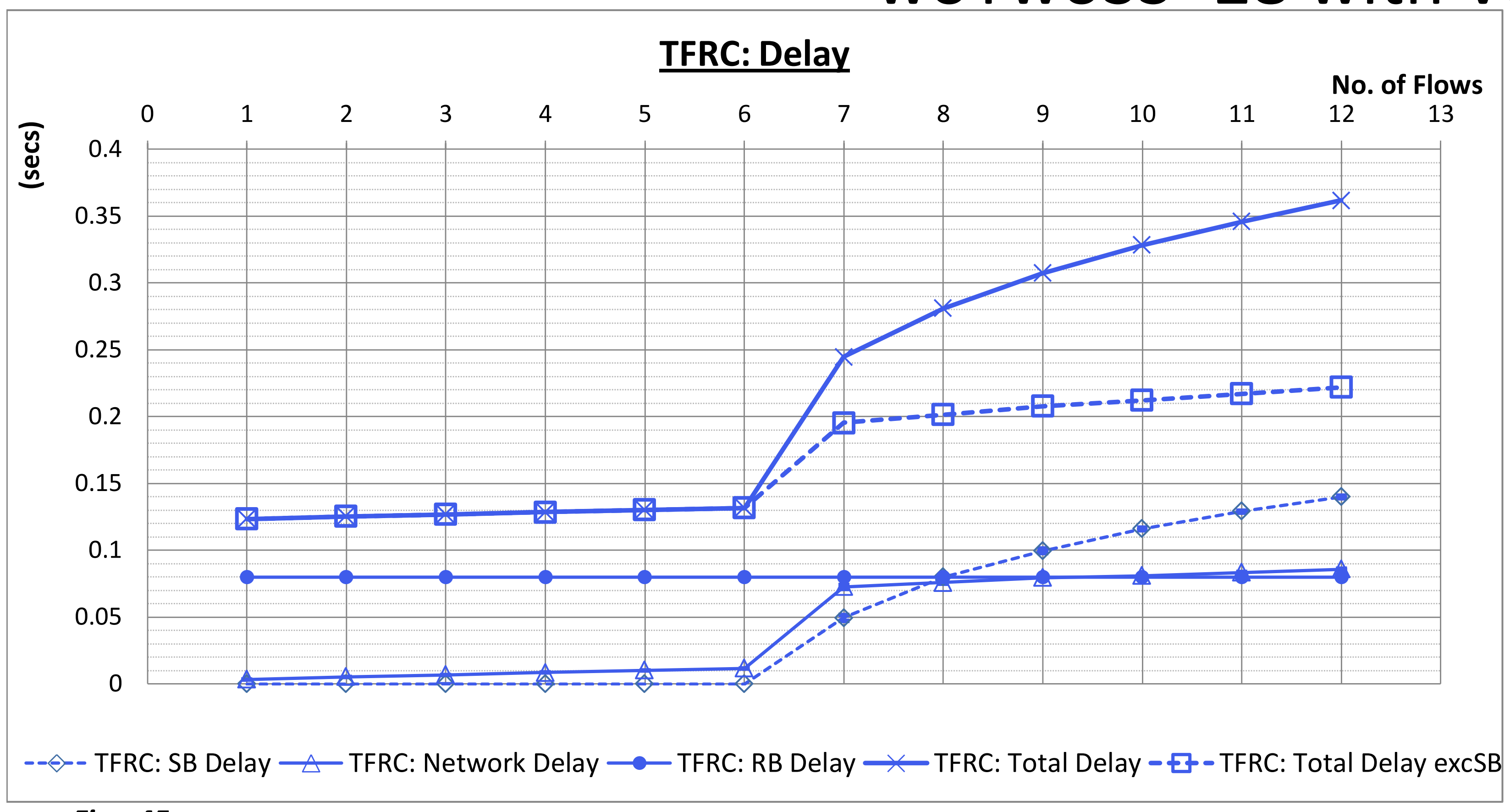


***Fig.* 45**

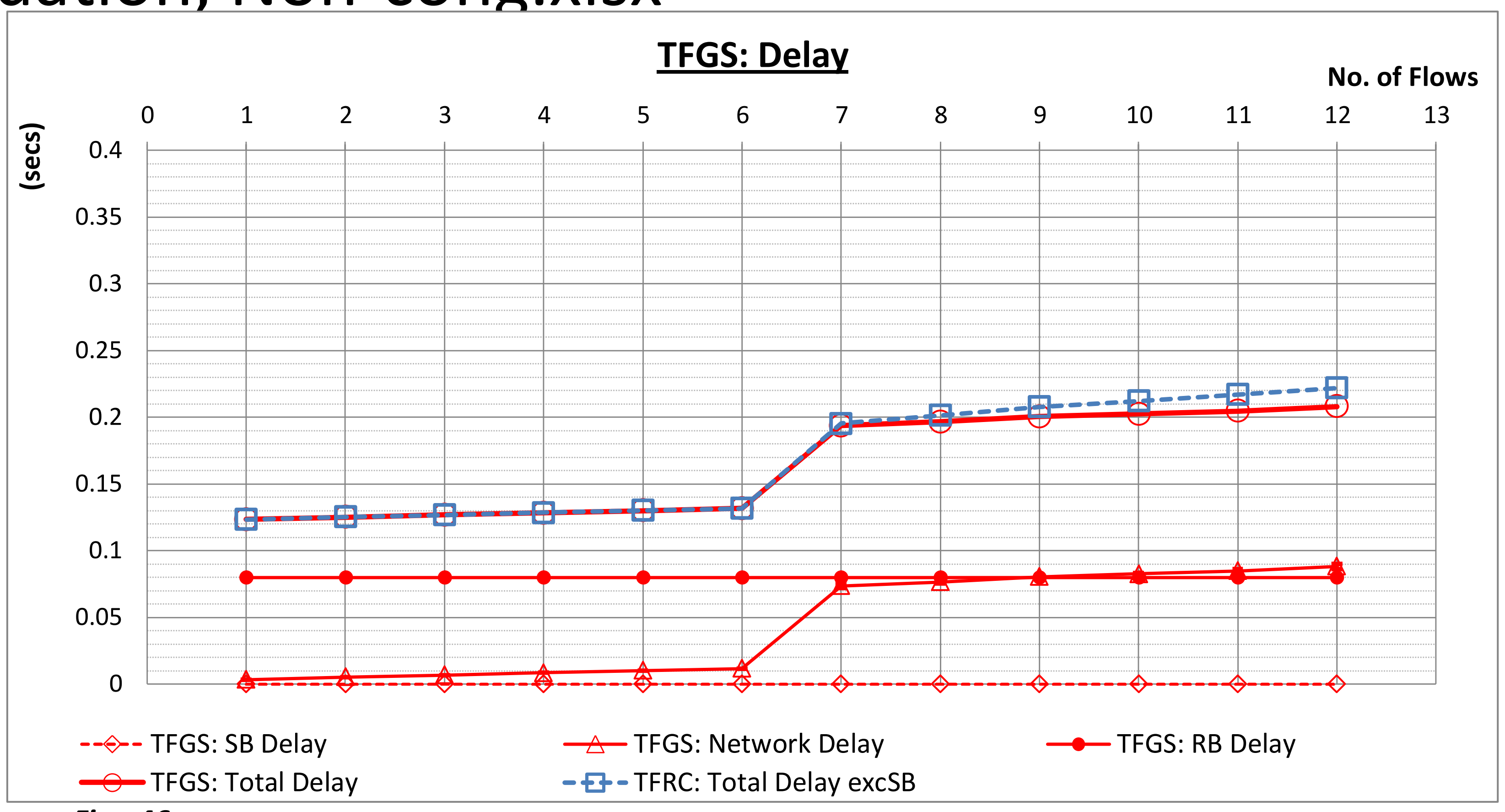


***Fig.* 46**

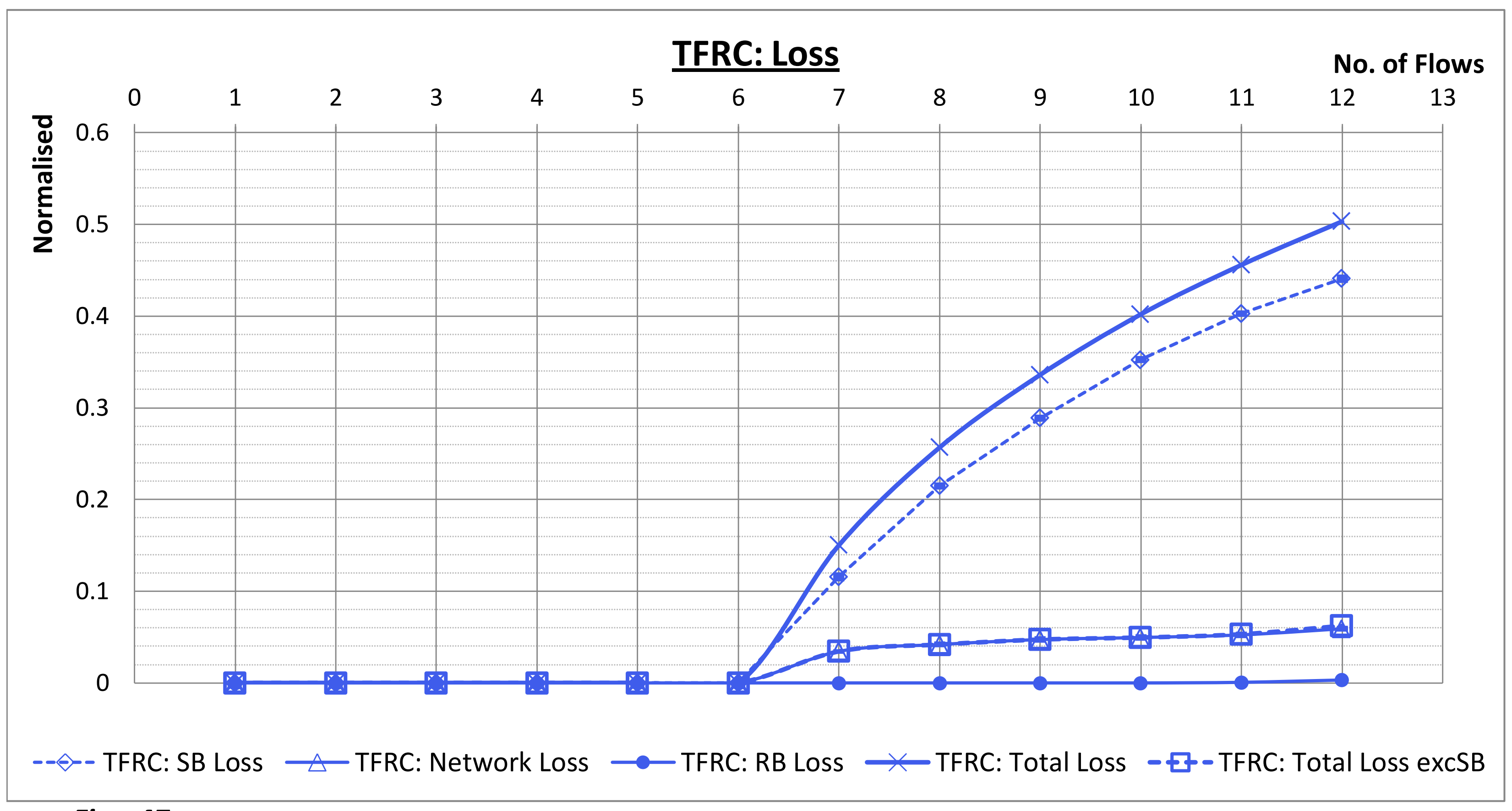


***Fig.* 47**

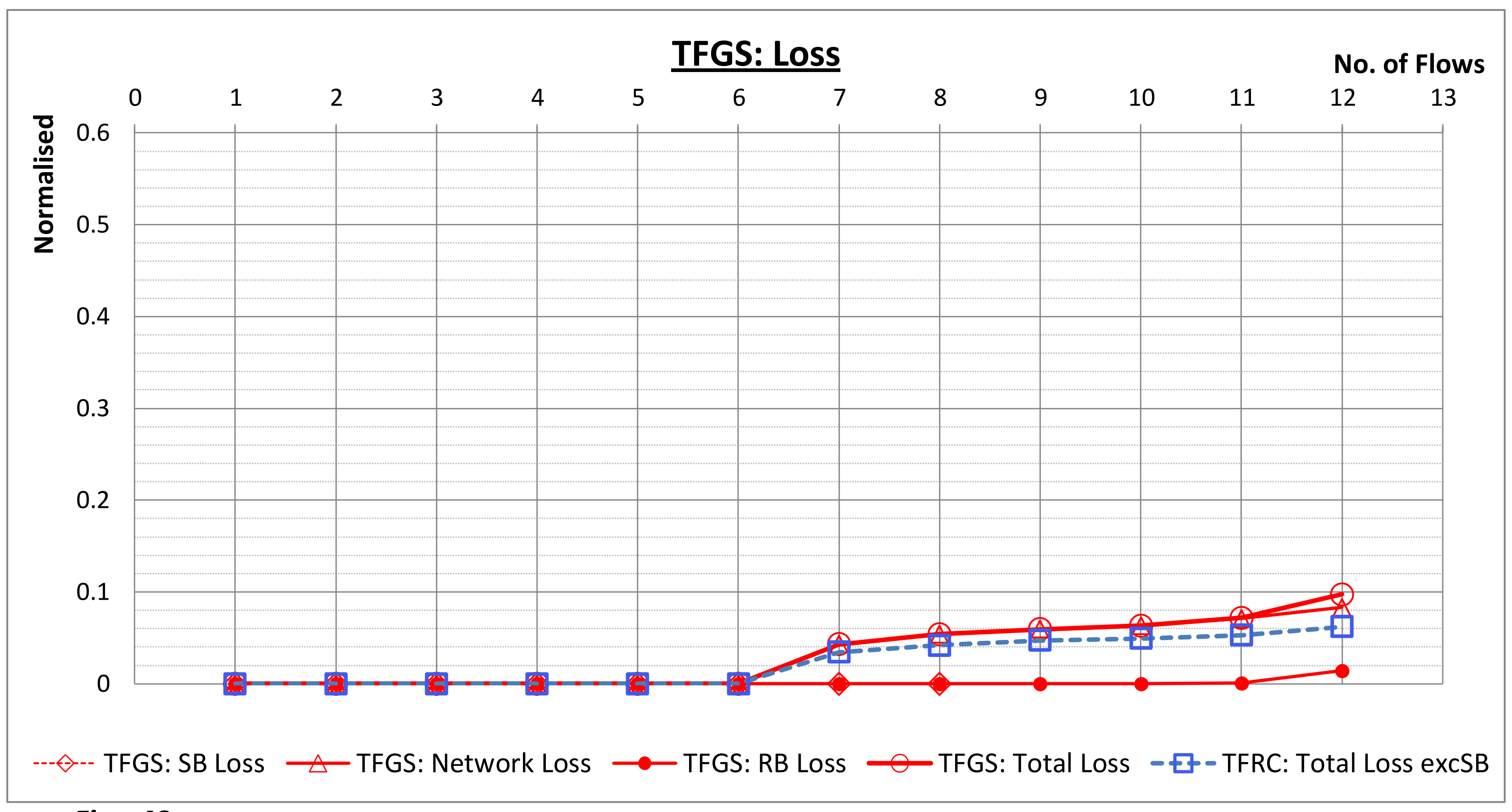


***Fig.* 48**

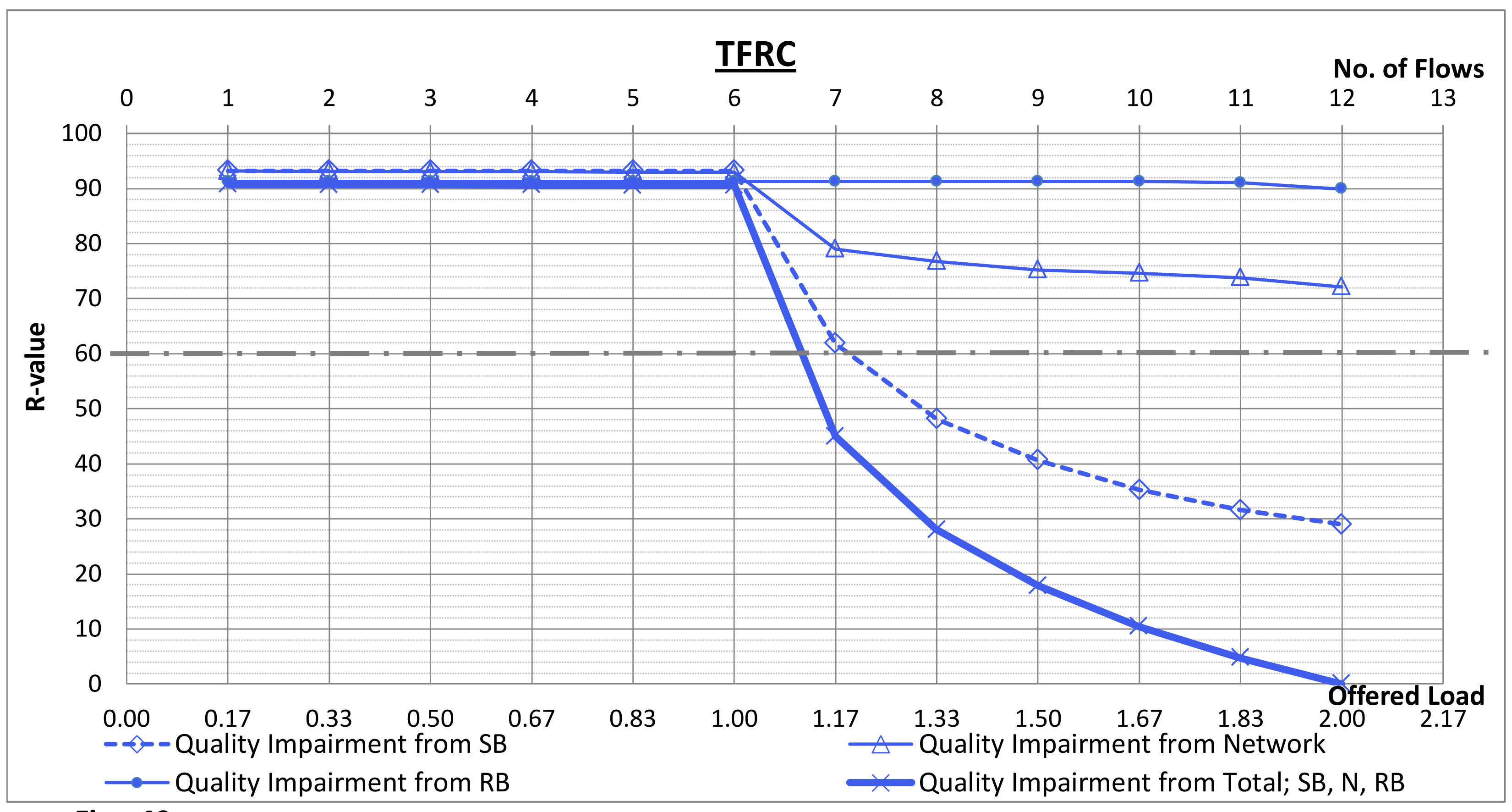


***Fig.* 49**

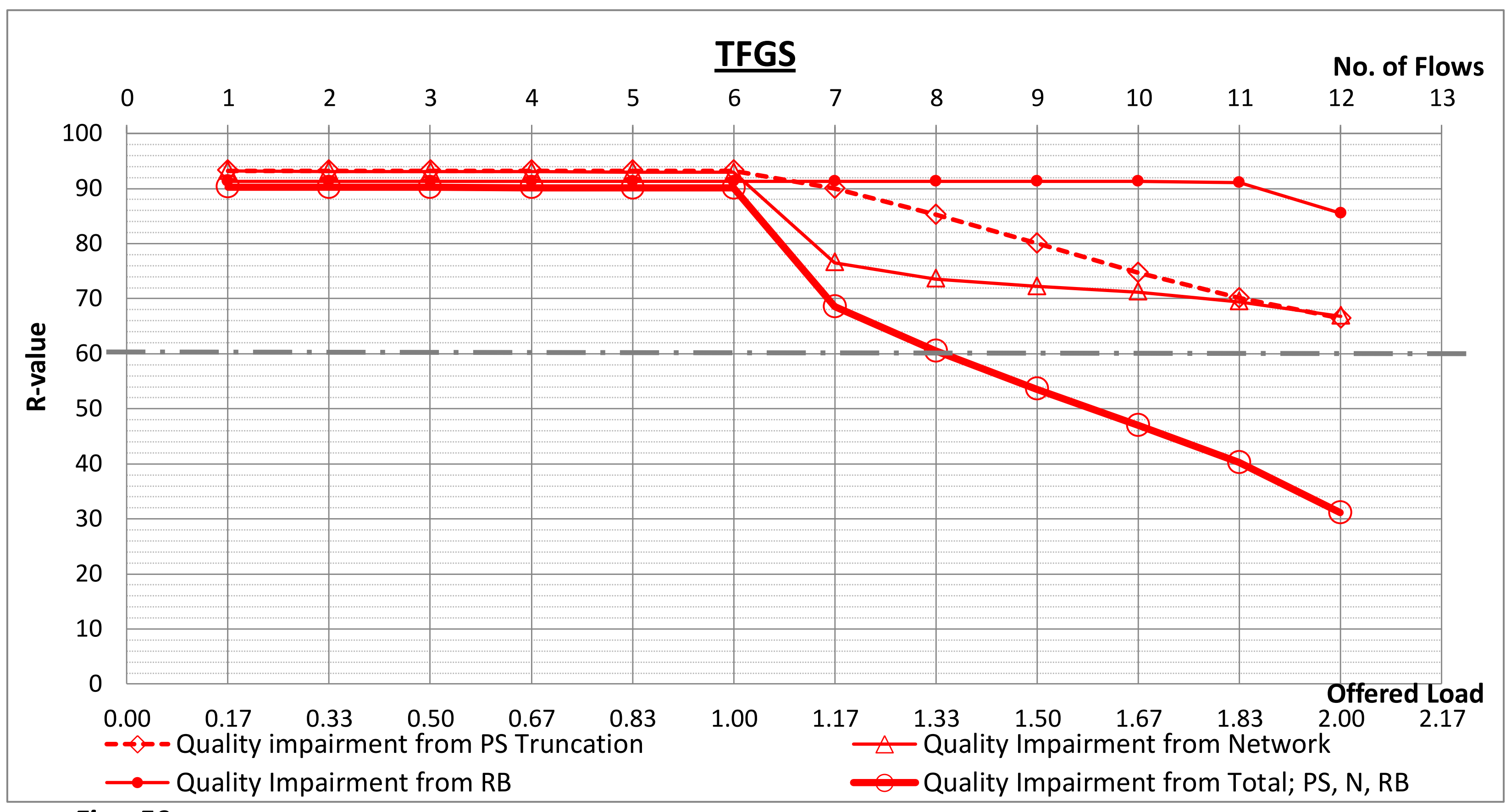


***Fig.* 50**

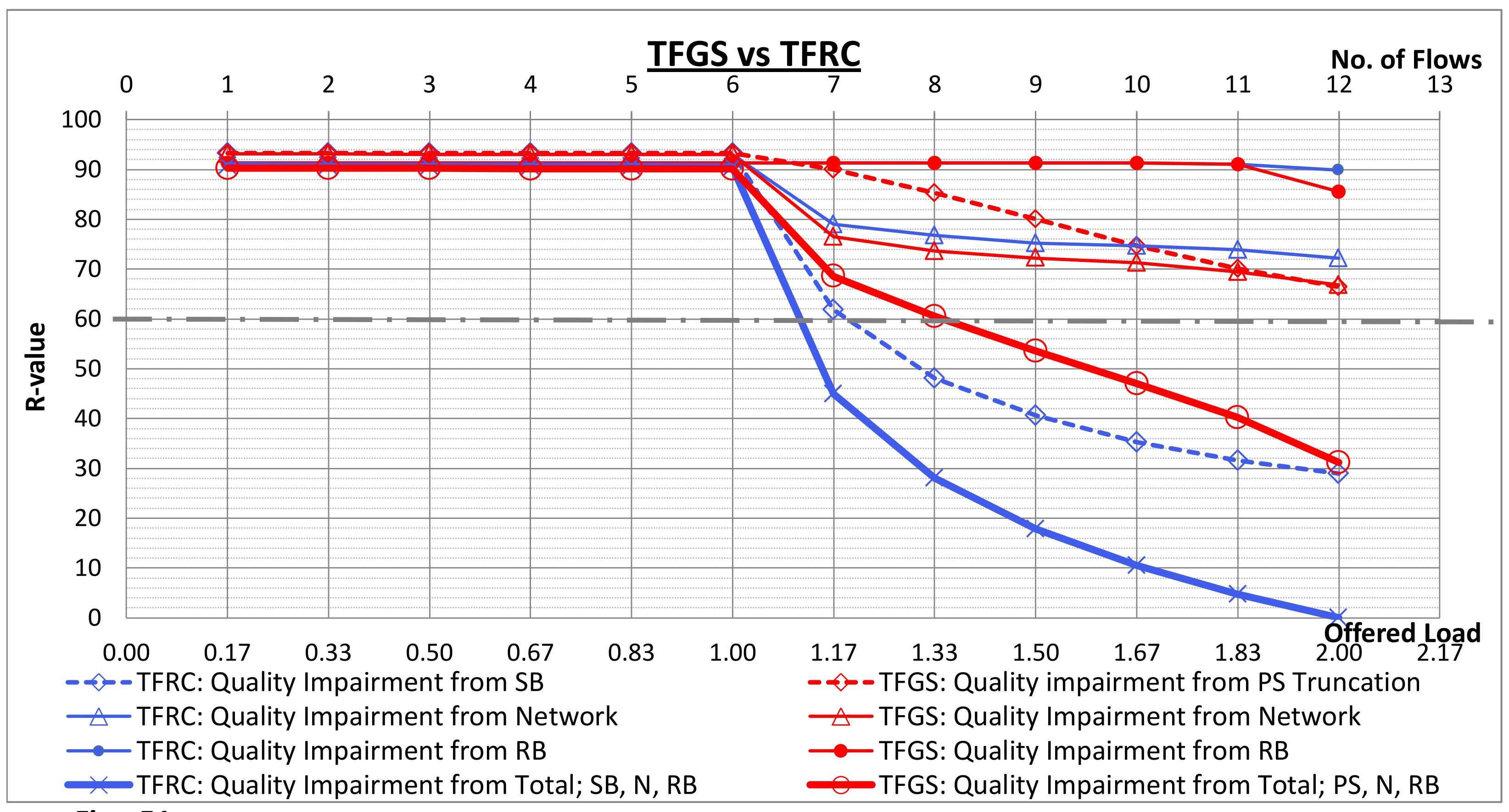


***Fig.* 51**

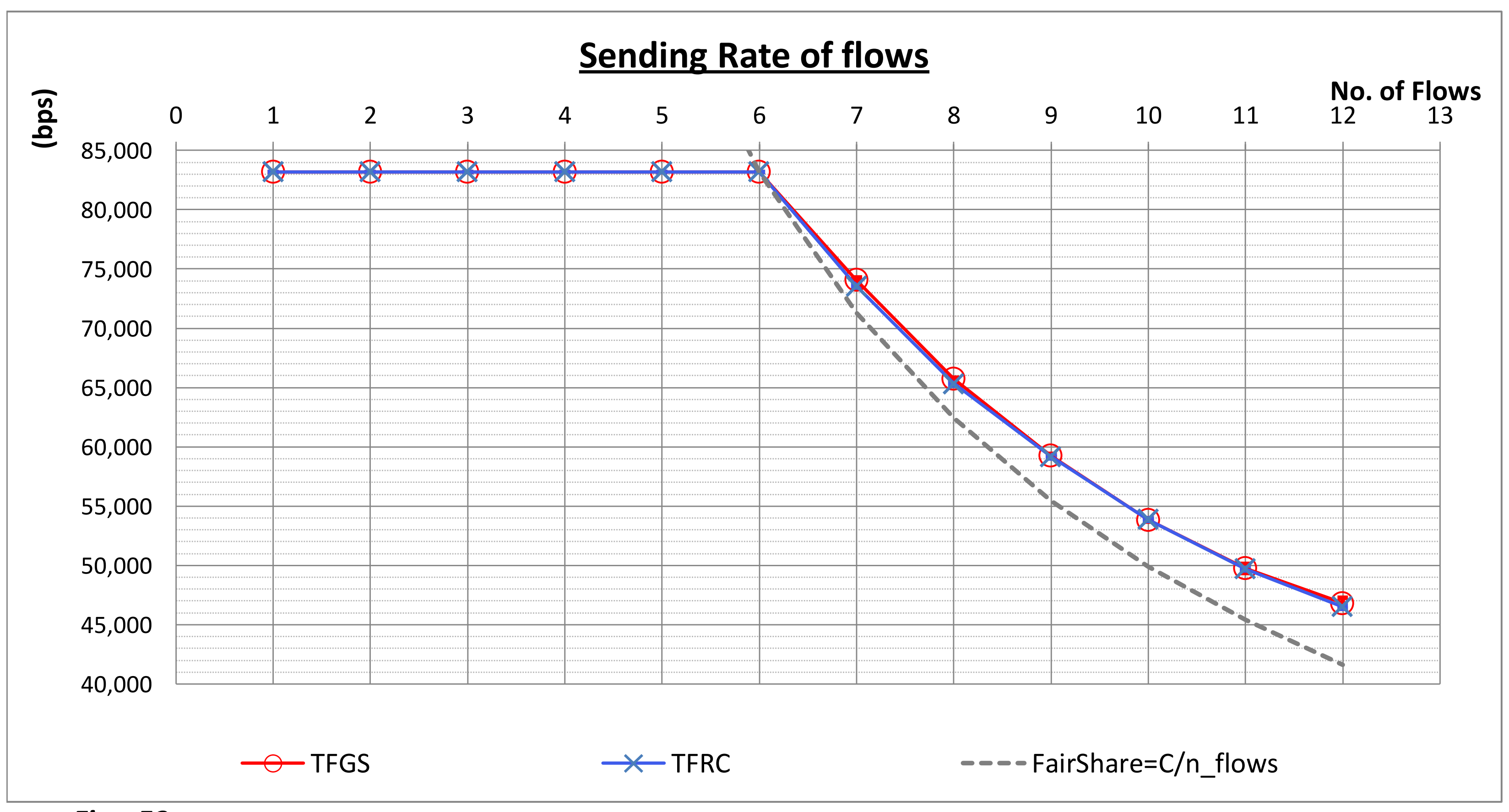


***Fig.* 52**

woTwoSS -28 with Validation, Non-cong.xlsx

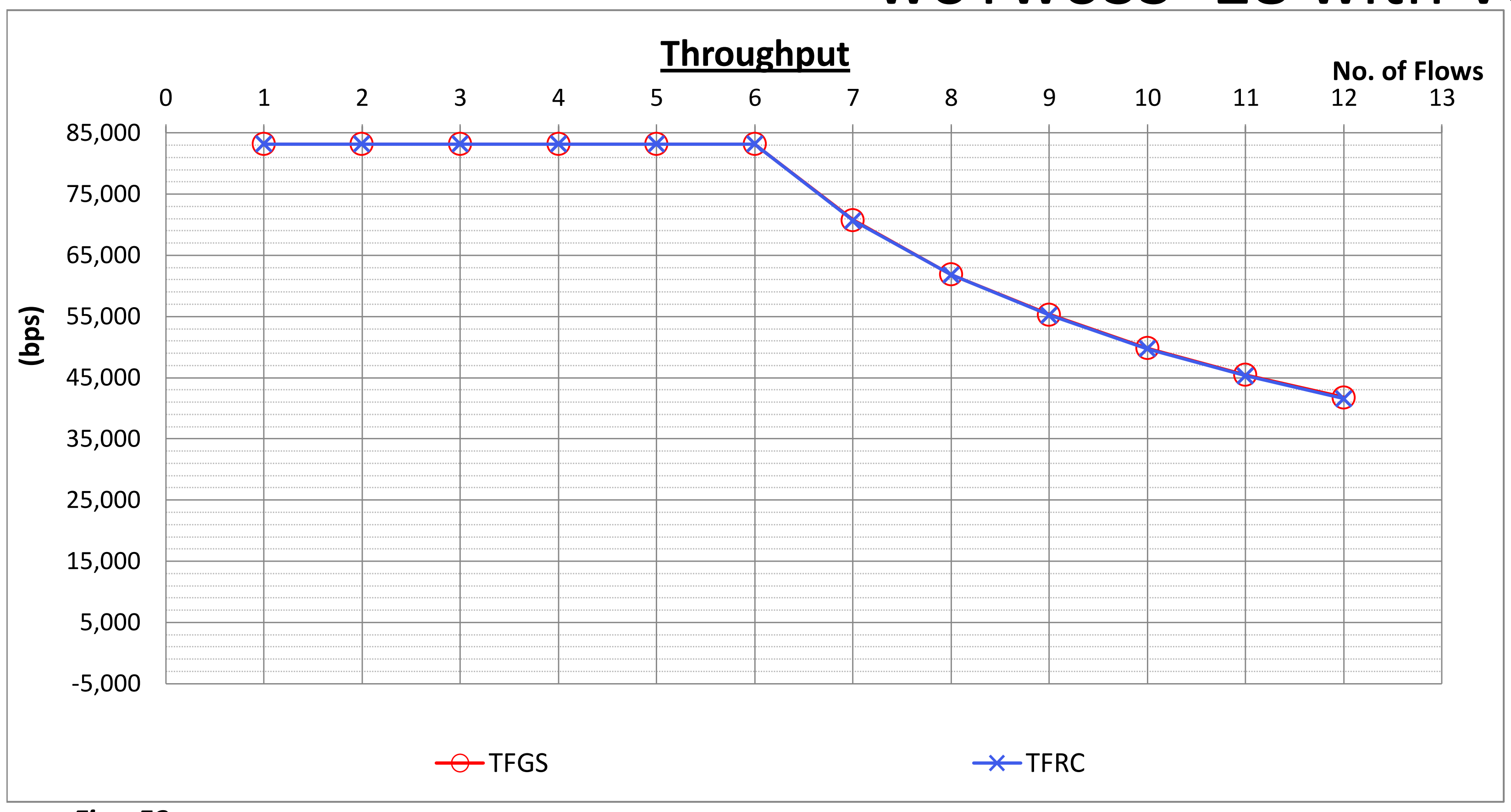


***Fig.* 53**

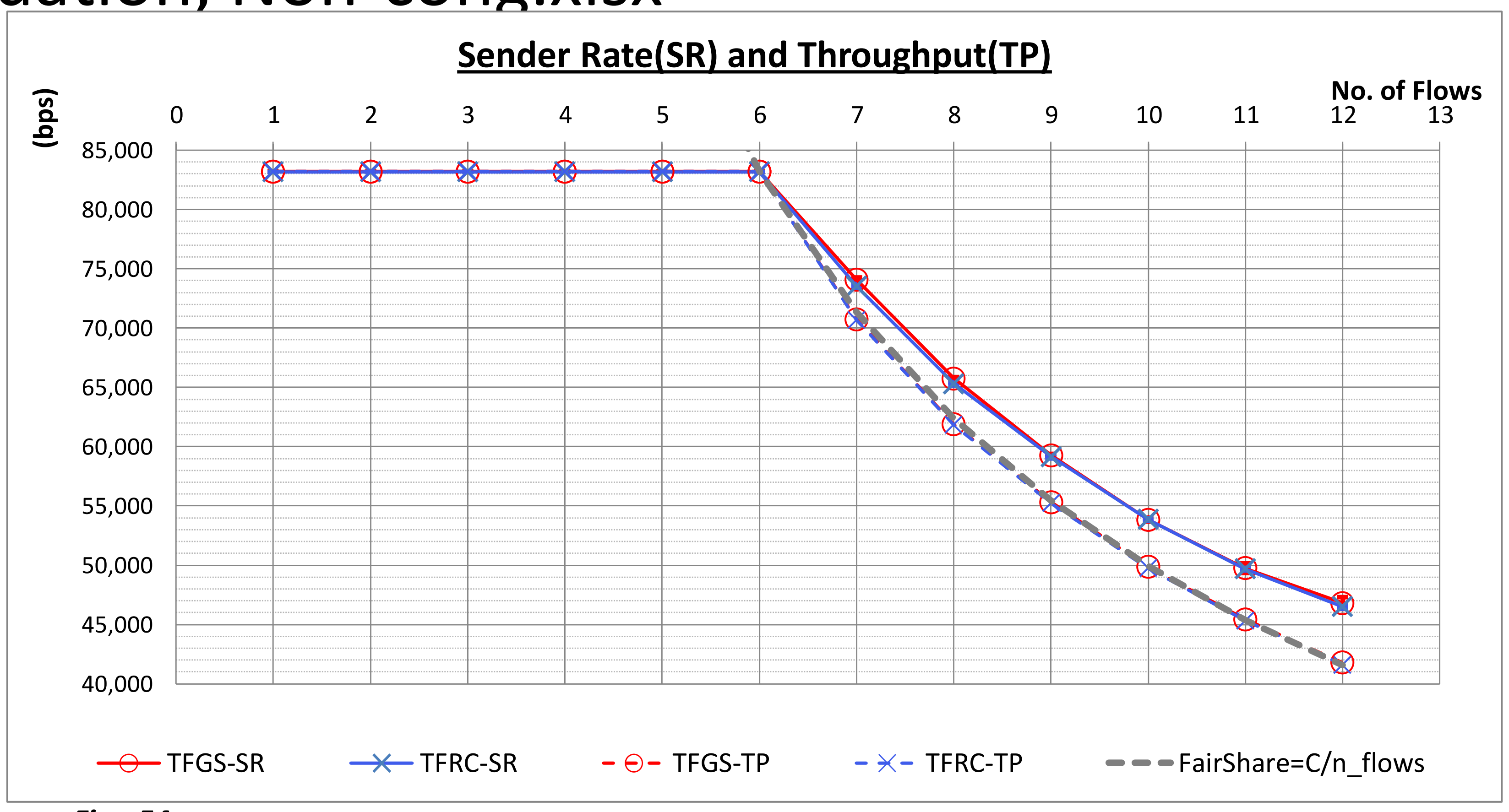


***Fig.* 54**

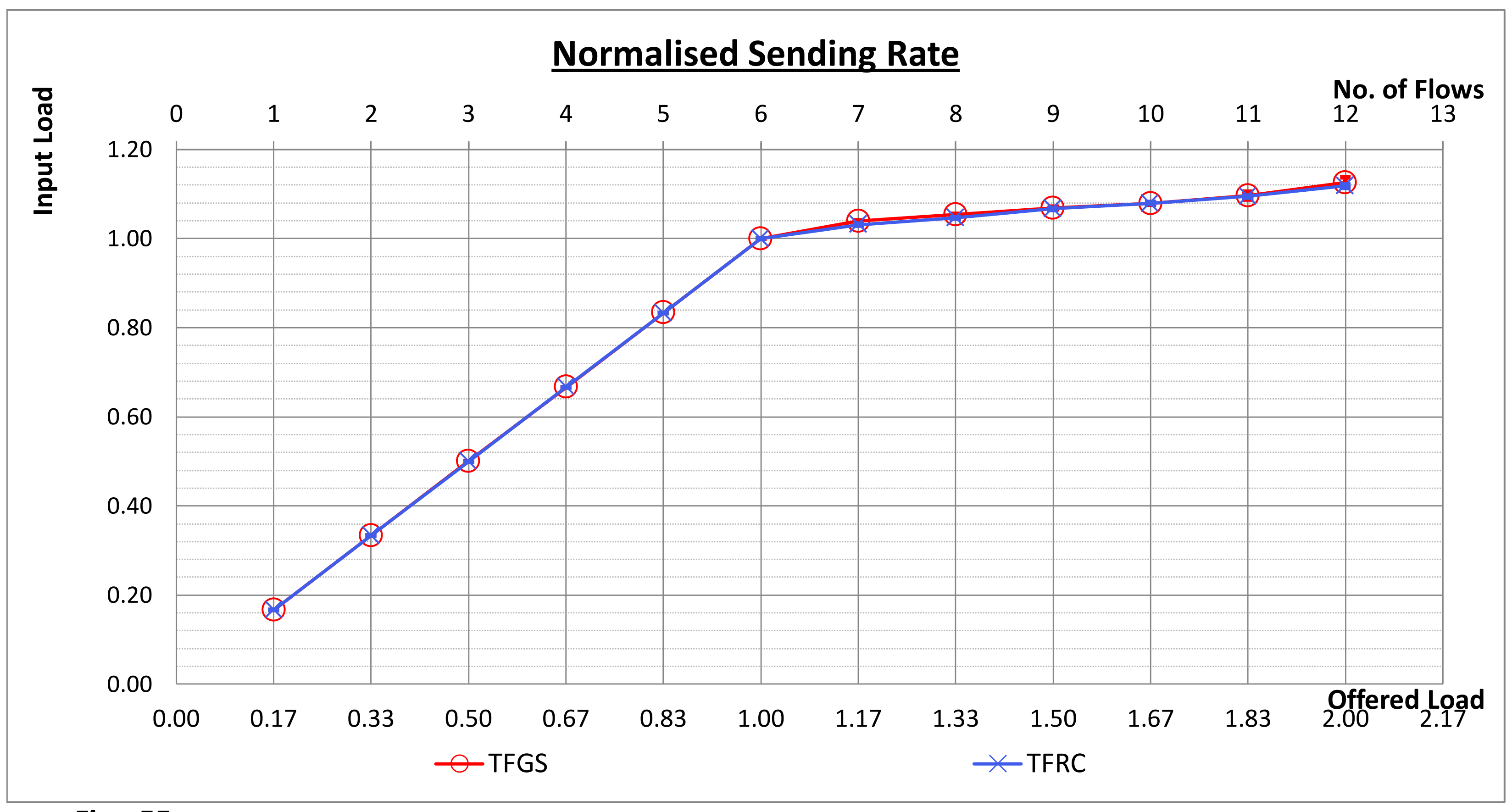


***Fig.* 55**

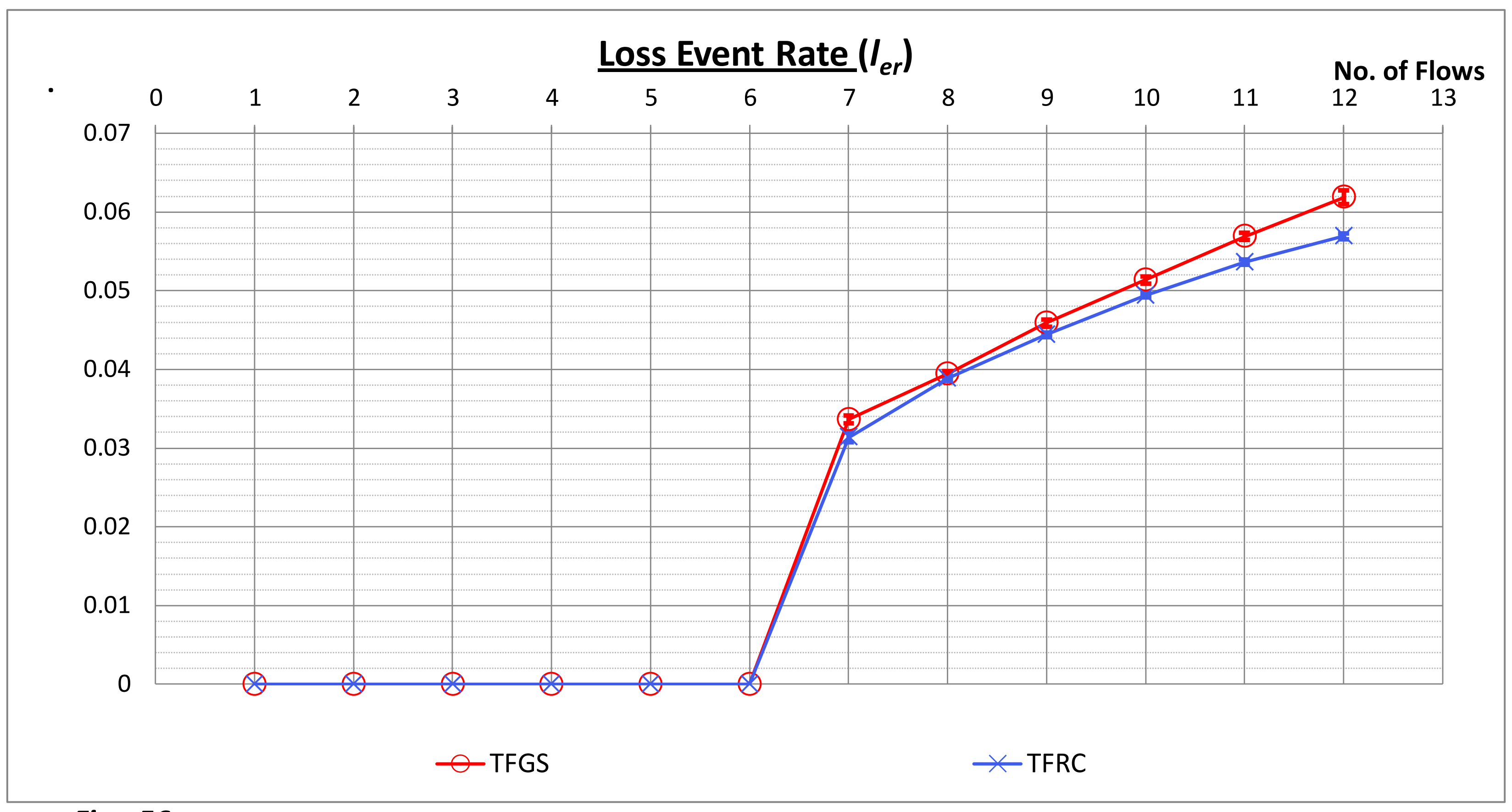


***Fig.* 56**

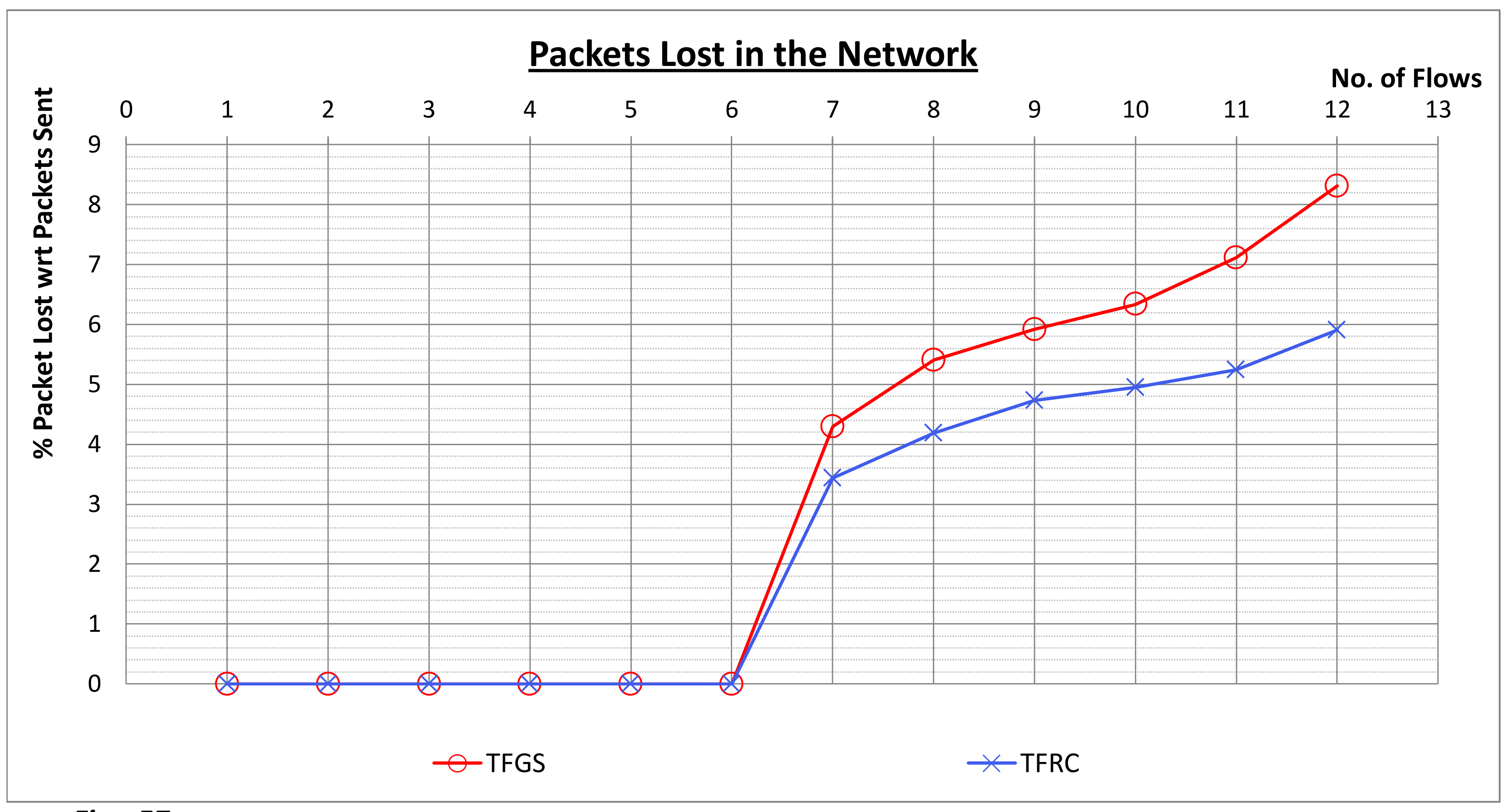


***Fig.* 57**

10. Appendix II, Results: Heterogeneous Traffic Mix

This section shows the simulation results for a heterogeneous traffic mix, giving a comparison of TFRC and TFGS CCM across various congestion levels. The network scenario description is defined in Table 5.4.

Simulation Study 2 - Heterogeneous Traffic Scenario							
Simulation Set	**Flow (Application) Type**	**Codec Type**	**Duration (secs)**	**CCM**	**No. of Flows**	**Capacity, C, (bps)**	**RED Queue Parameters (Min / Max) in packets**
A	Voice	MPEG-2	60	TFRC	X	998,400	40 / 120
	FTP	n/a	60	TCP	X	998,400	40 / 120
B	Voice	MPEG-2	60	TFGS	X	998,400	40 / 120
	FTP	n/a	60	TCP	X	998,400	40 / 120

from Table 5.4 *(Chapter 5)*

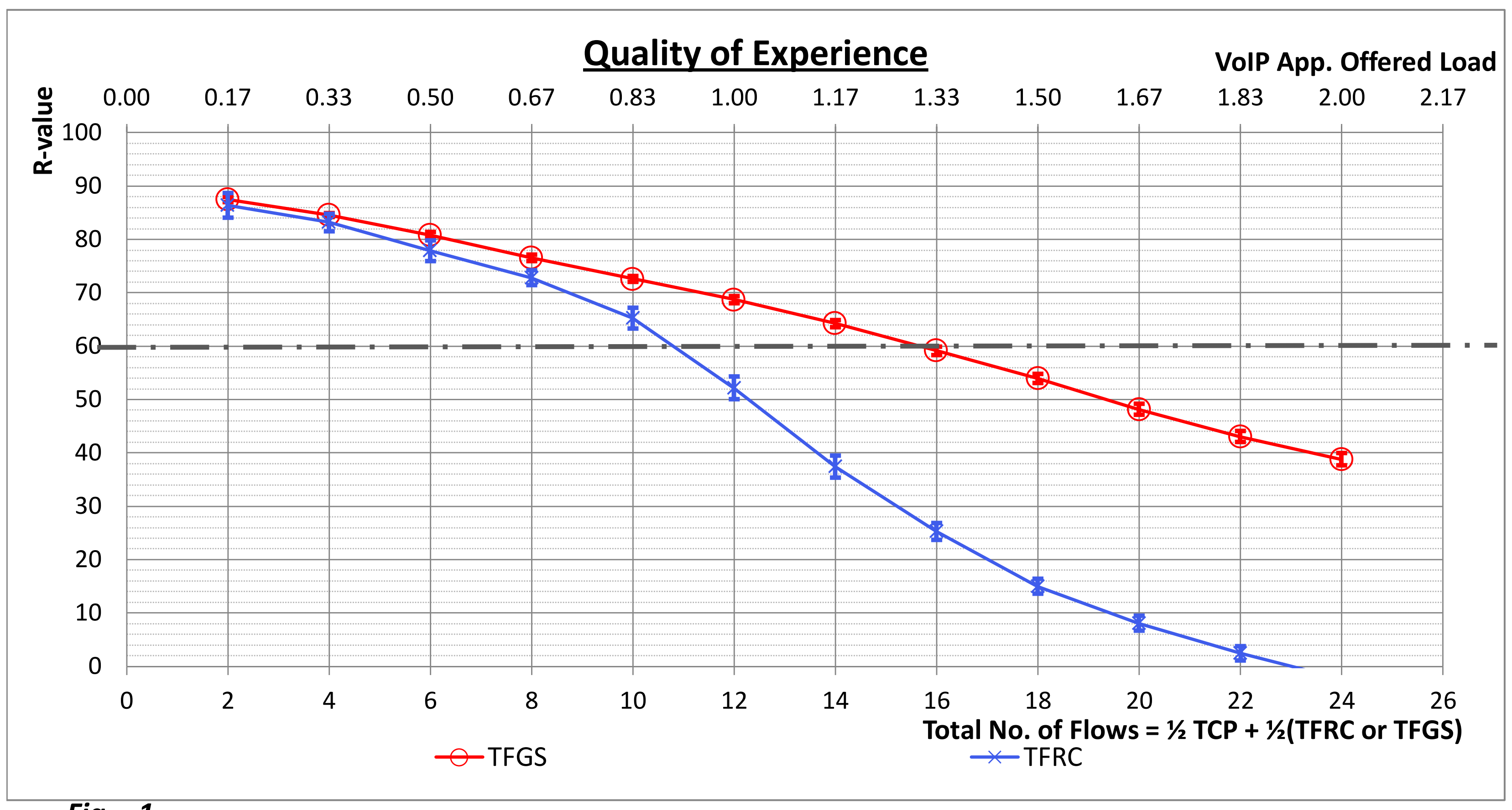


Fig. 1

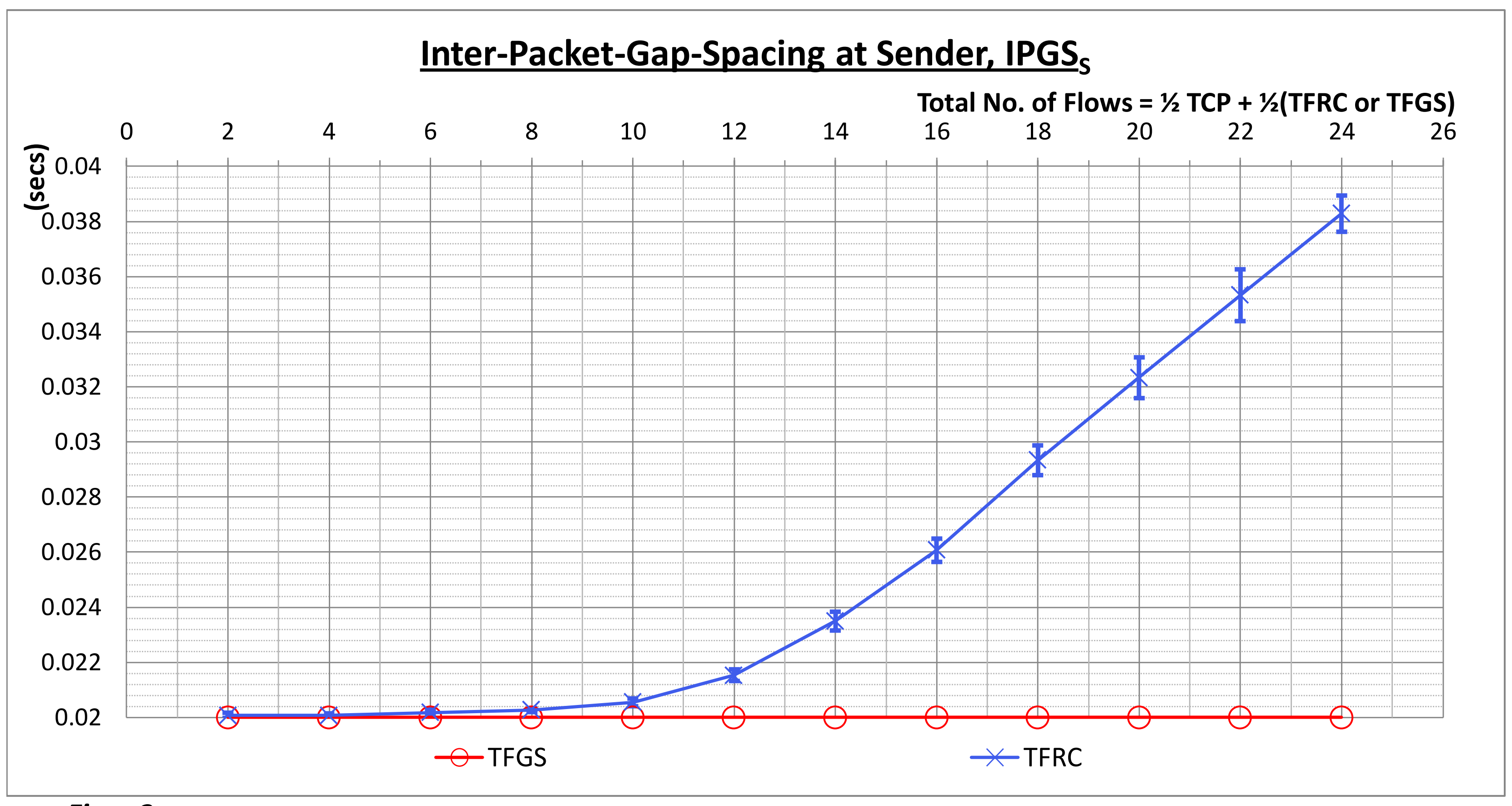


Fig. 2

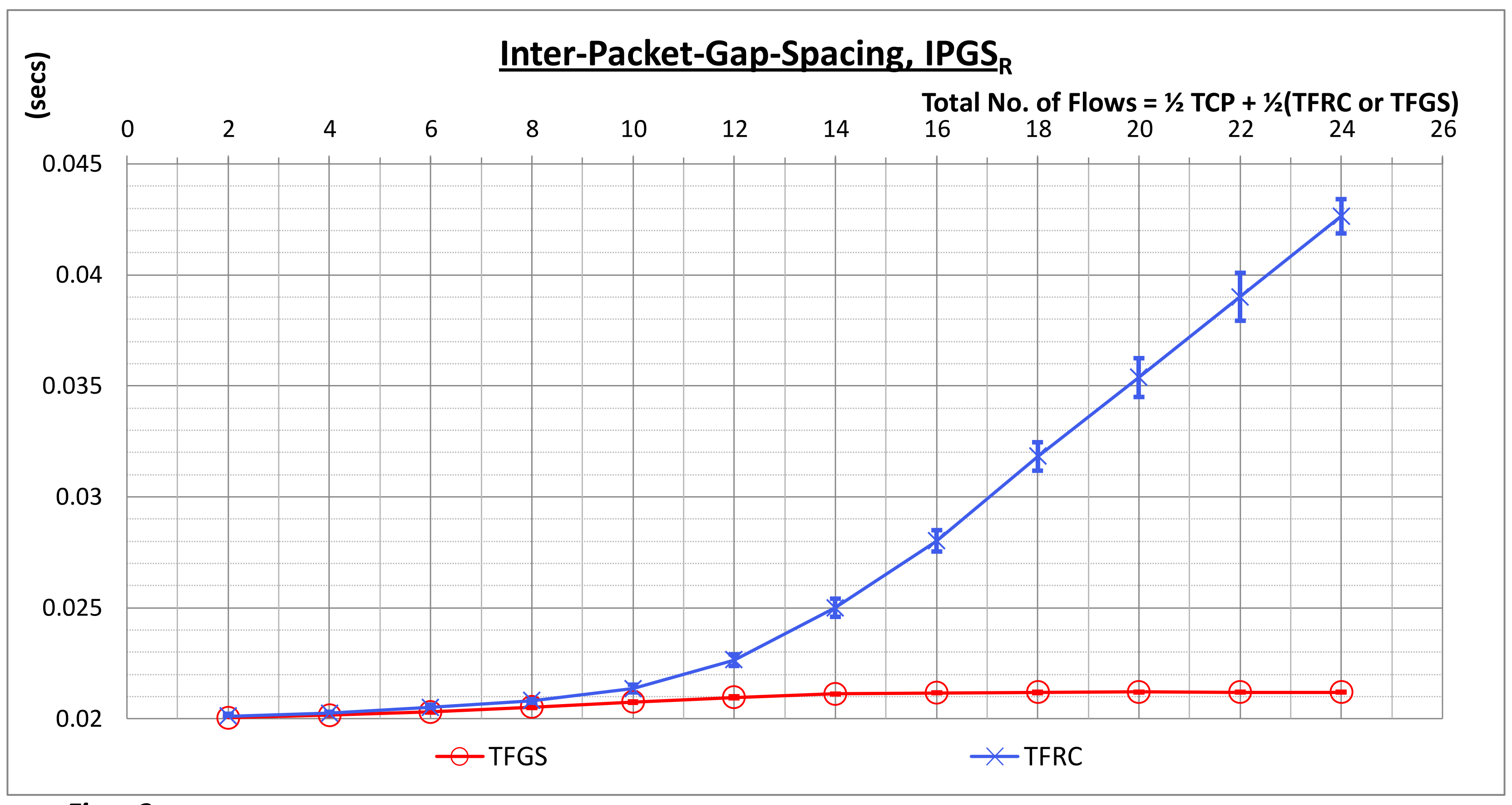


Fig. 3

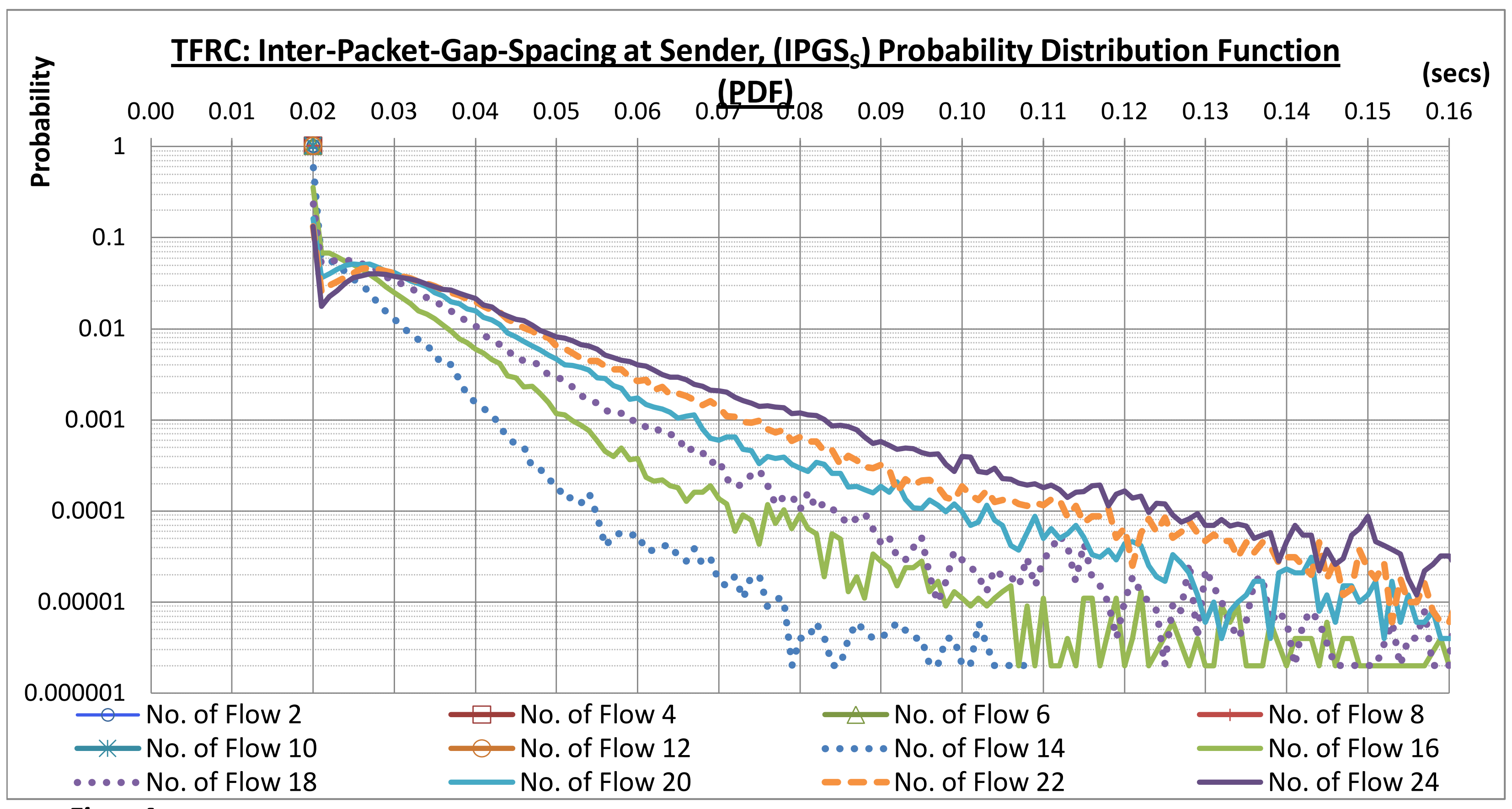


Fig. 4

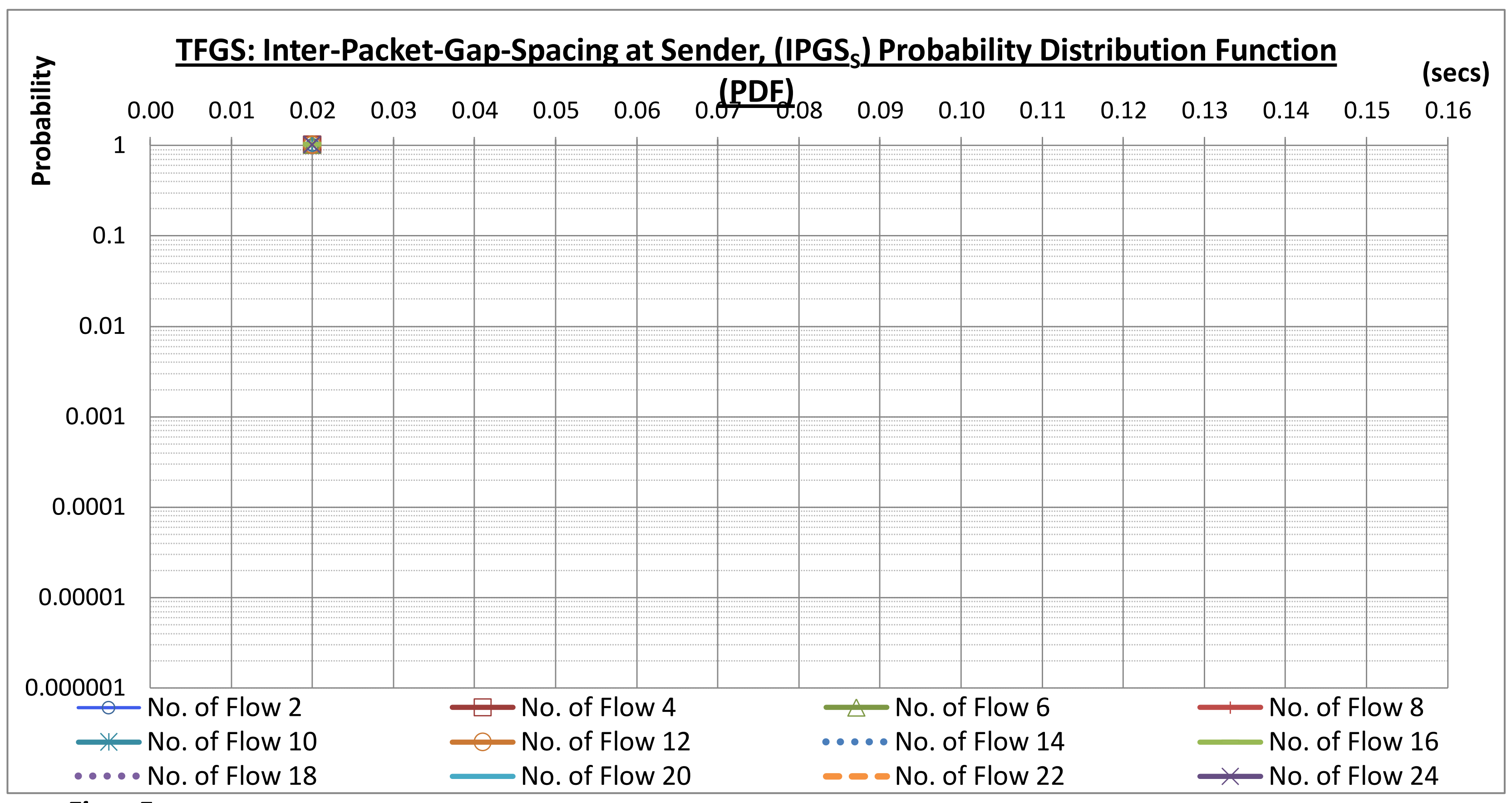


Fig. 5

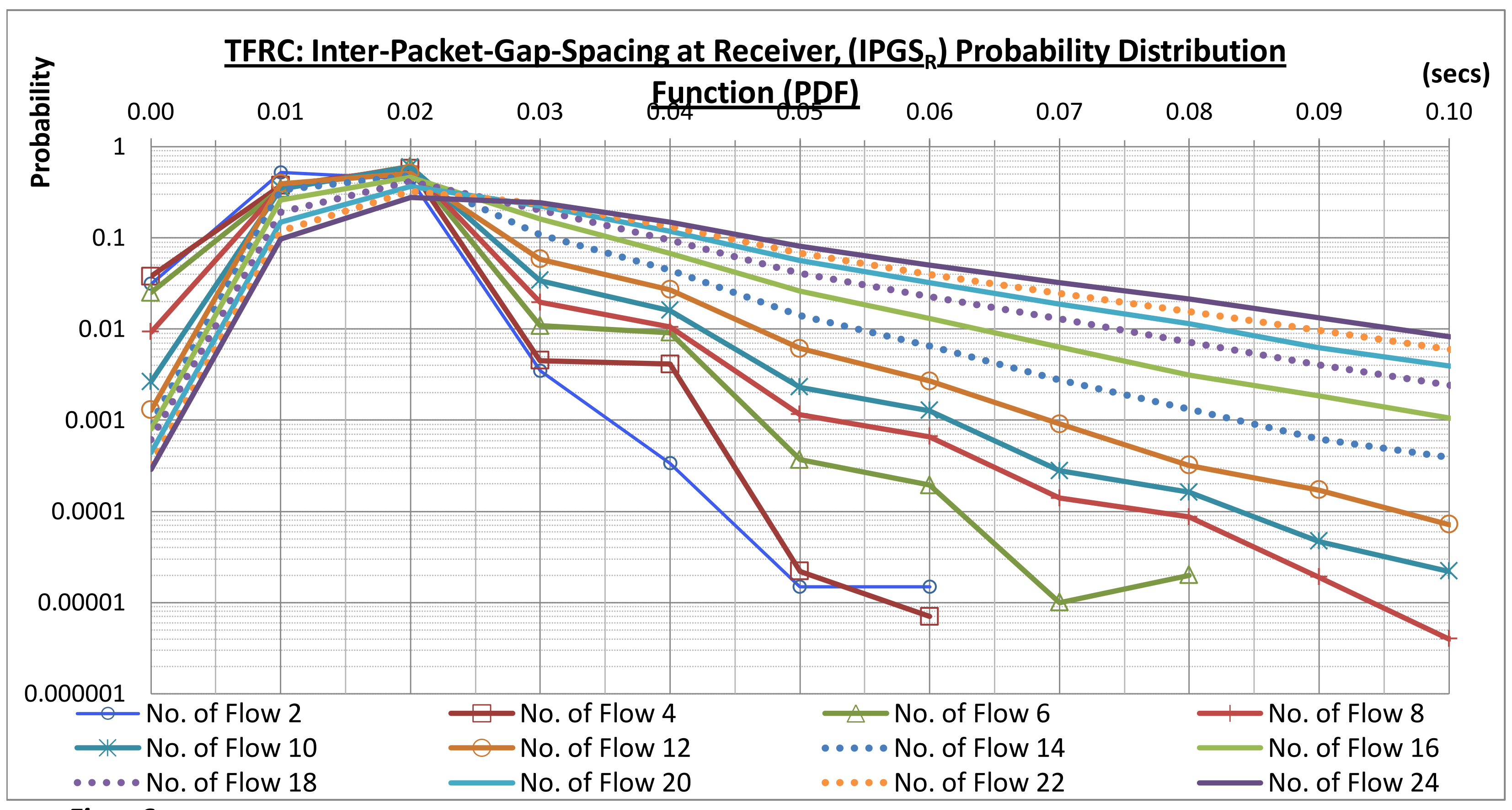


Fig. 6

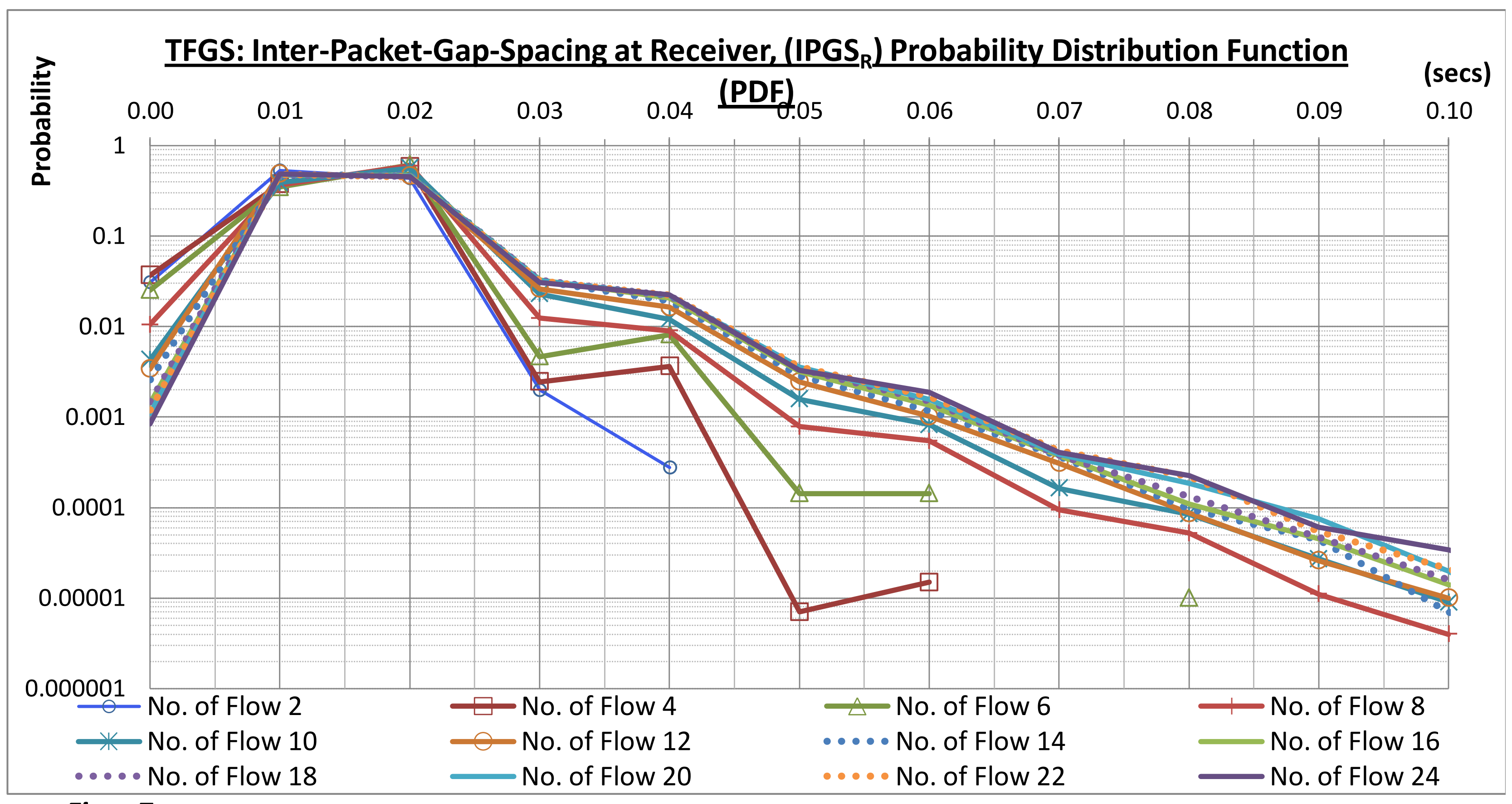


Fig. 7

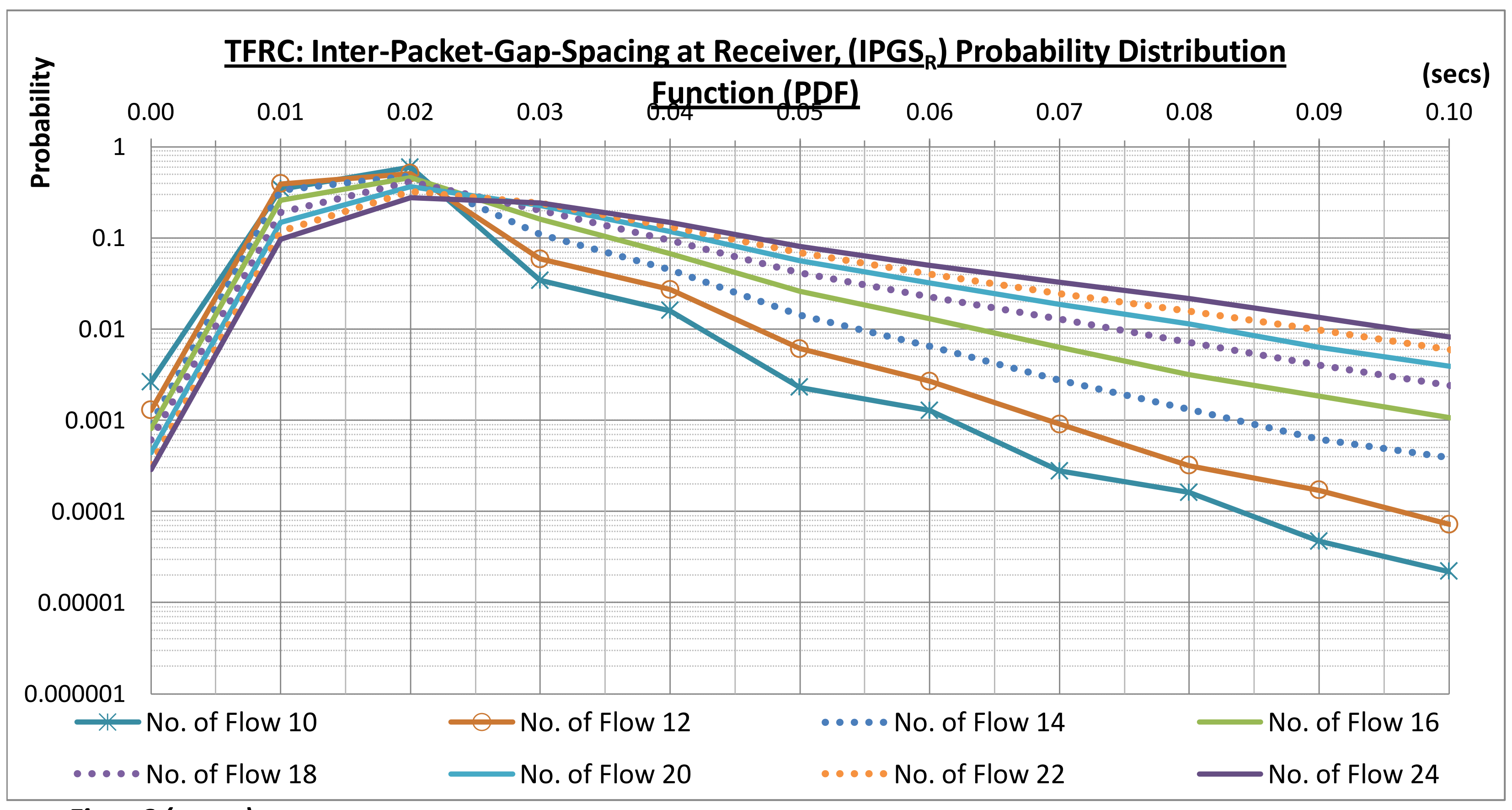


Fig. 6 (zoom)

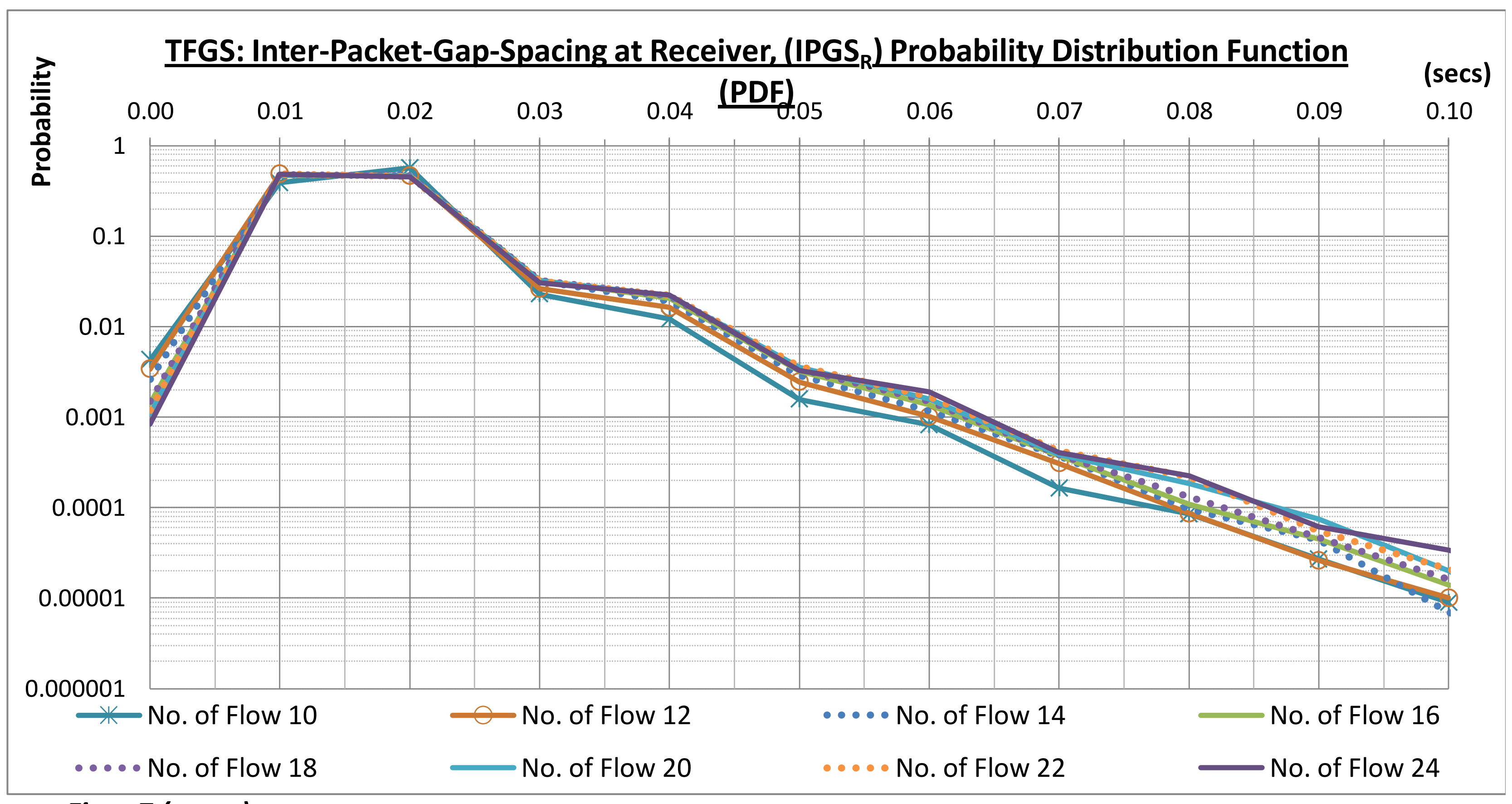


Fig. 7 (zoom)

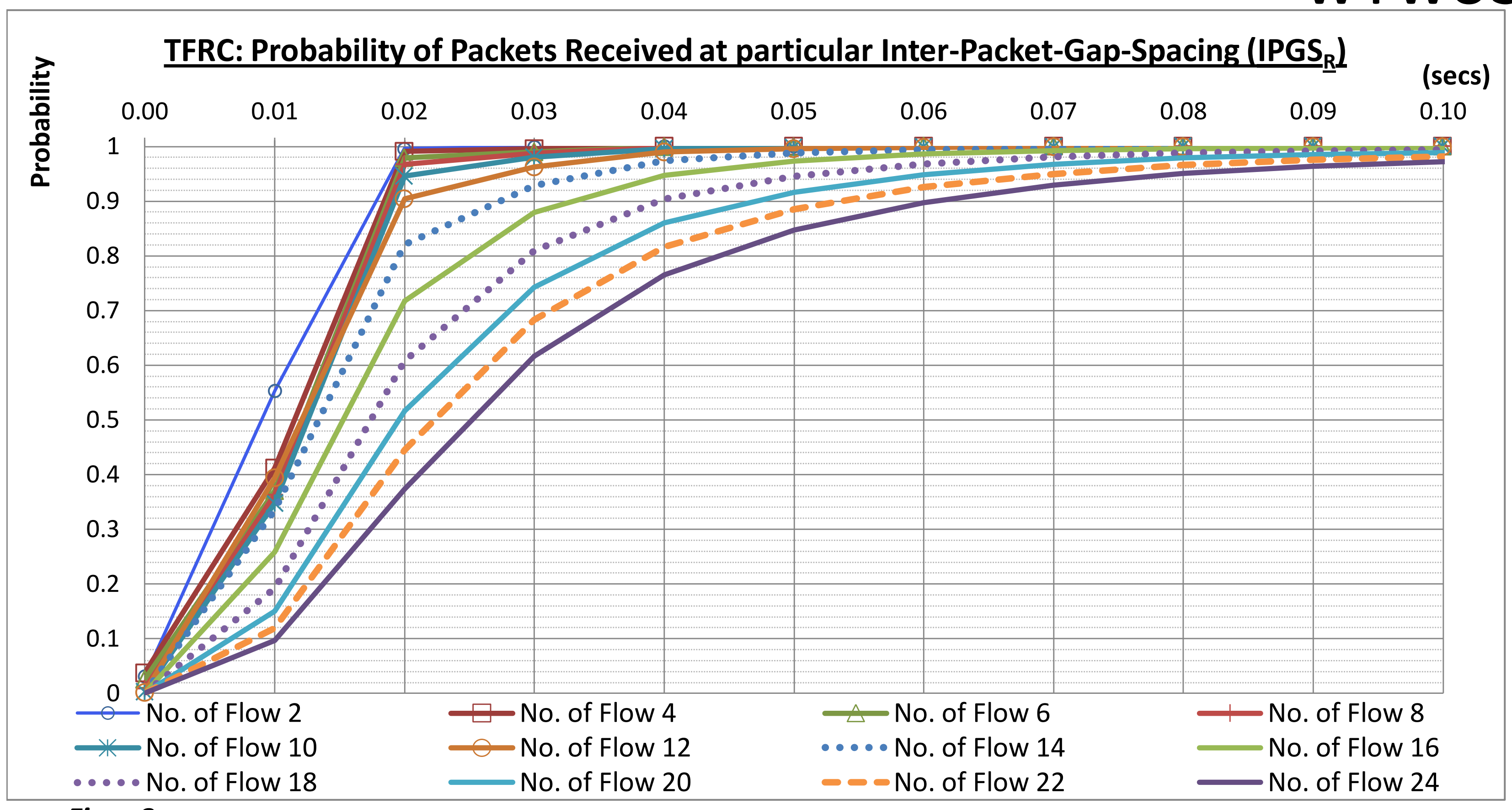


Fig. 8

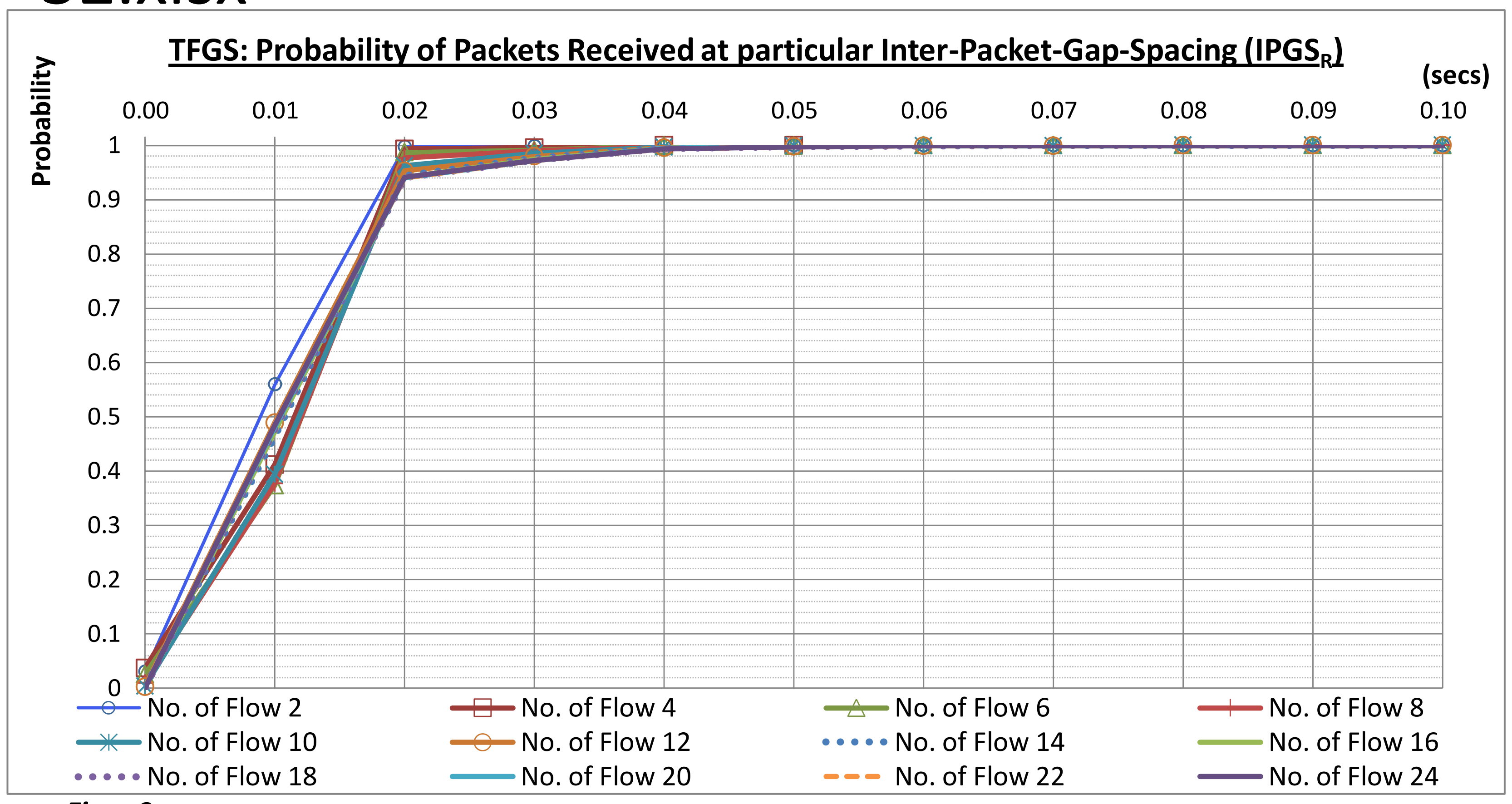


Fig. 9

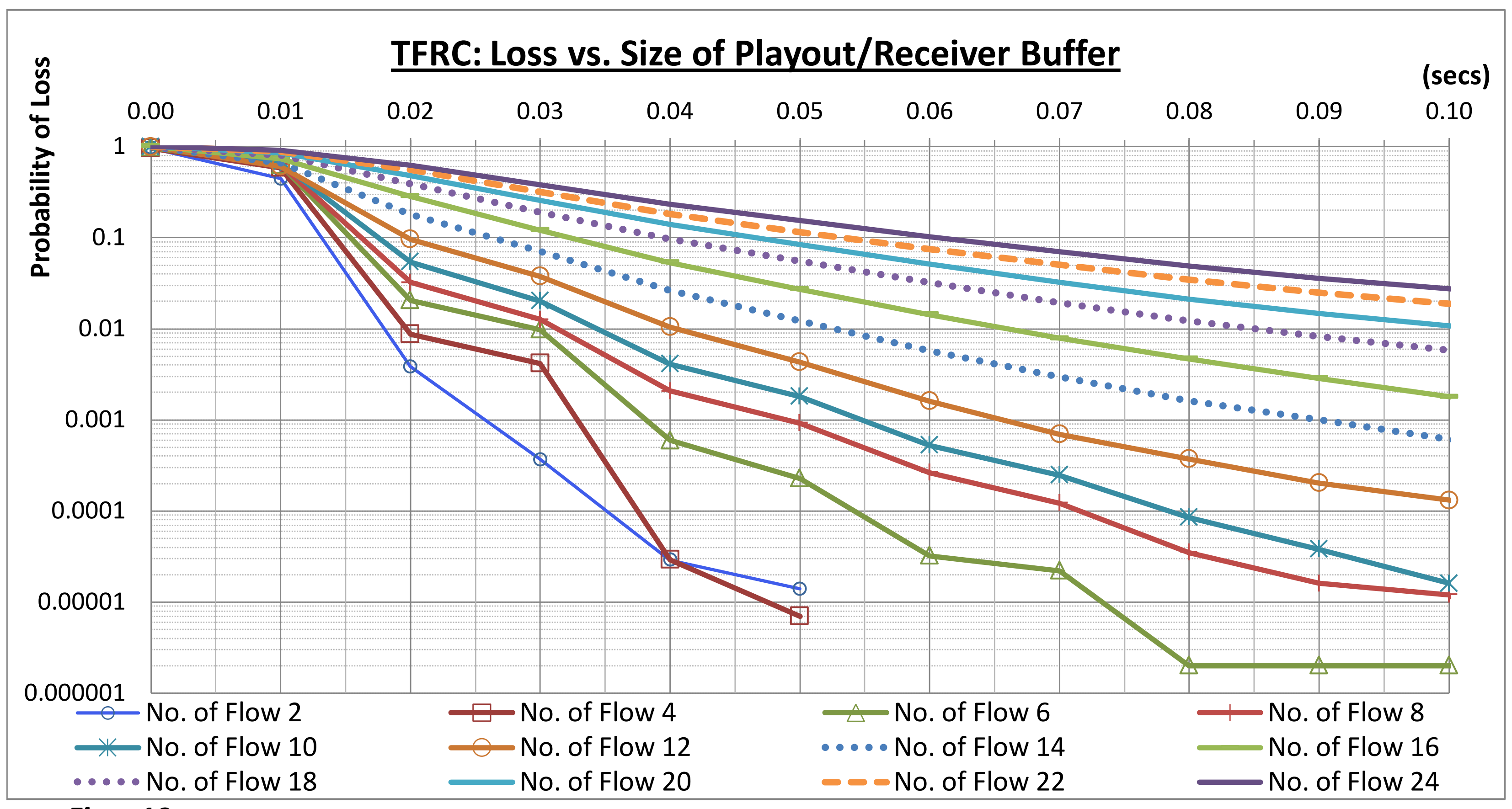


Fig. 10

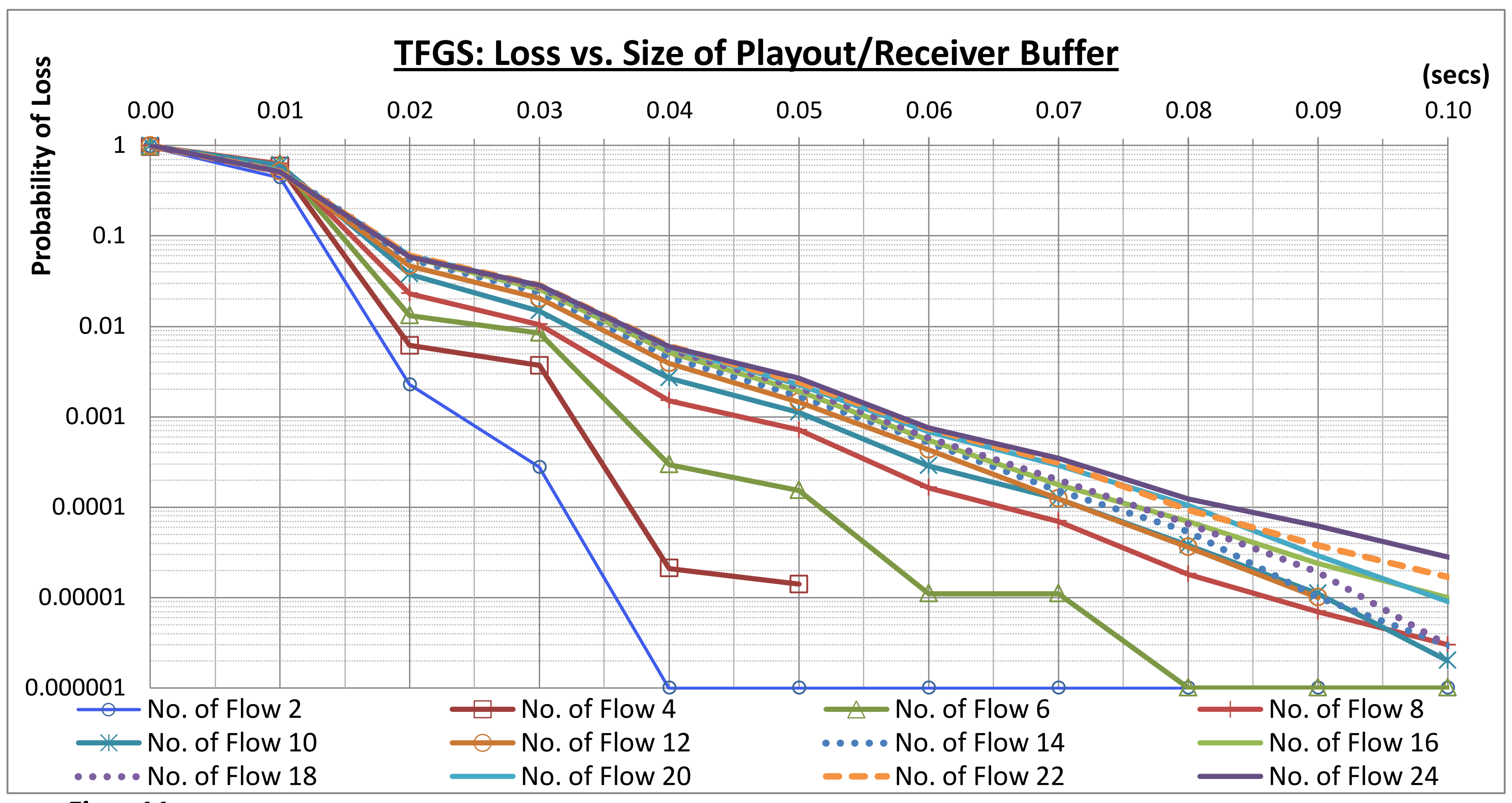


Fig. 11

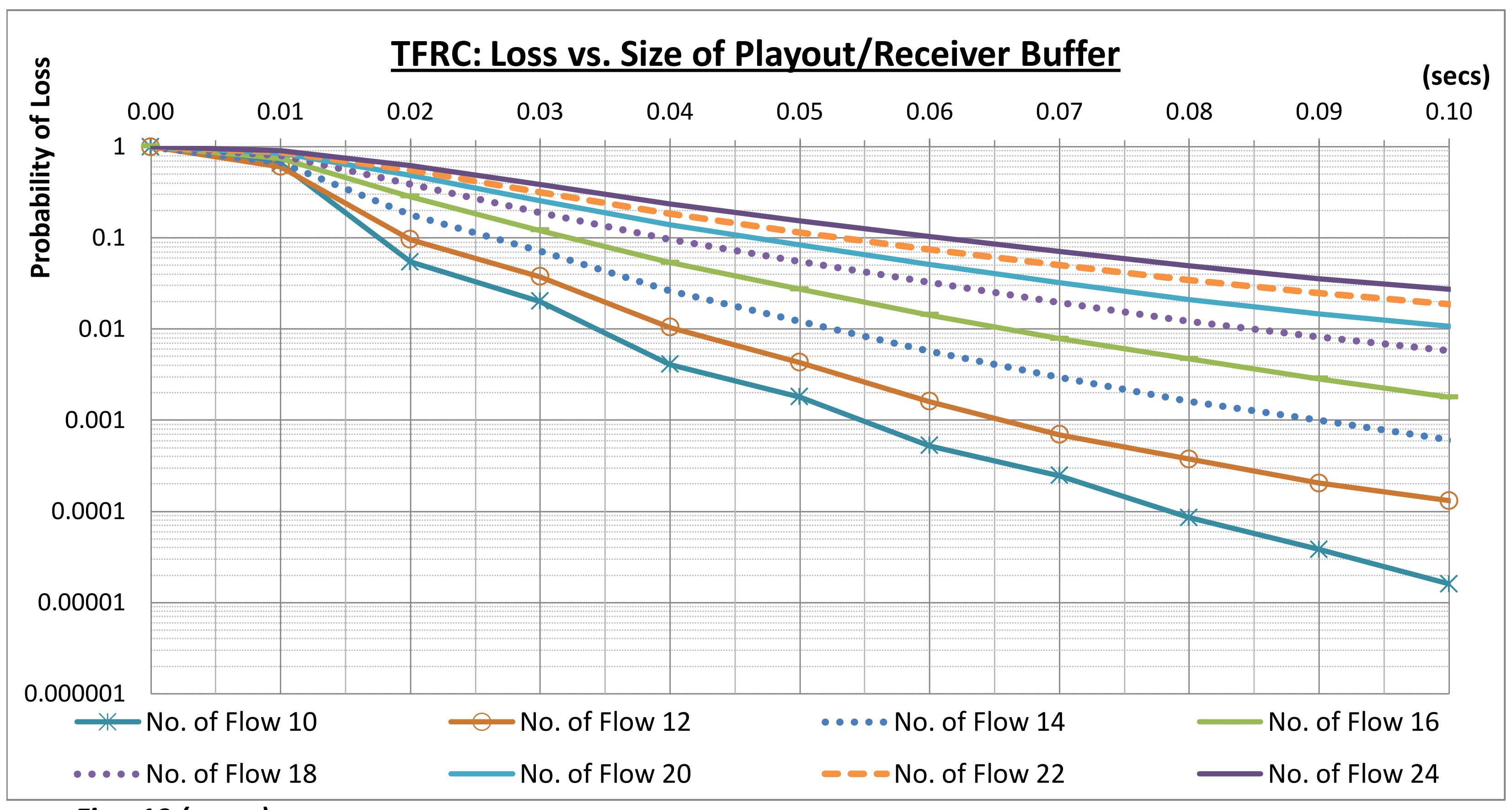


Fig. 10 (zoom)

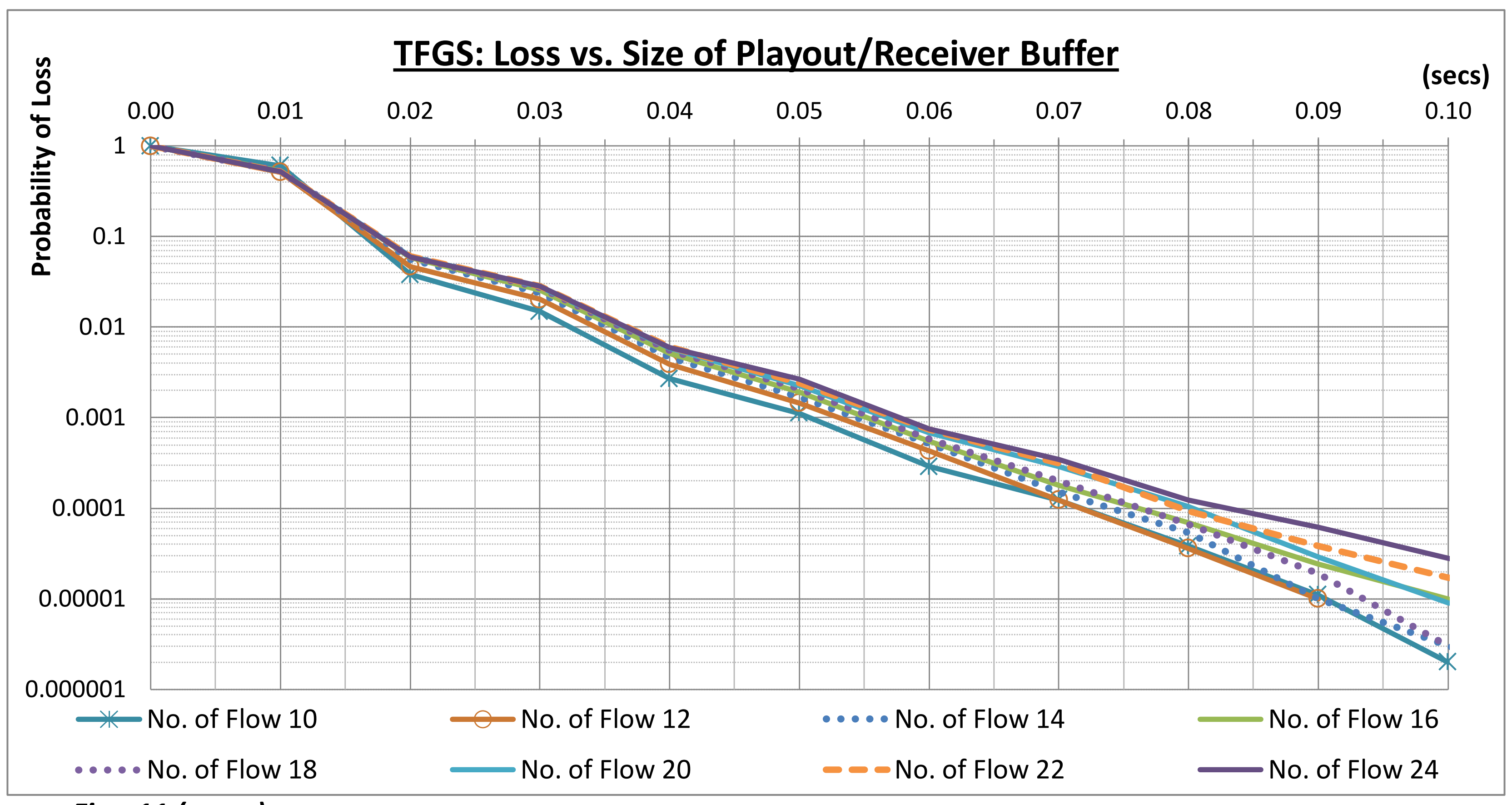


Fig. 11 (zoom)

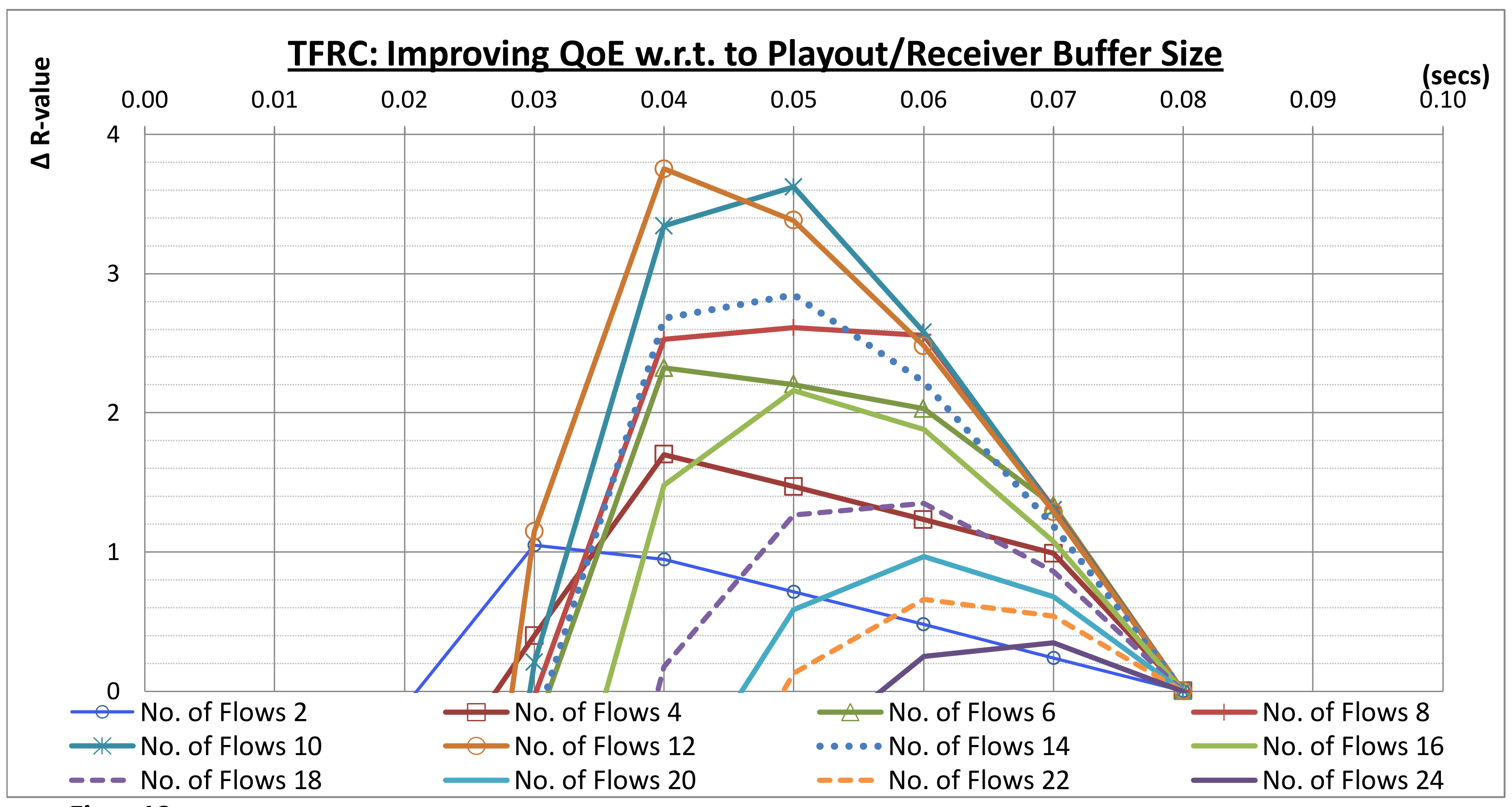


Fig. 12

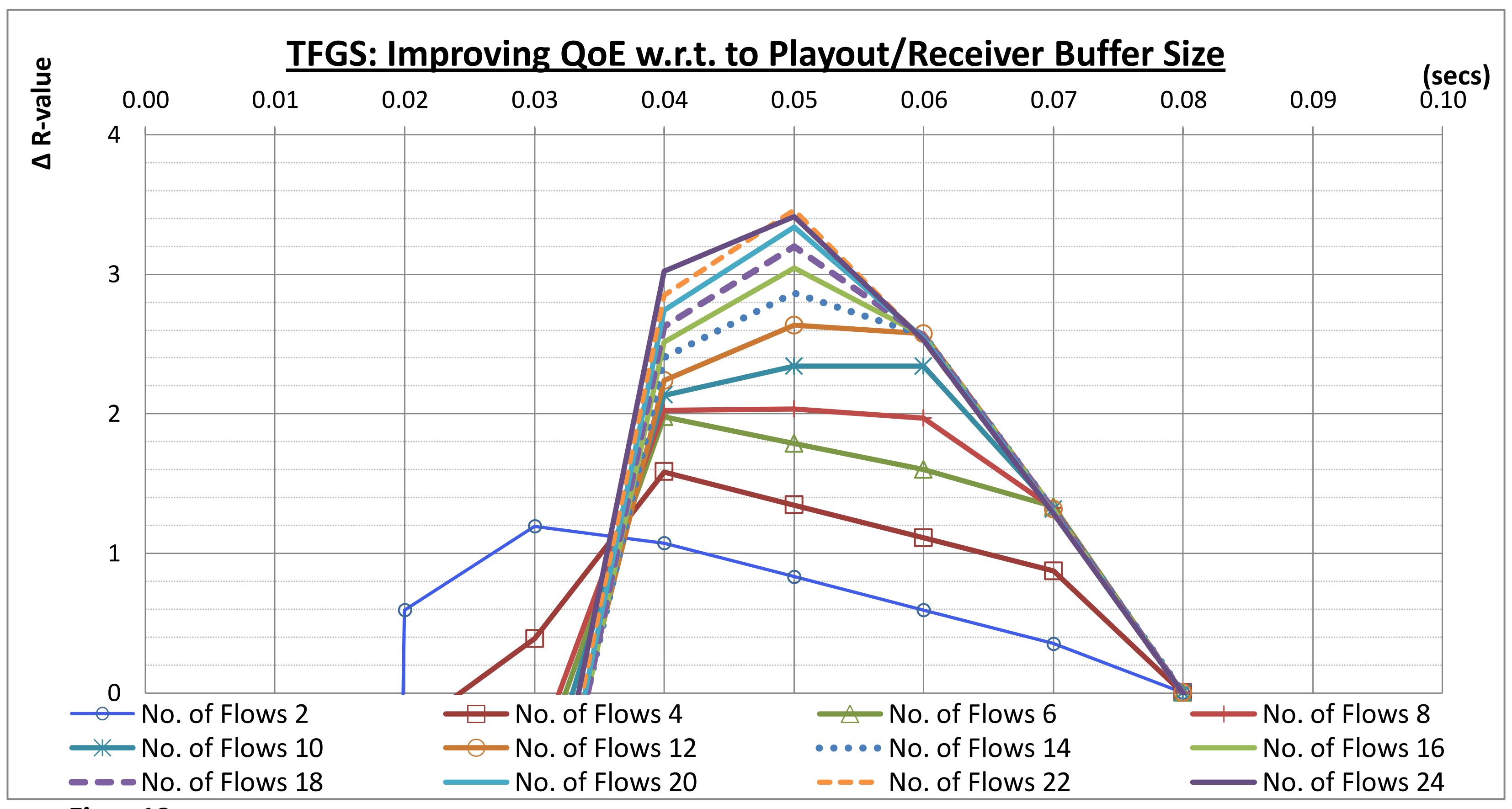


Fig. 13

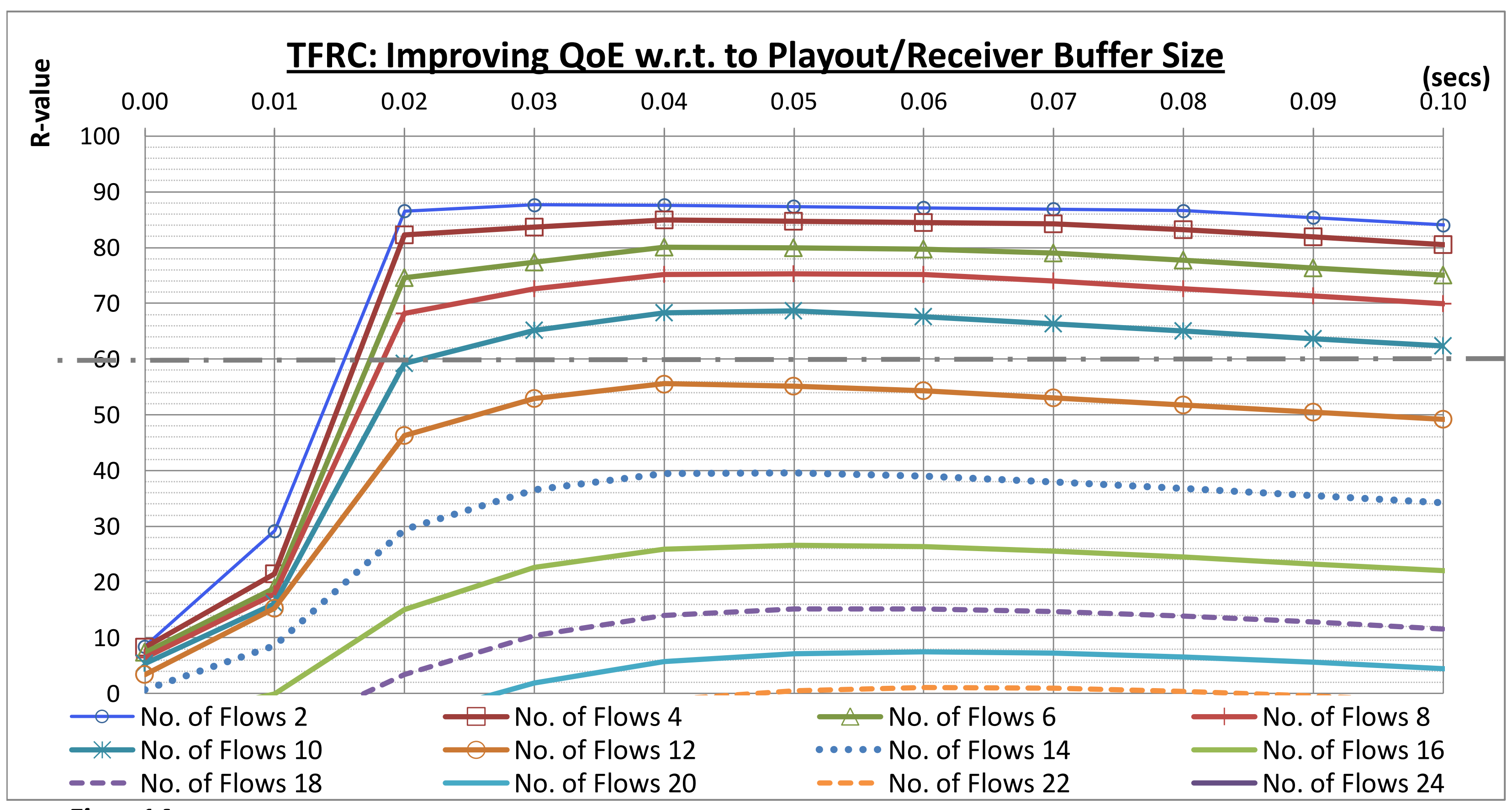


Fig. 14

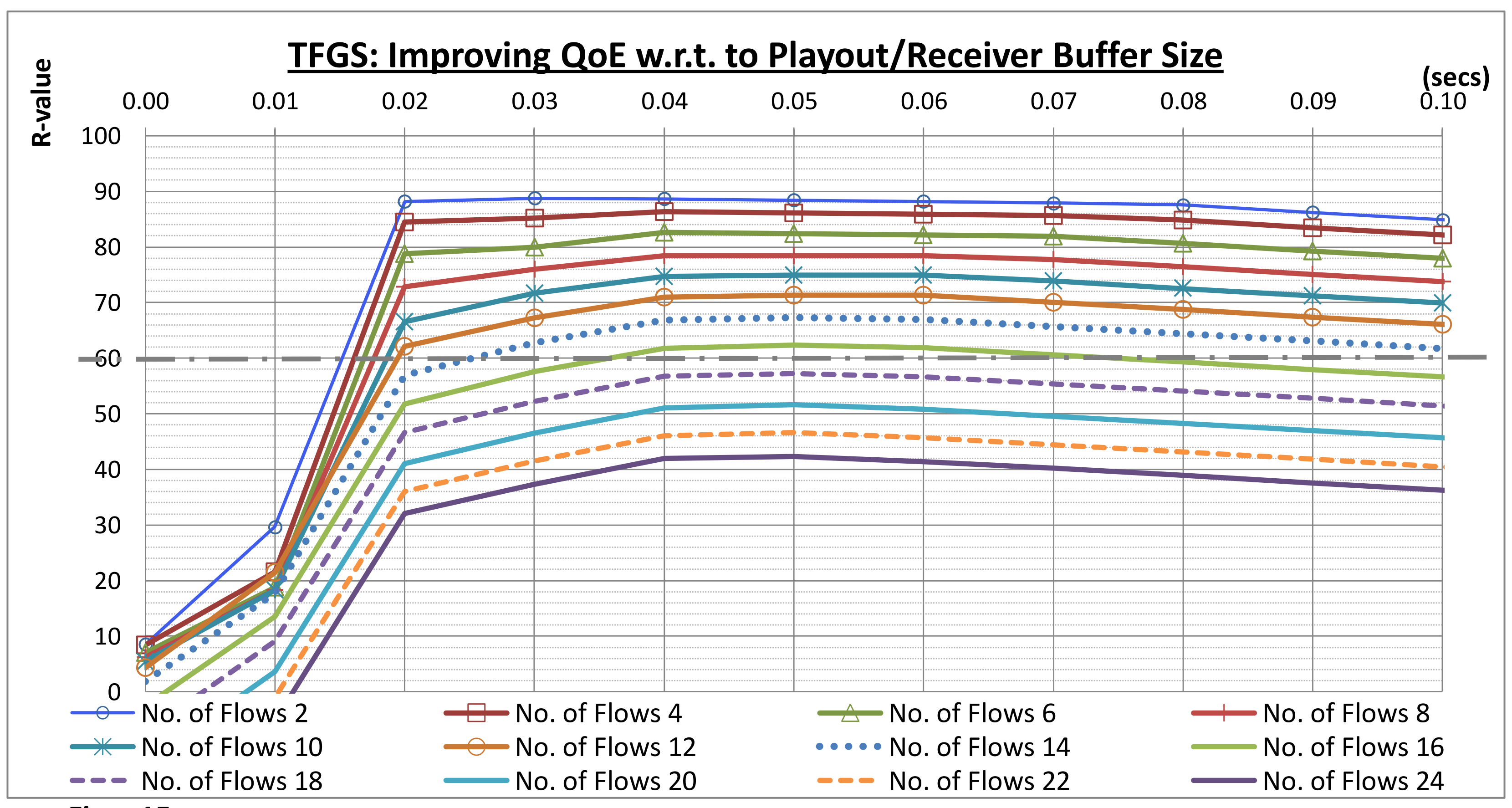


Fig. 15

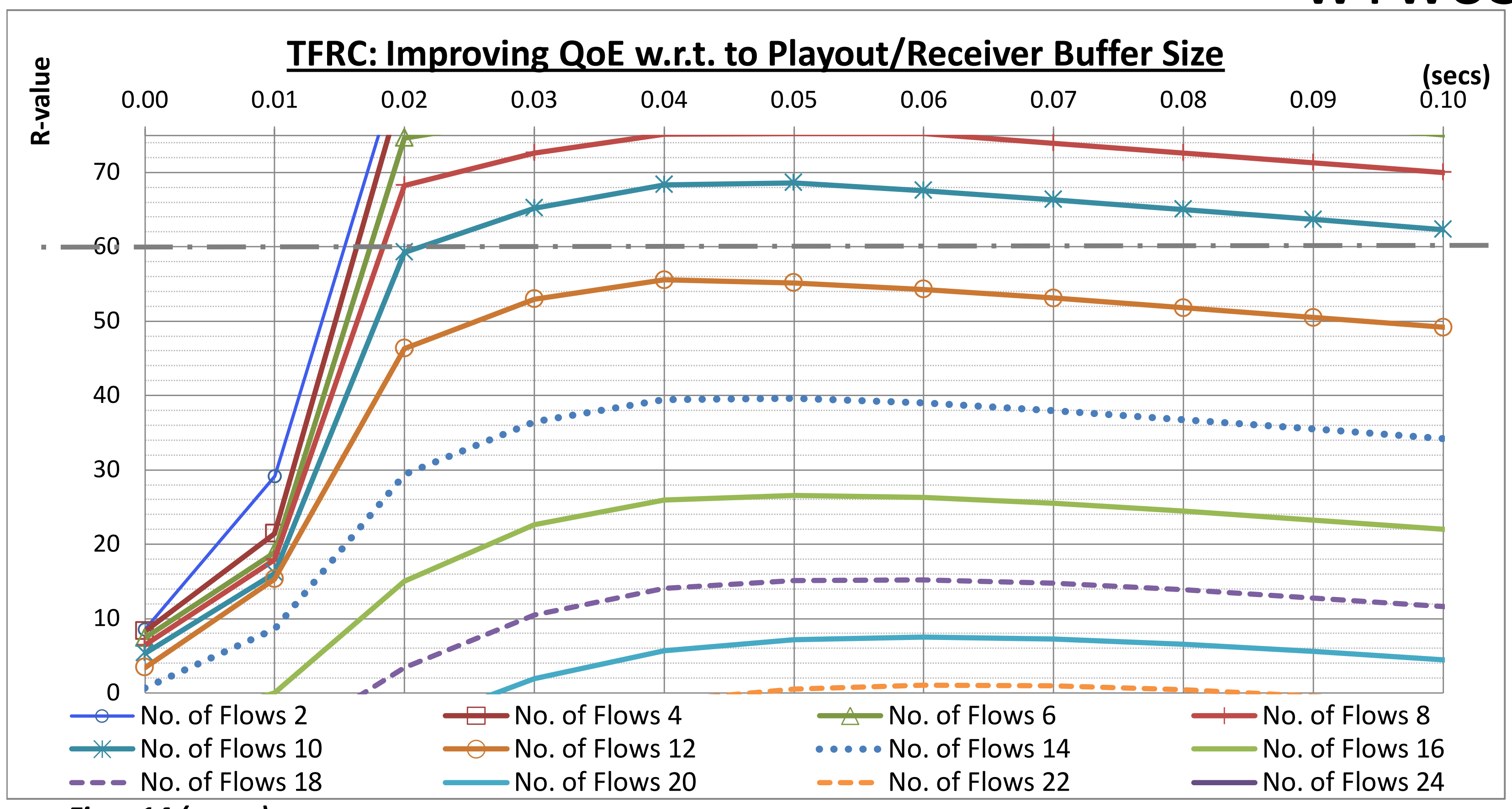


Fig. 14 (zoom)

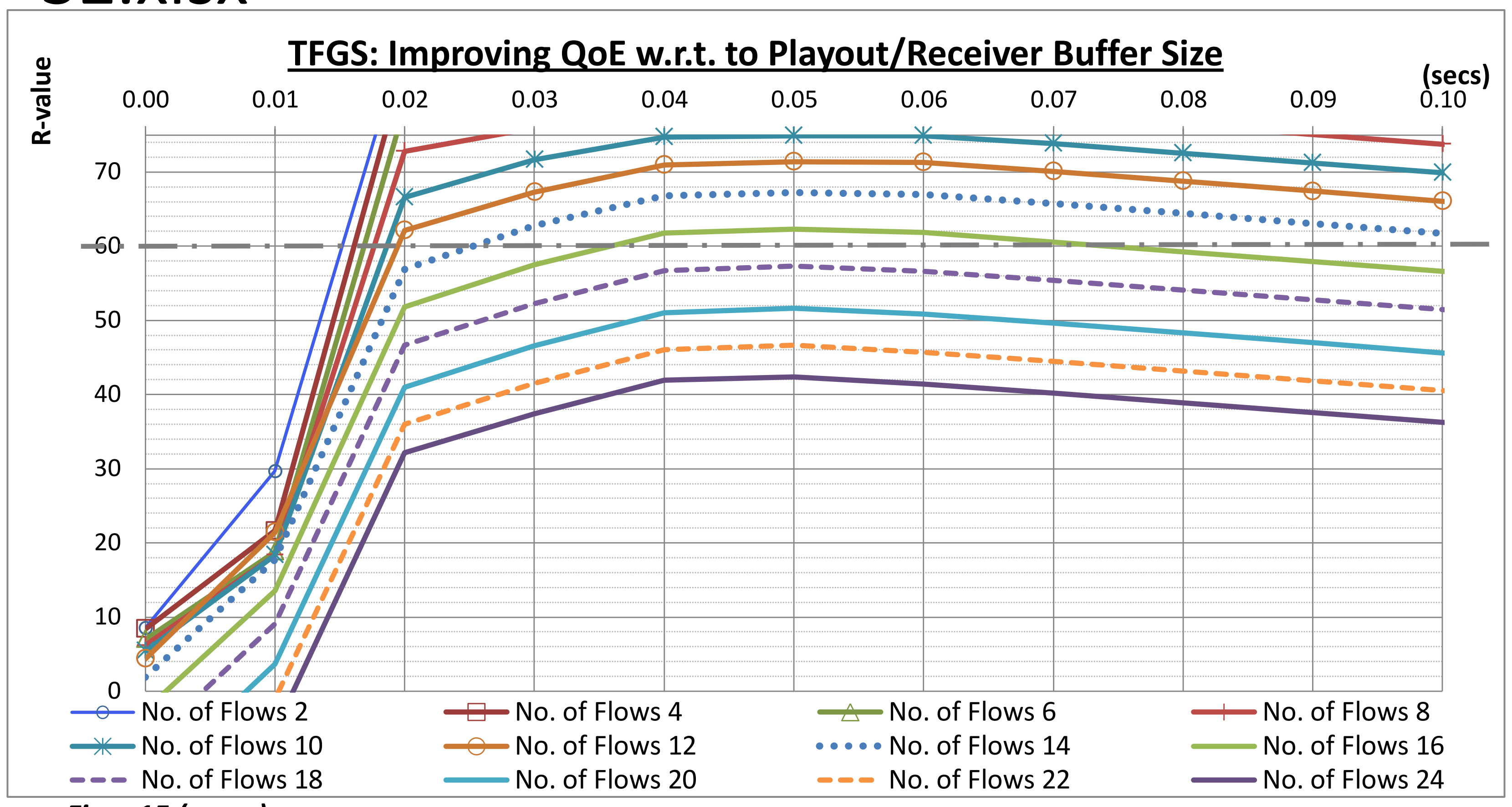


Fig. 15 (zoom)

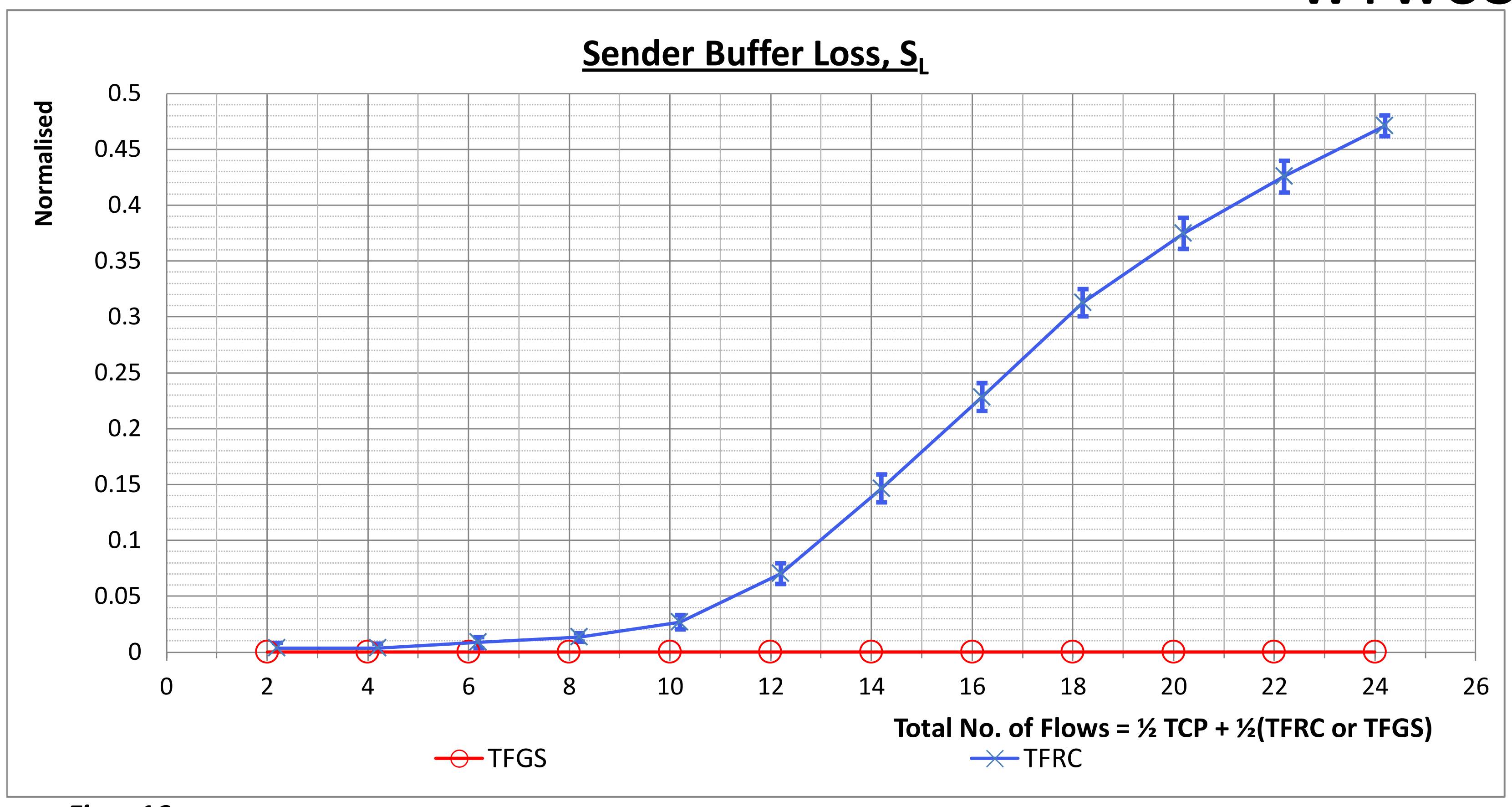


Fig. 16

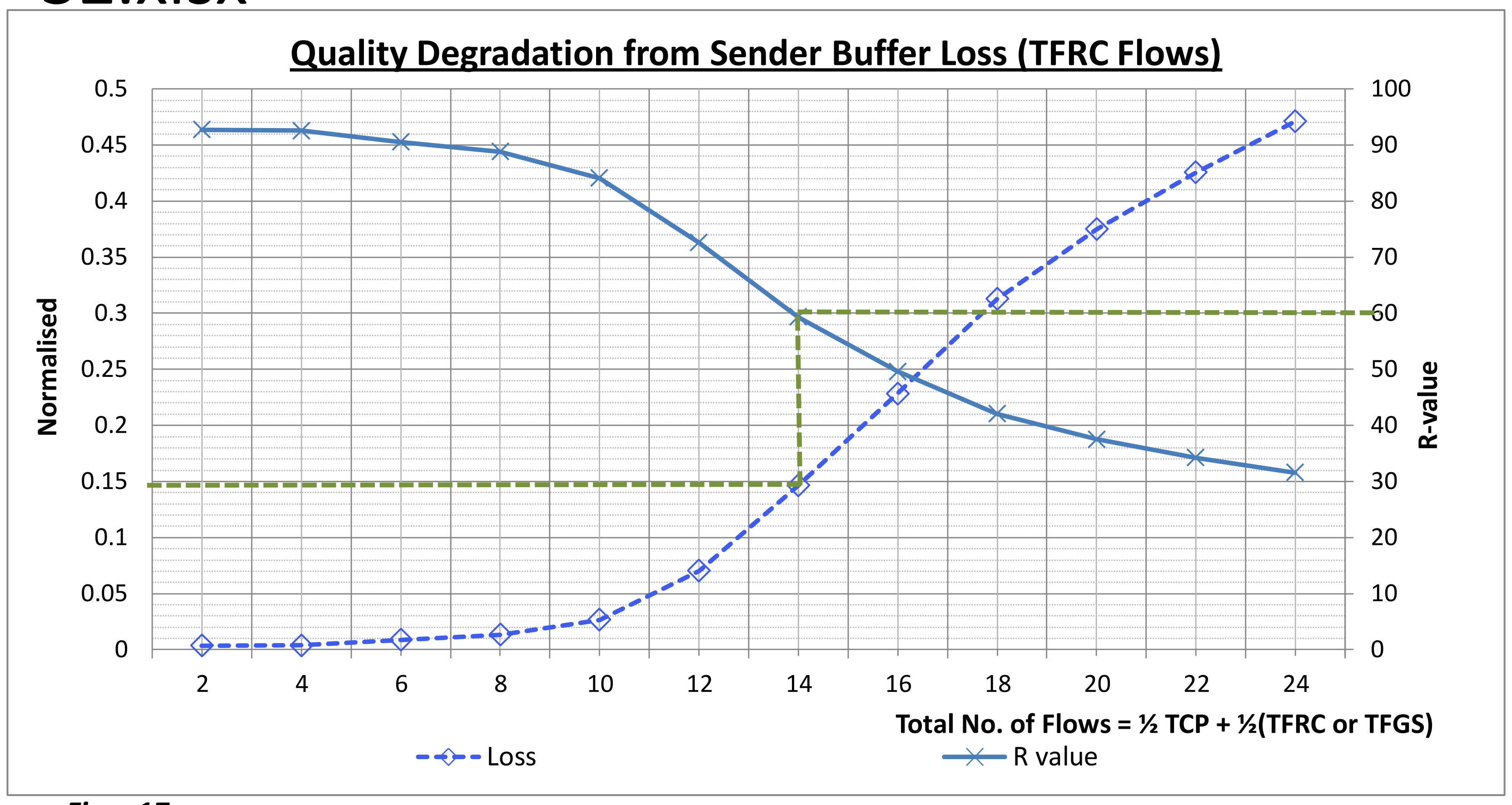


Fig. 17

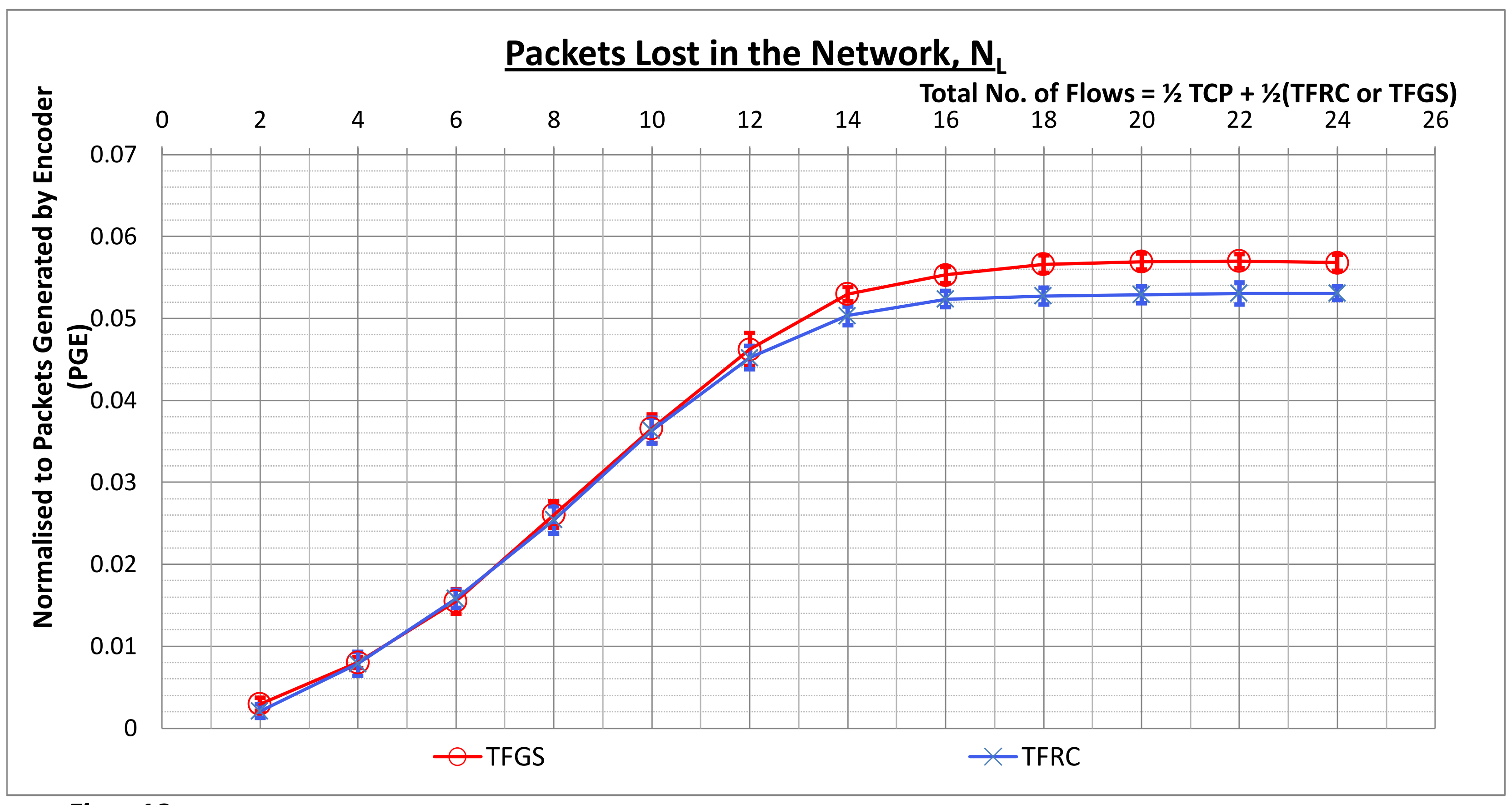


Fig. 18

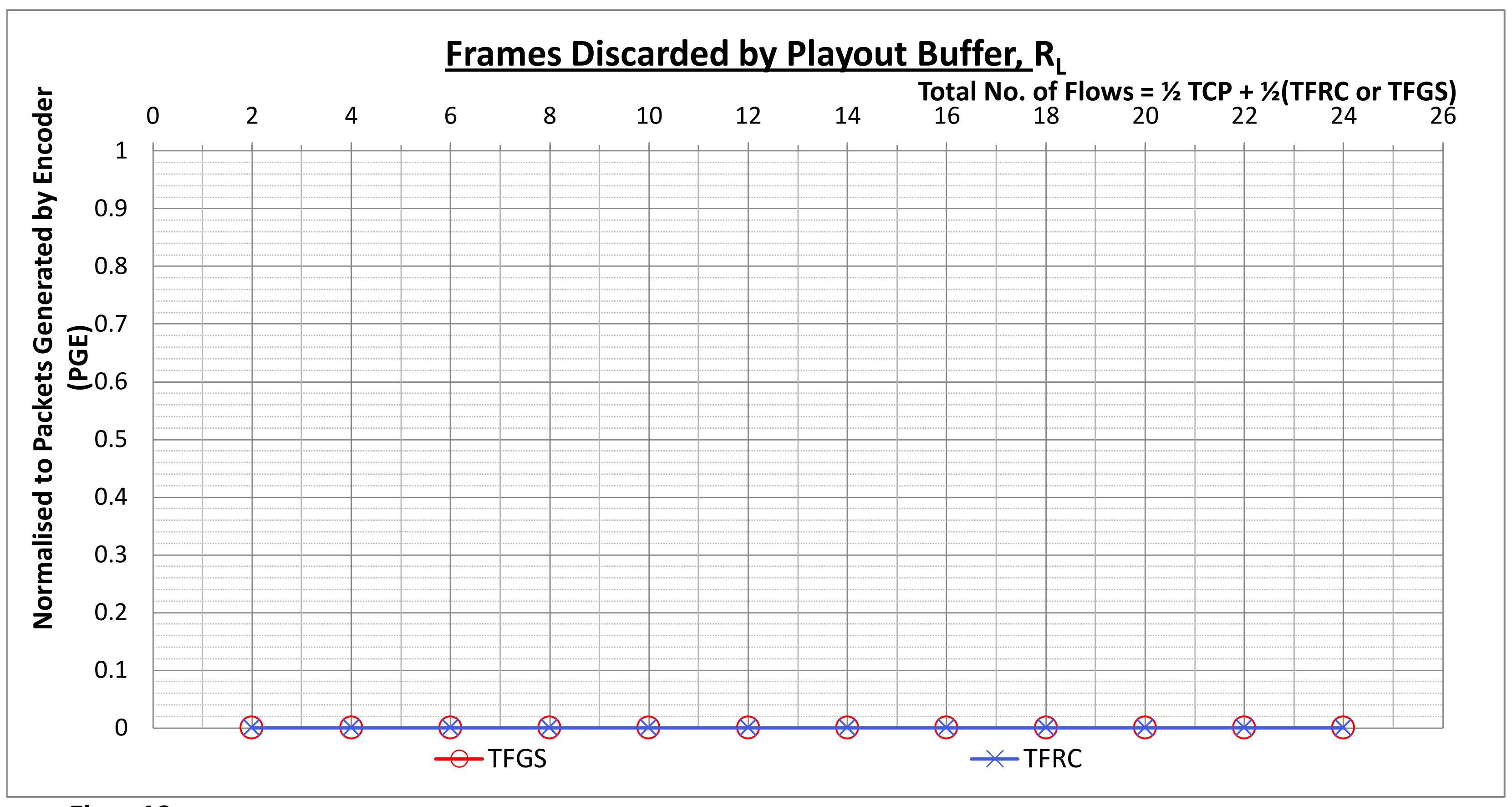


Fig. 19

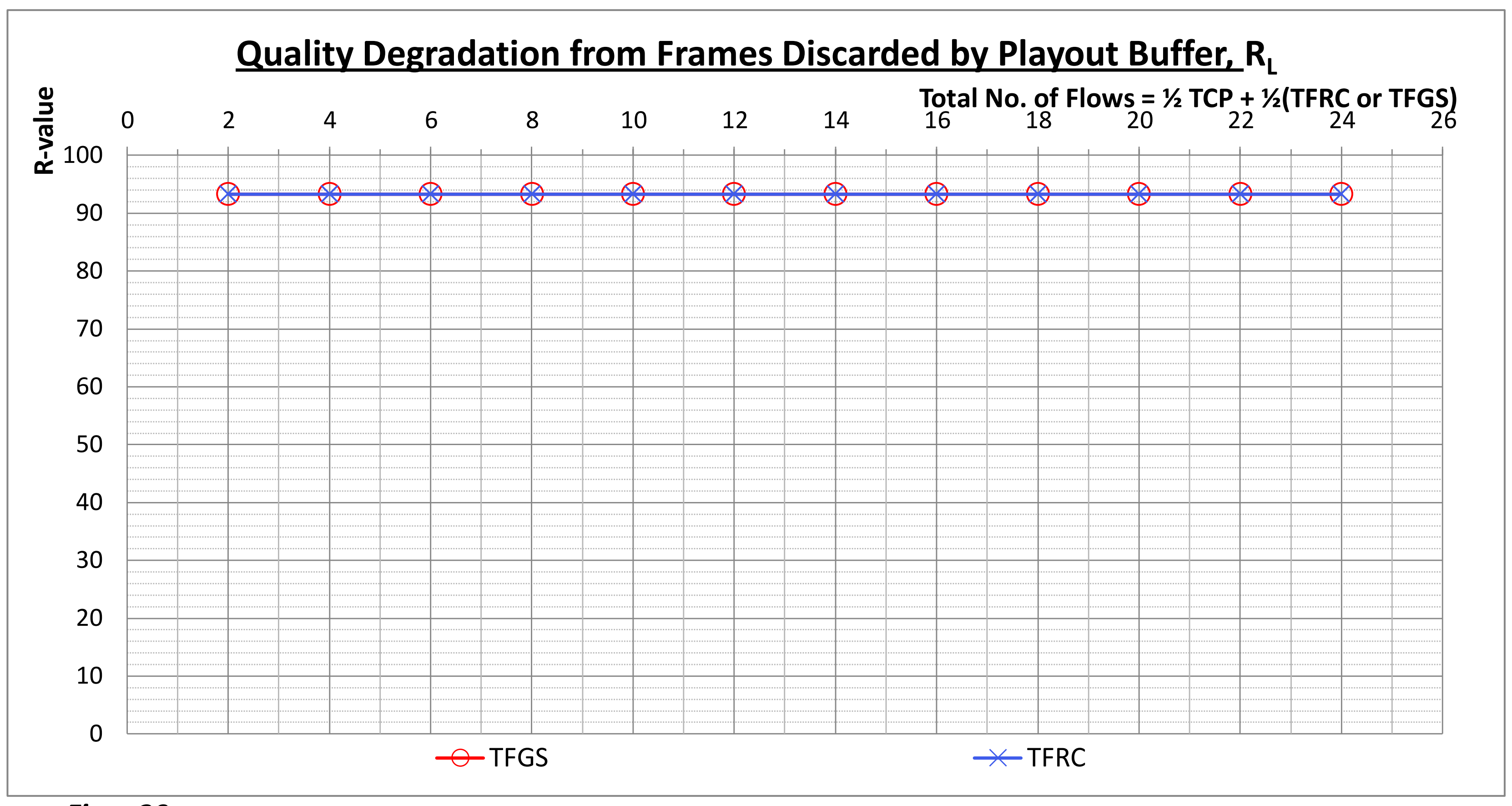


Fig. 20

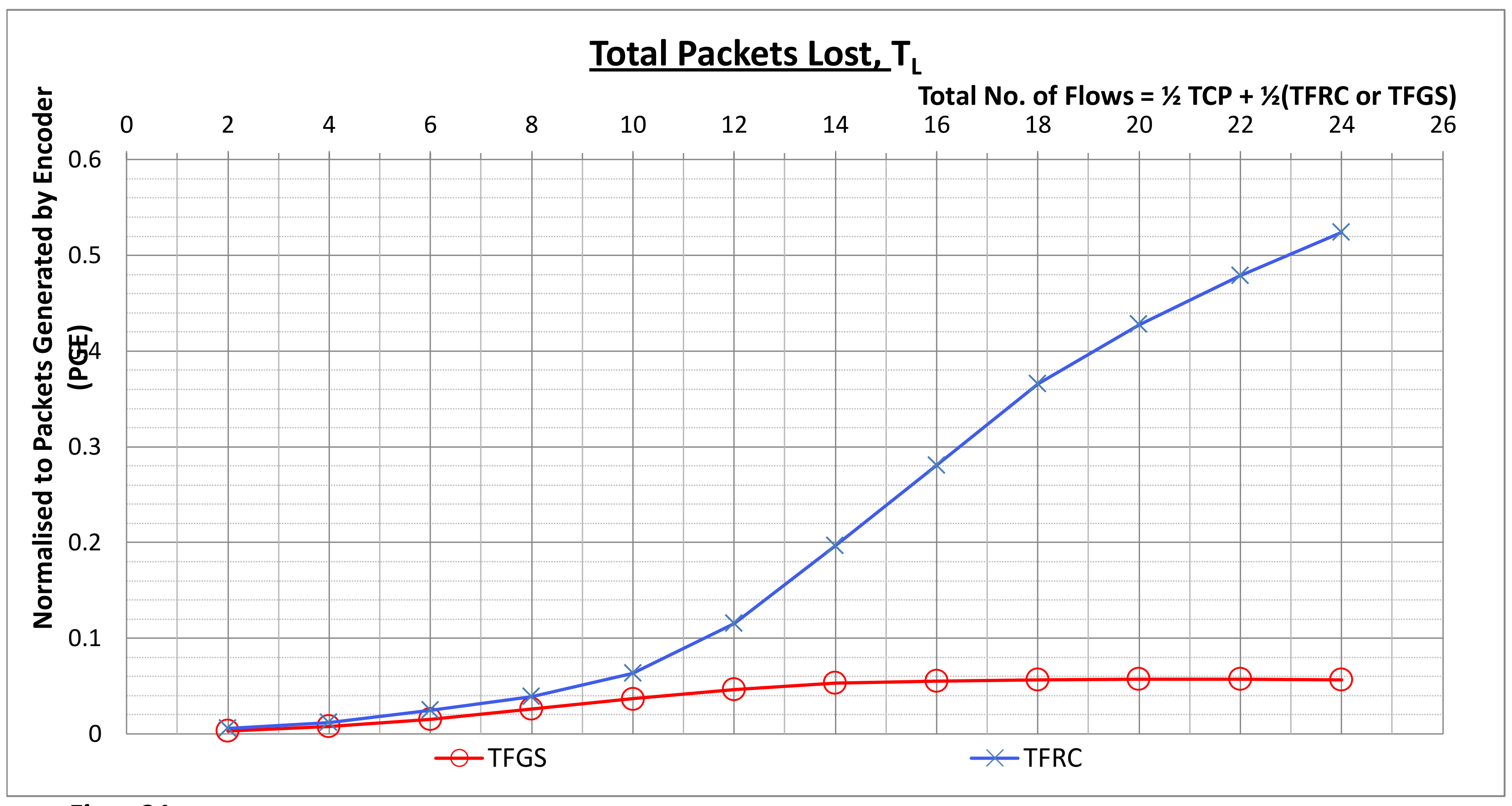


Fig. 21

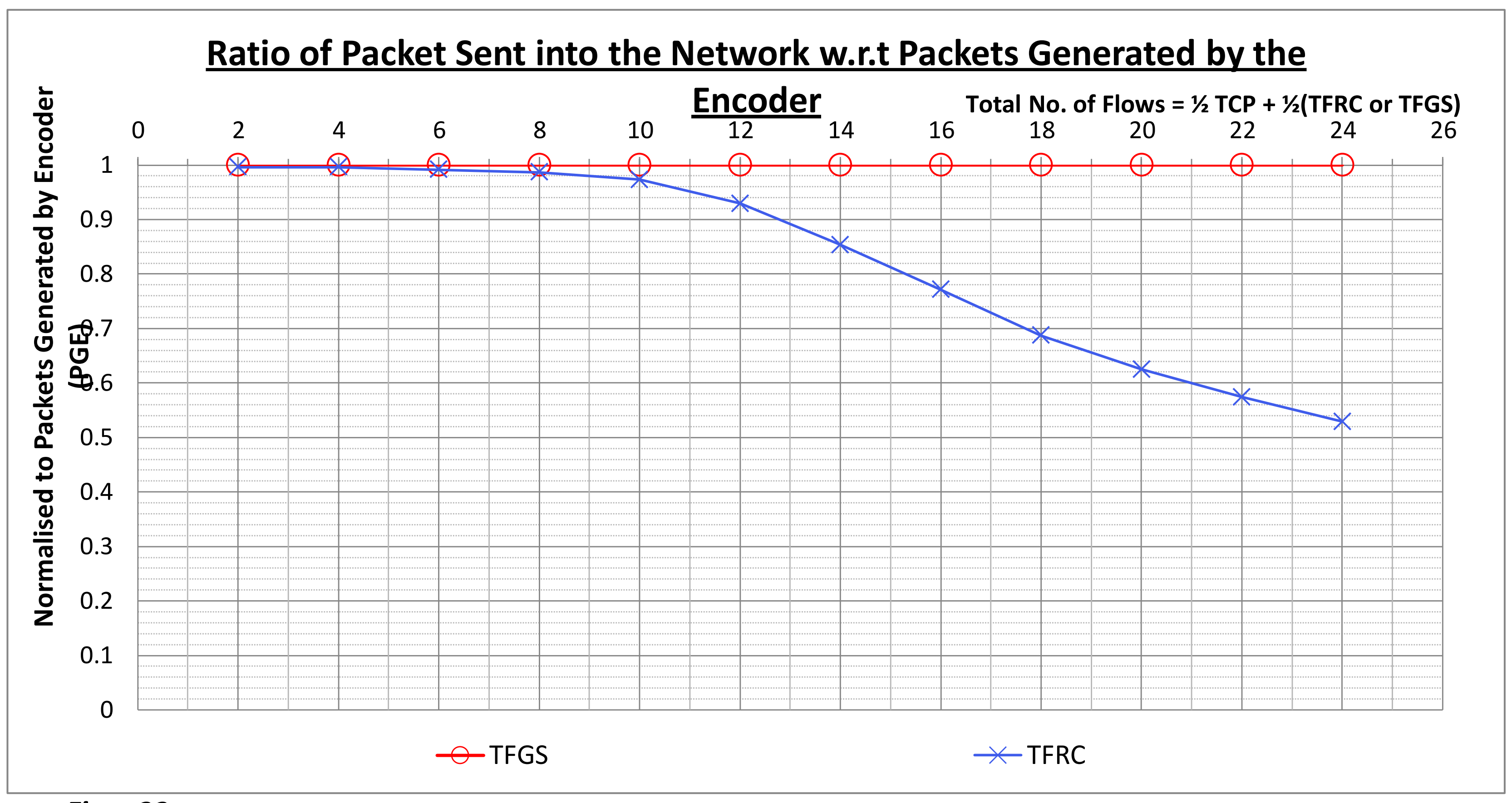


Fig. 22

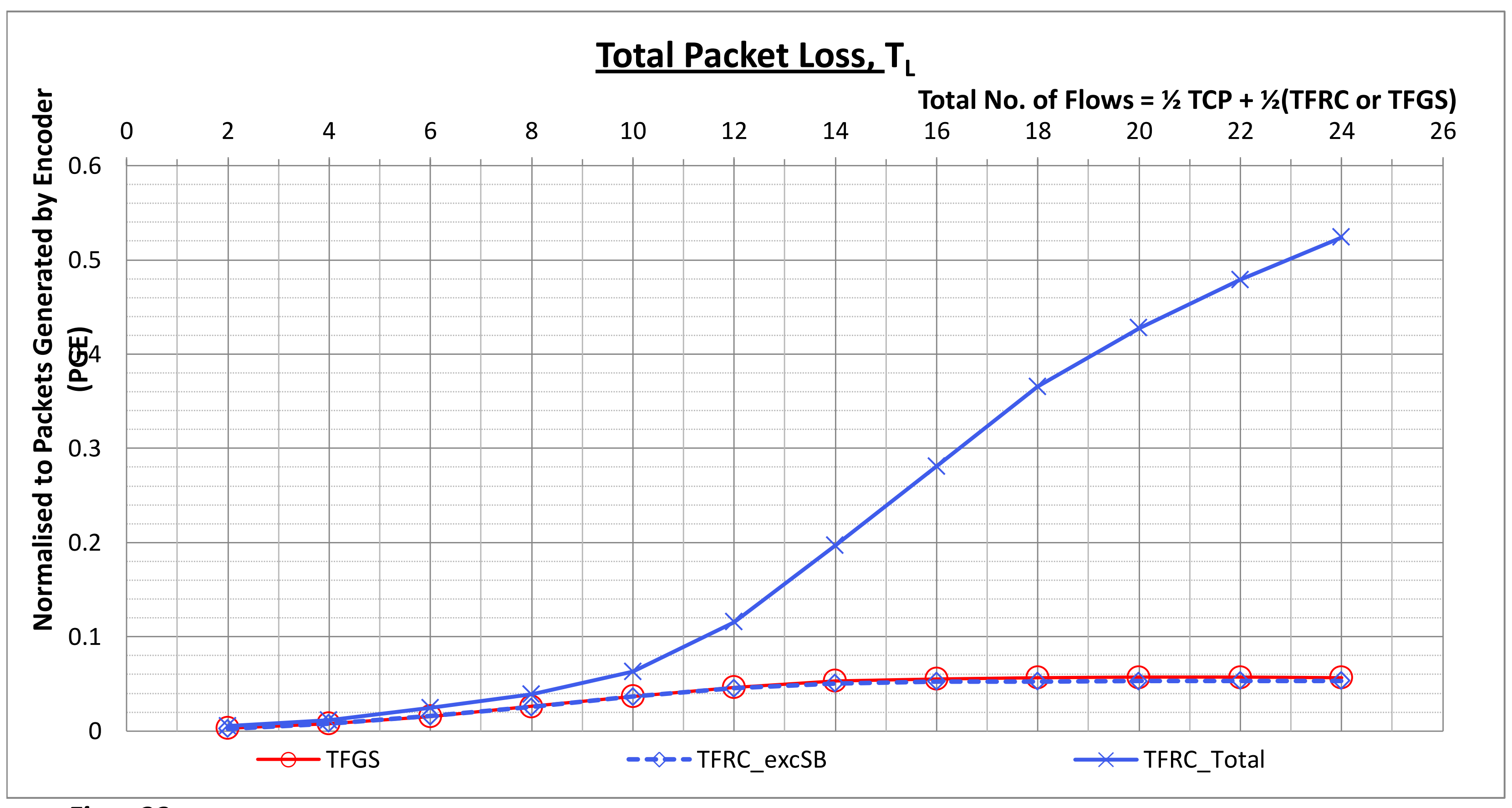


Fig. 23

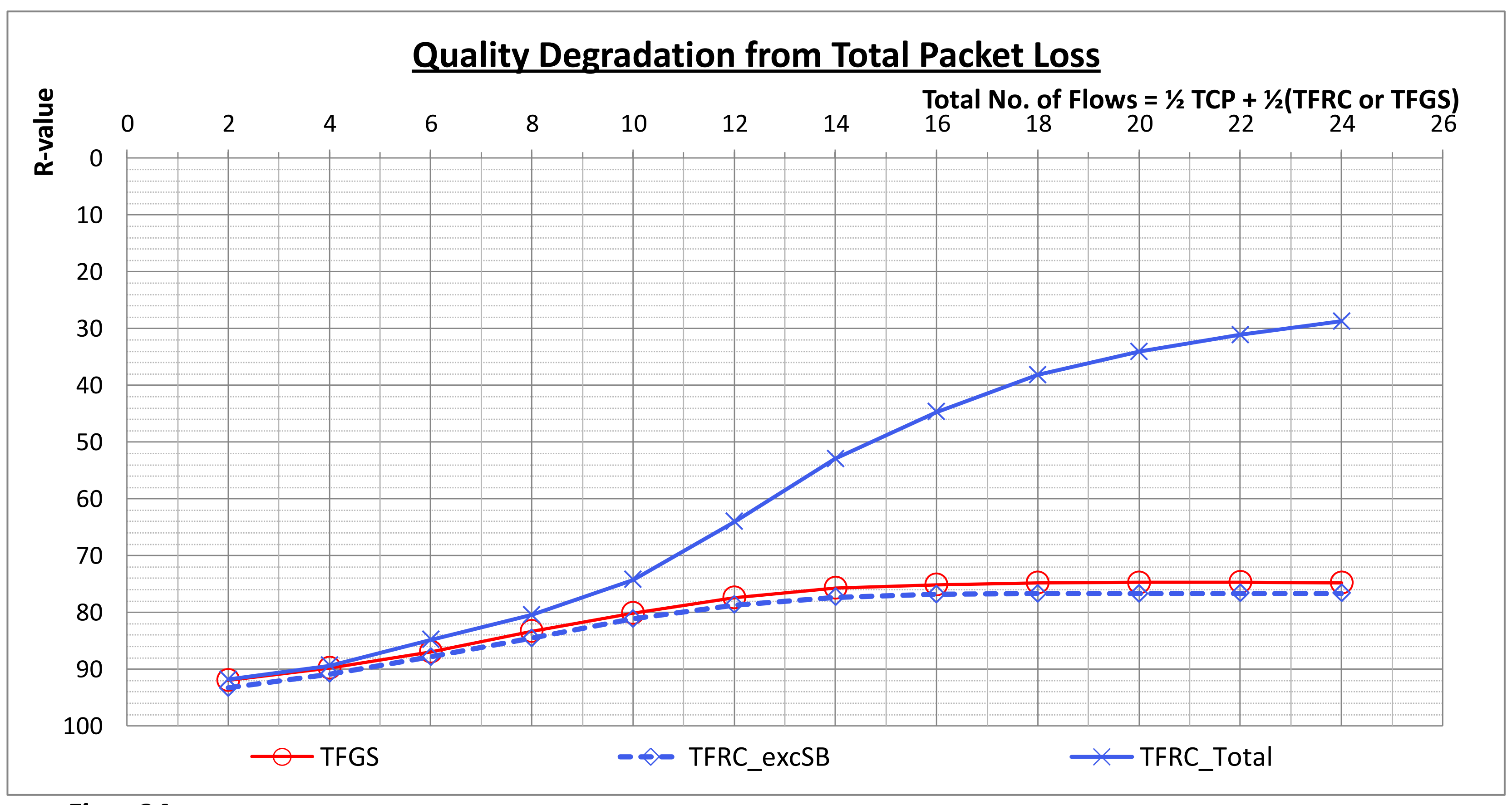


Fig. 24

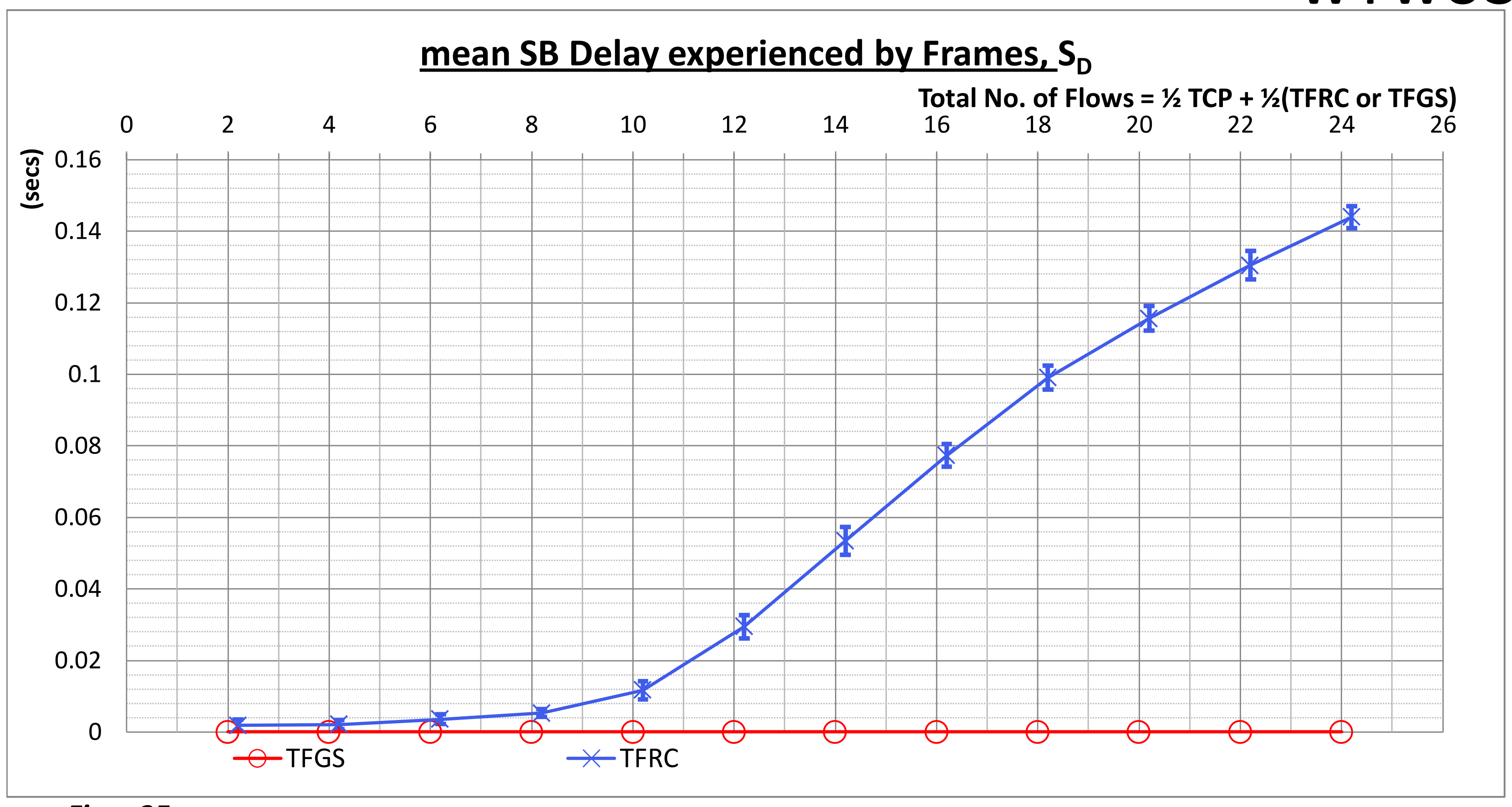


Fig. 25

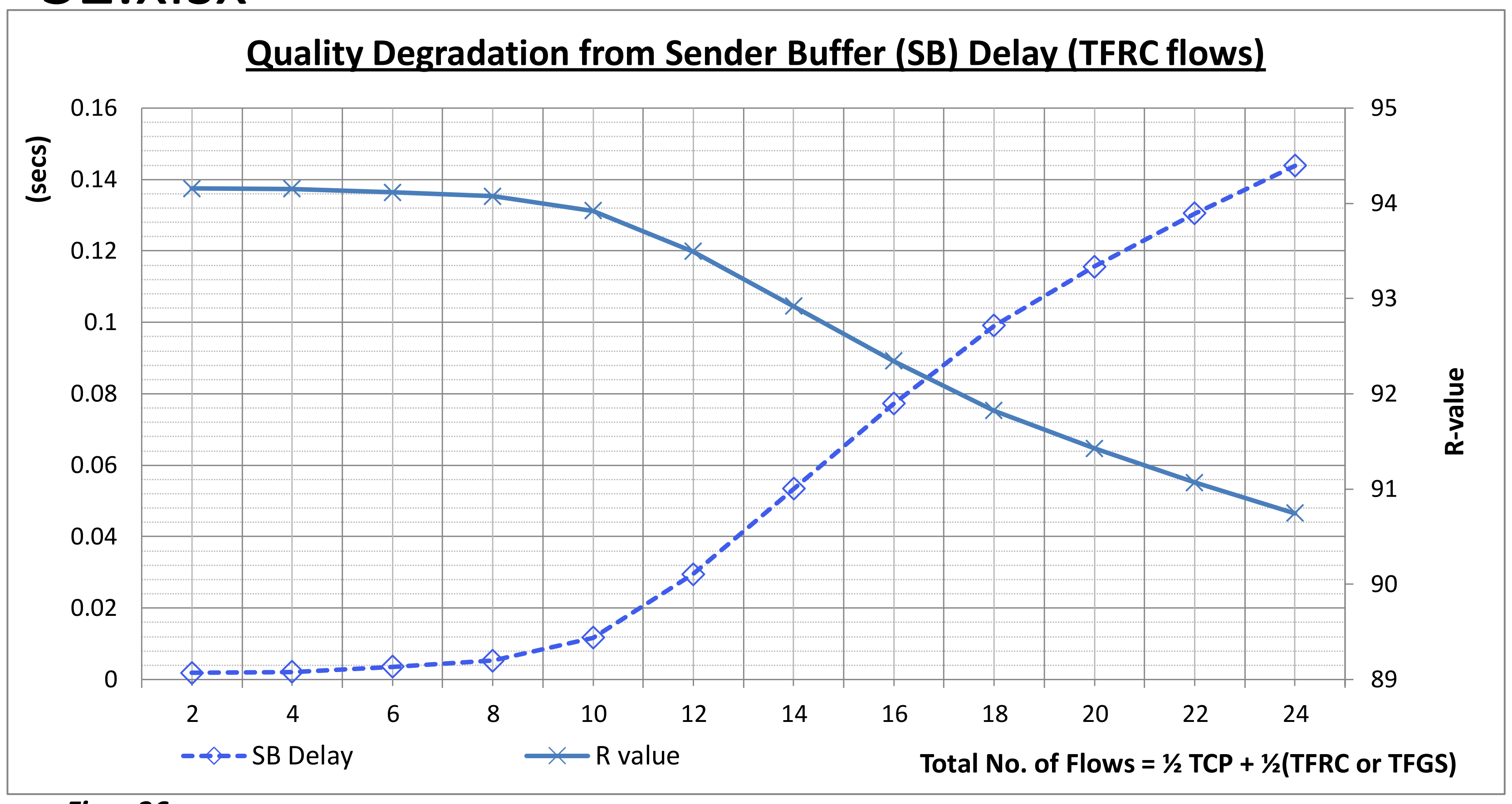


Fig. 26

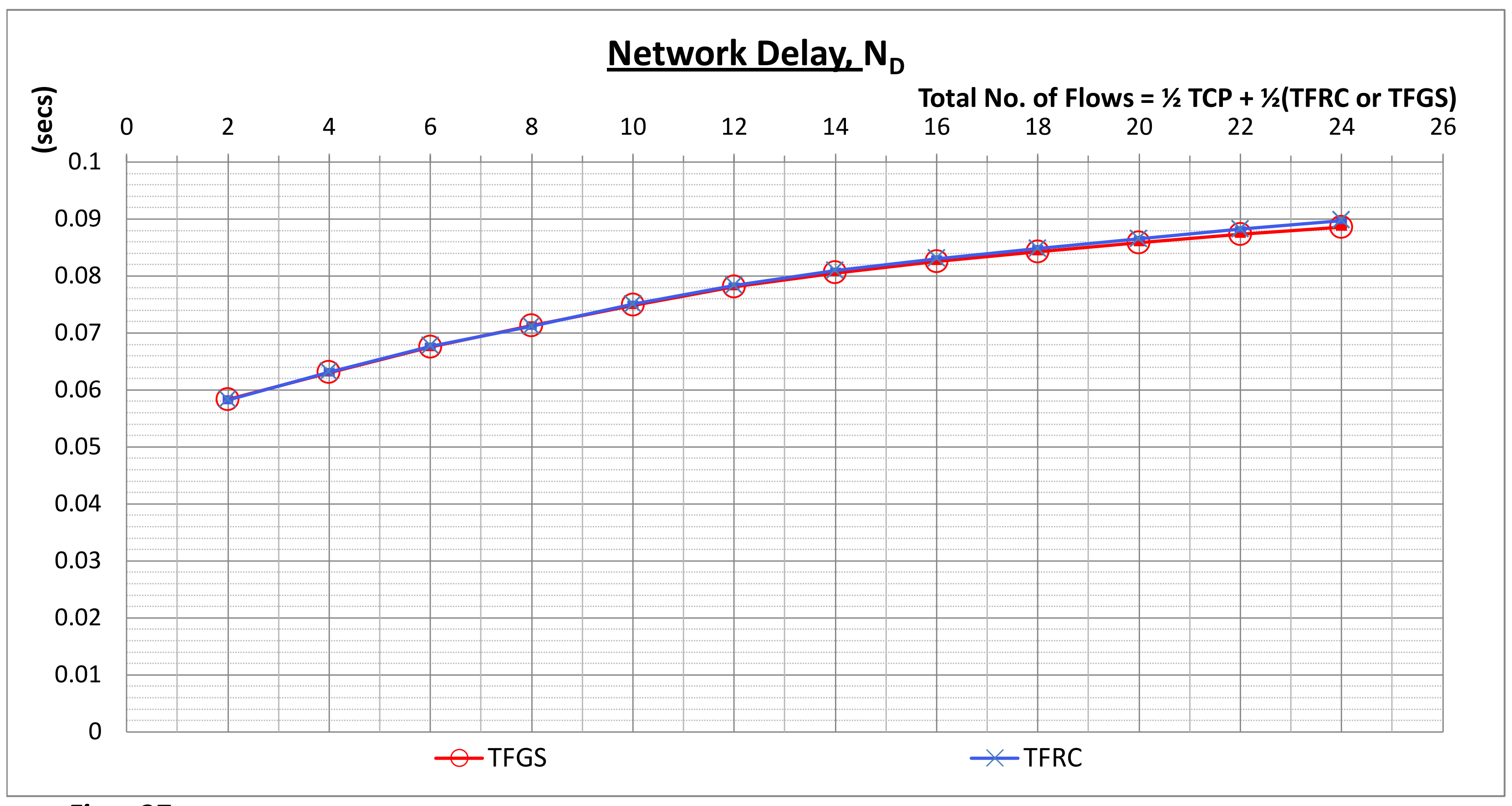


Fig. 27

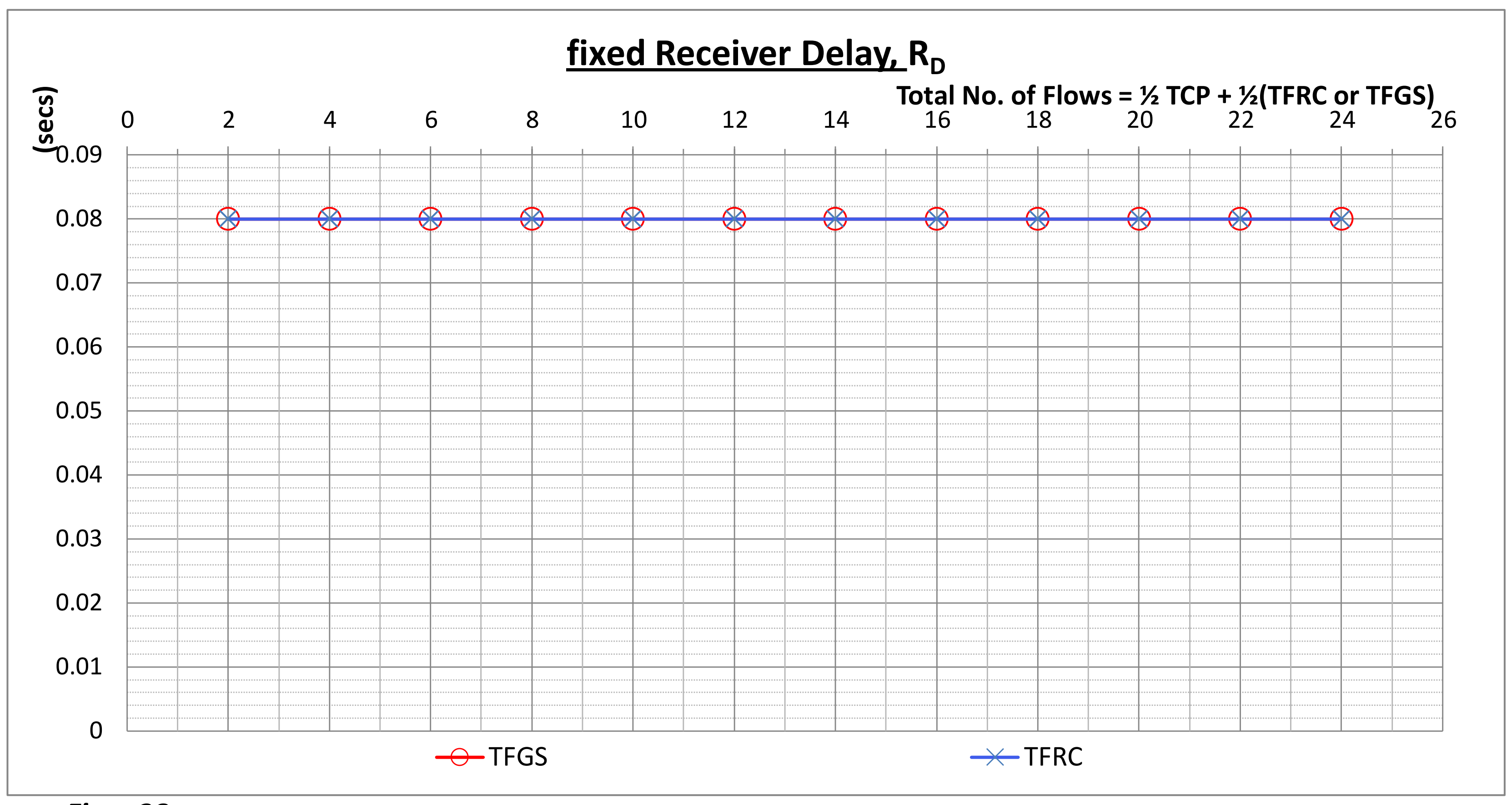


Fig. 28

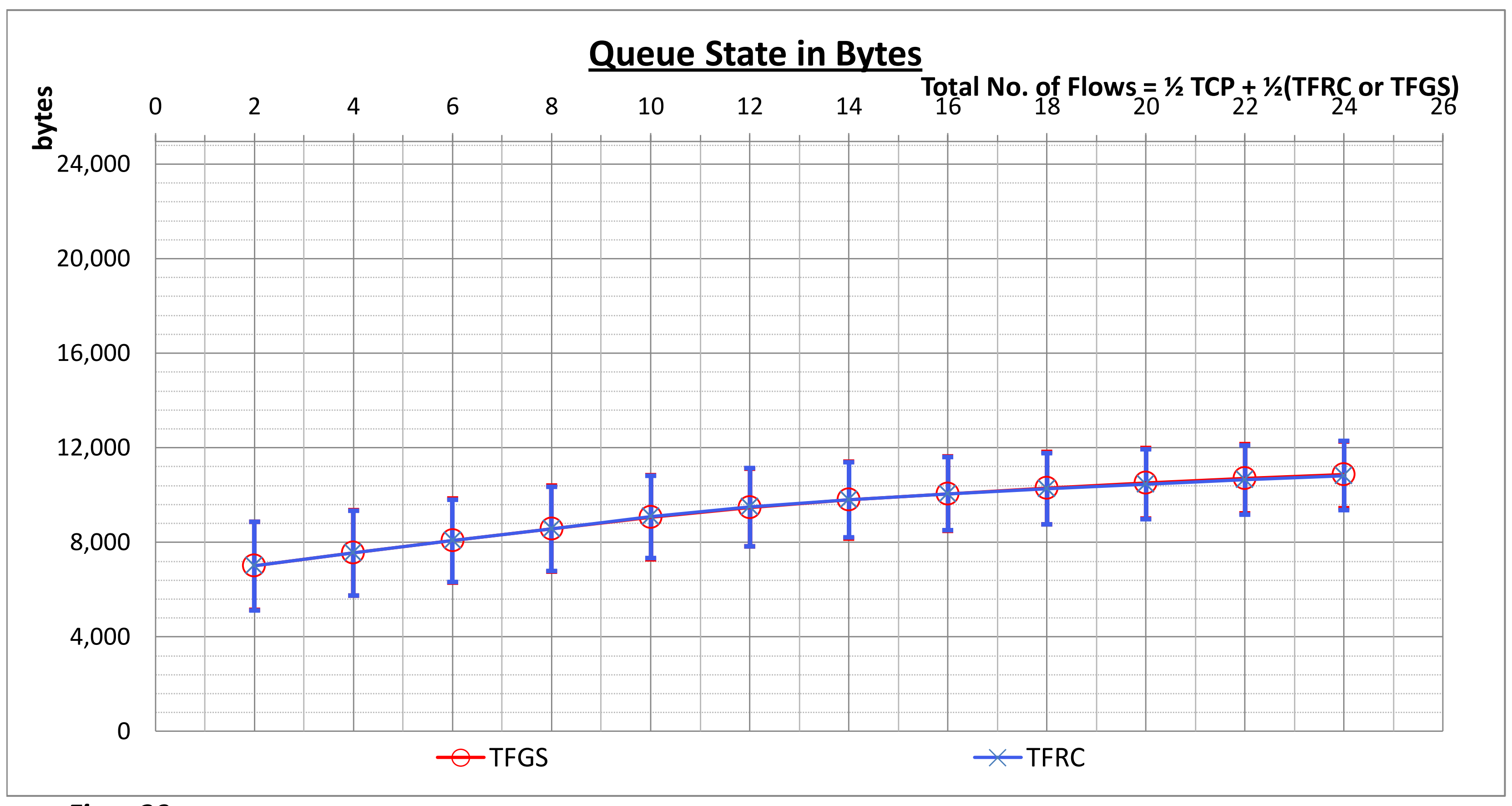


Fig. 29

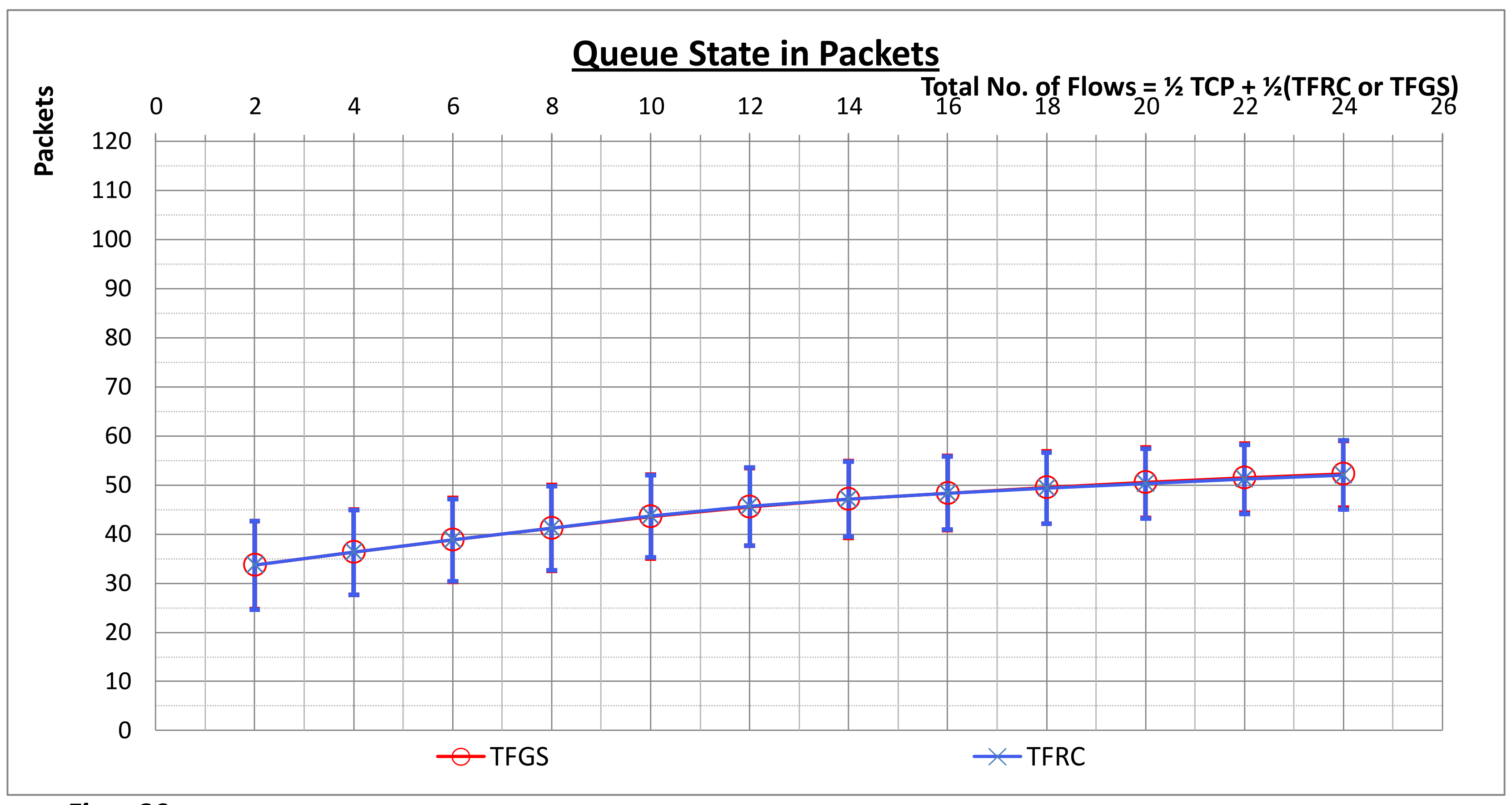


Fig. 30

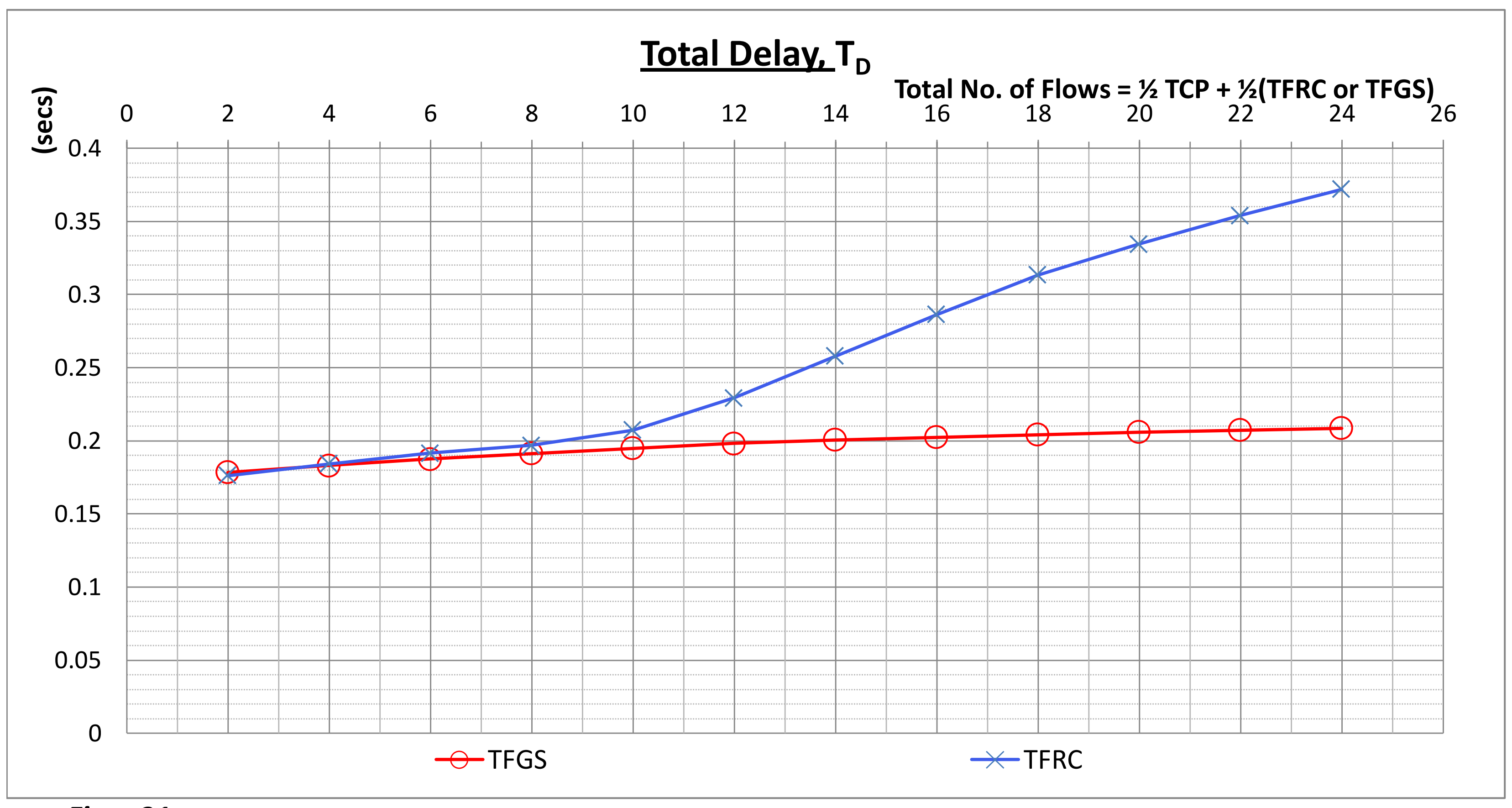


Fig. 31

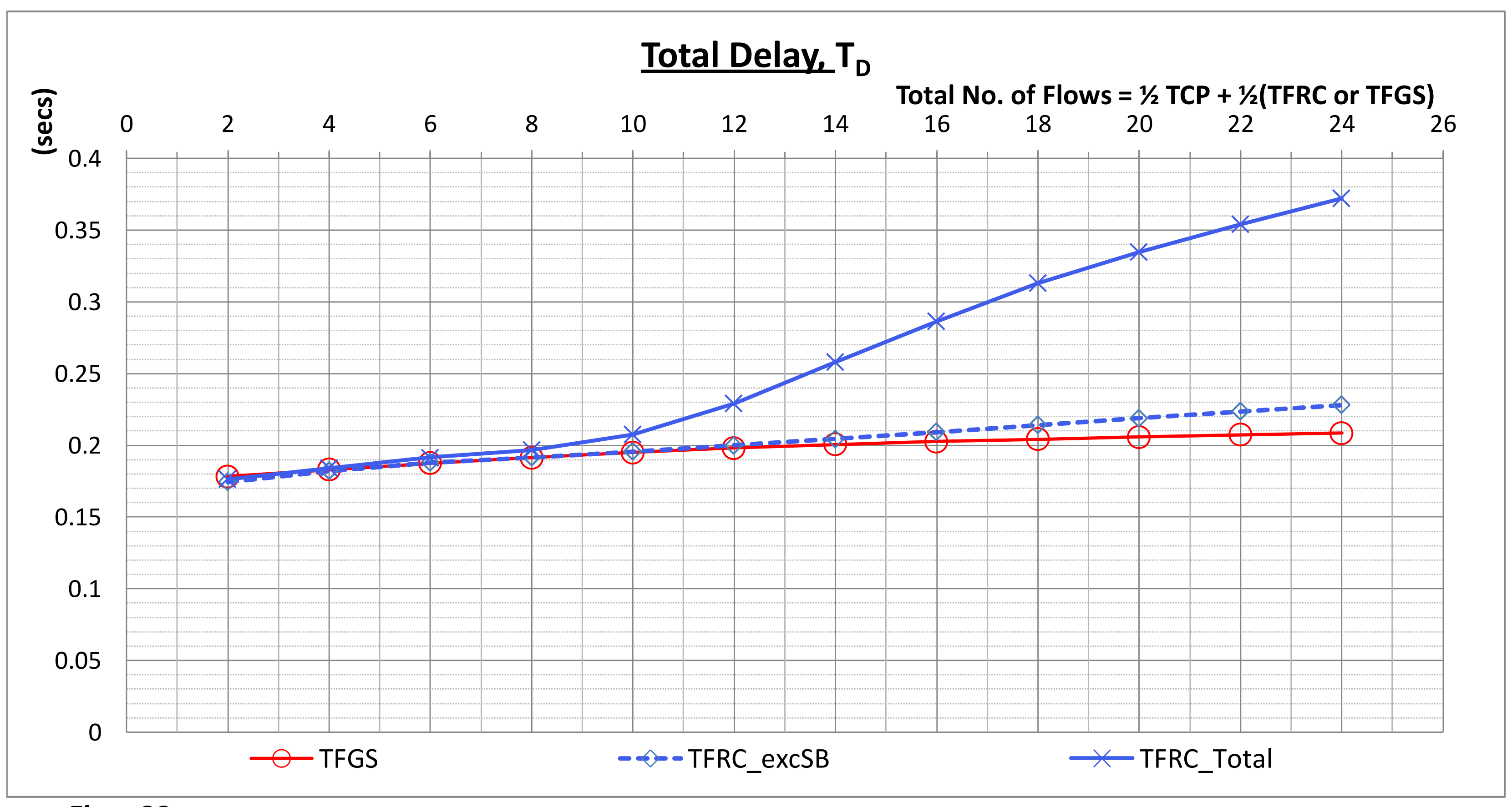


Fig. 32

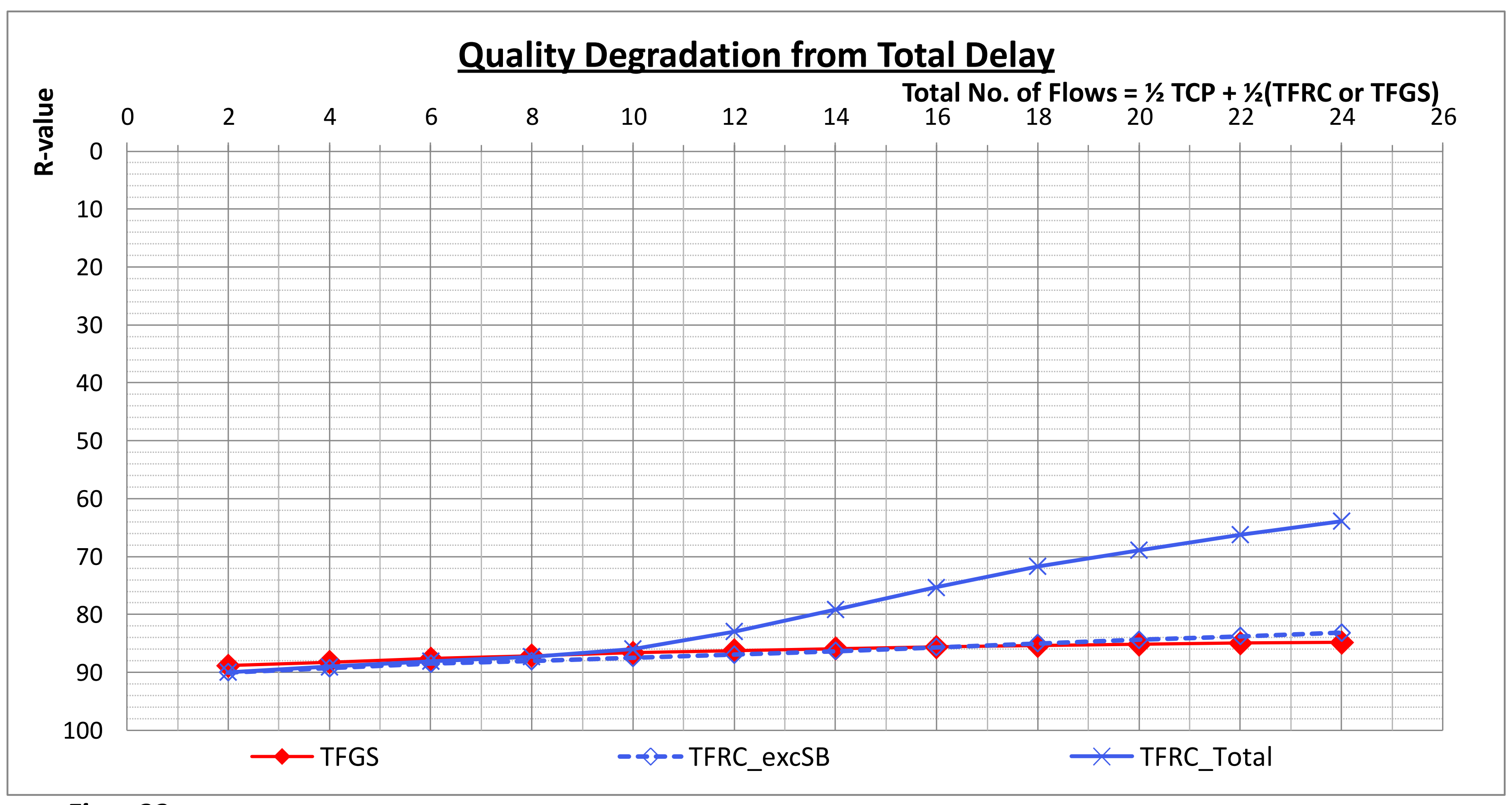


Fig. 33

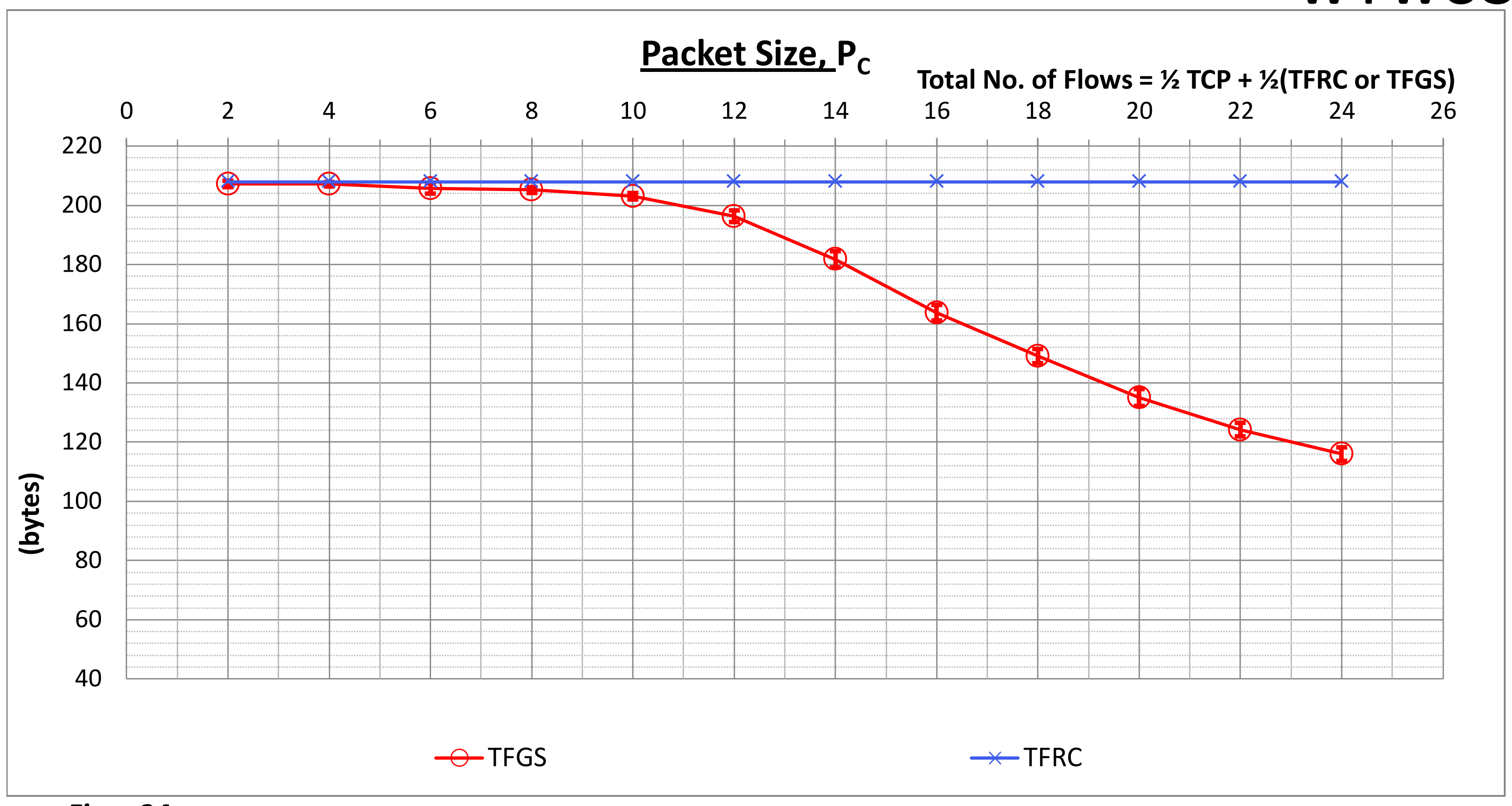


Fig. 34

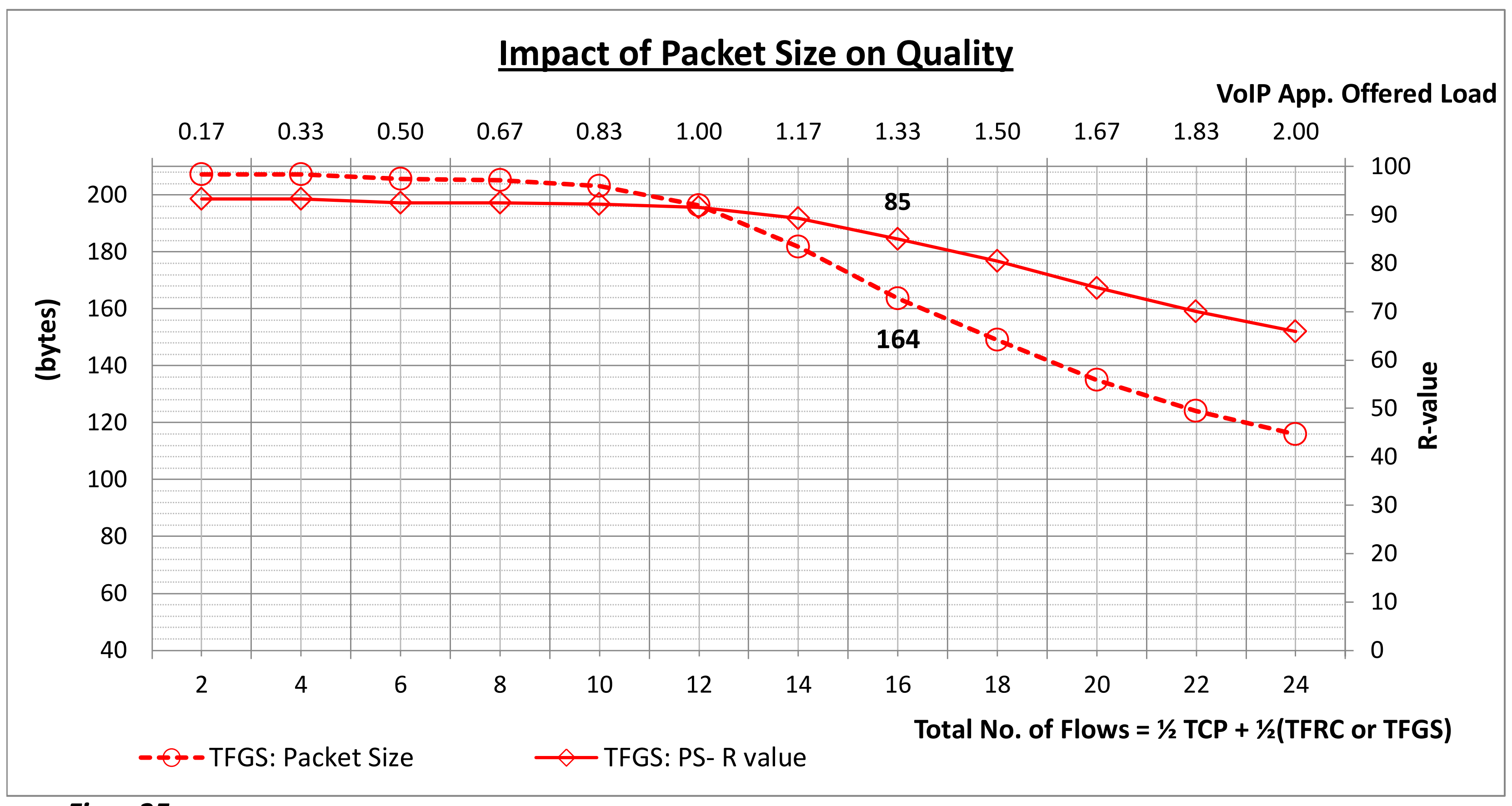


Fig. 35

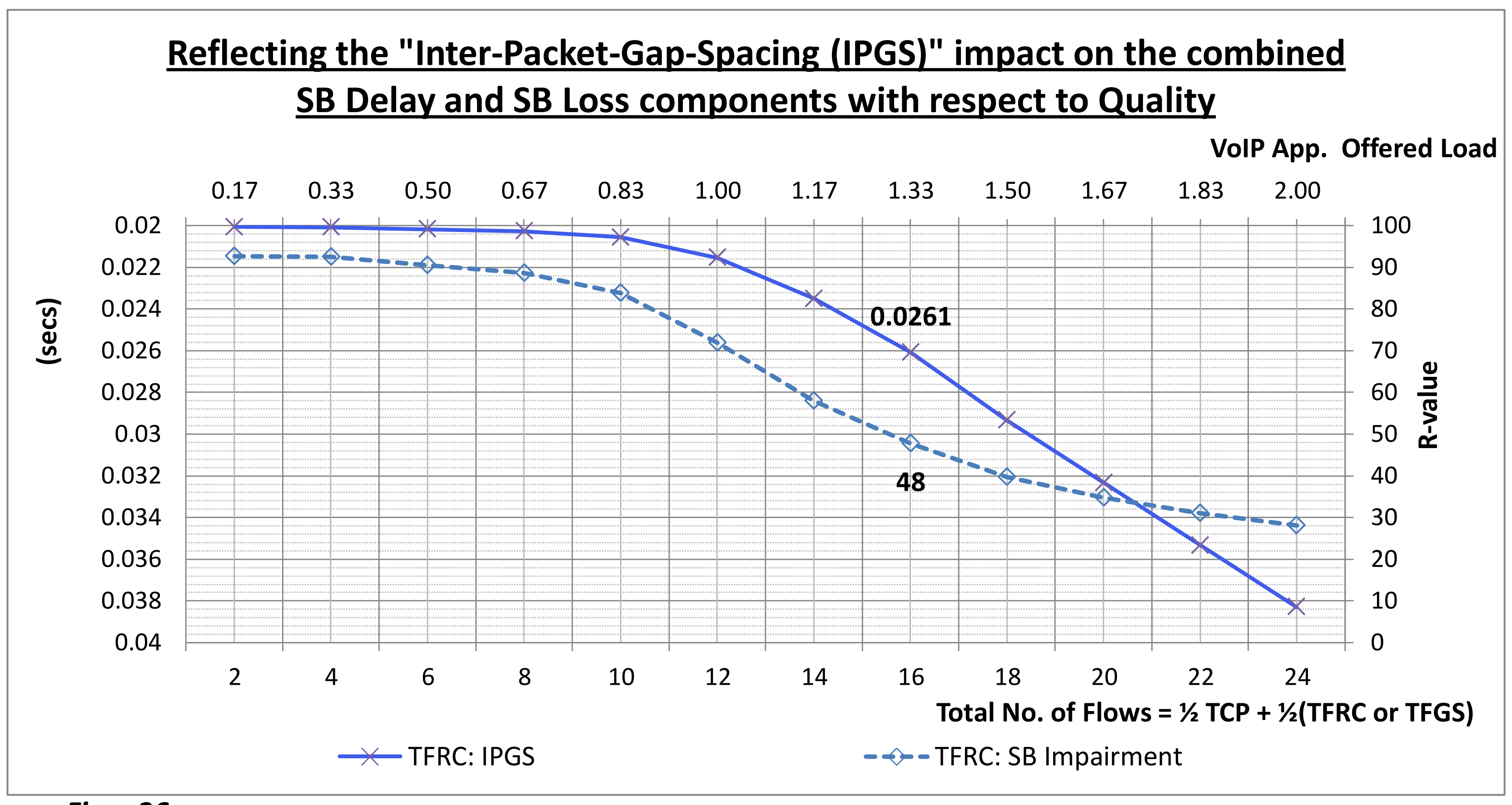


Fig. 36

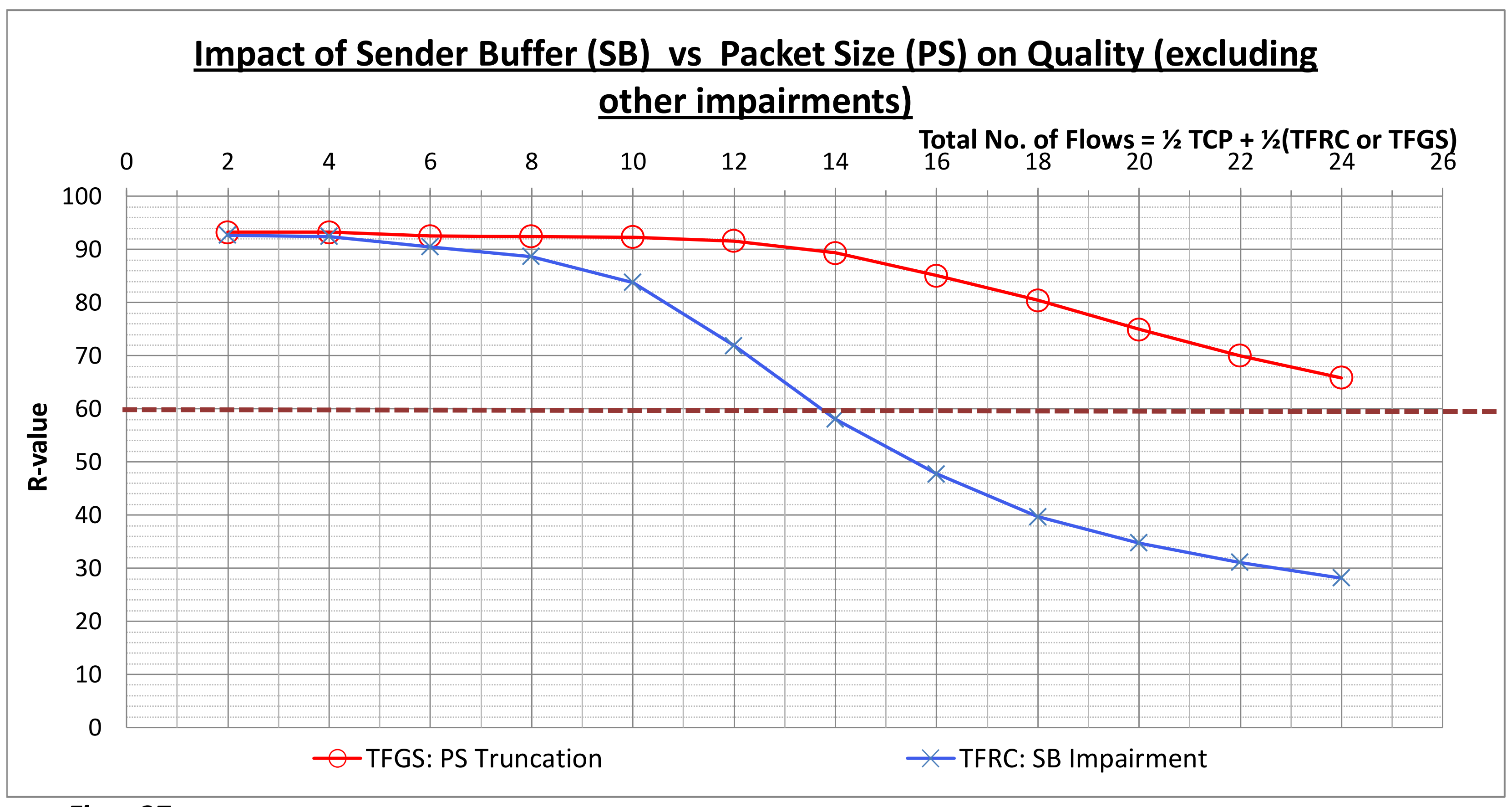


Fig. 37

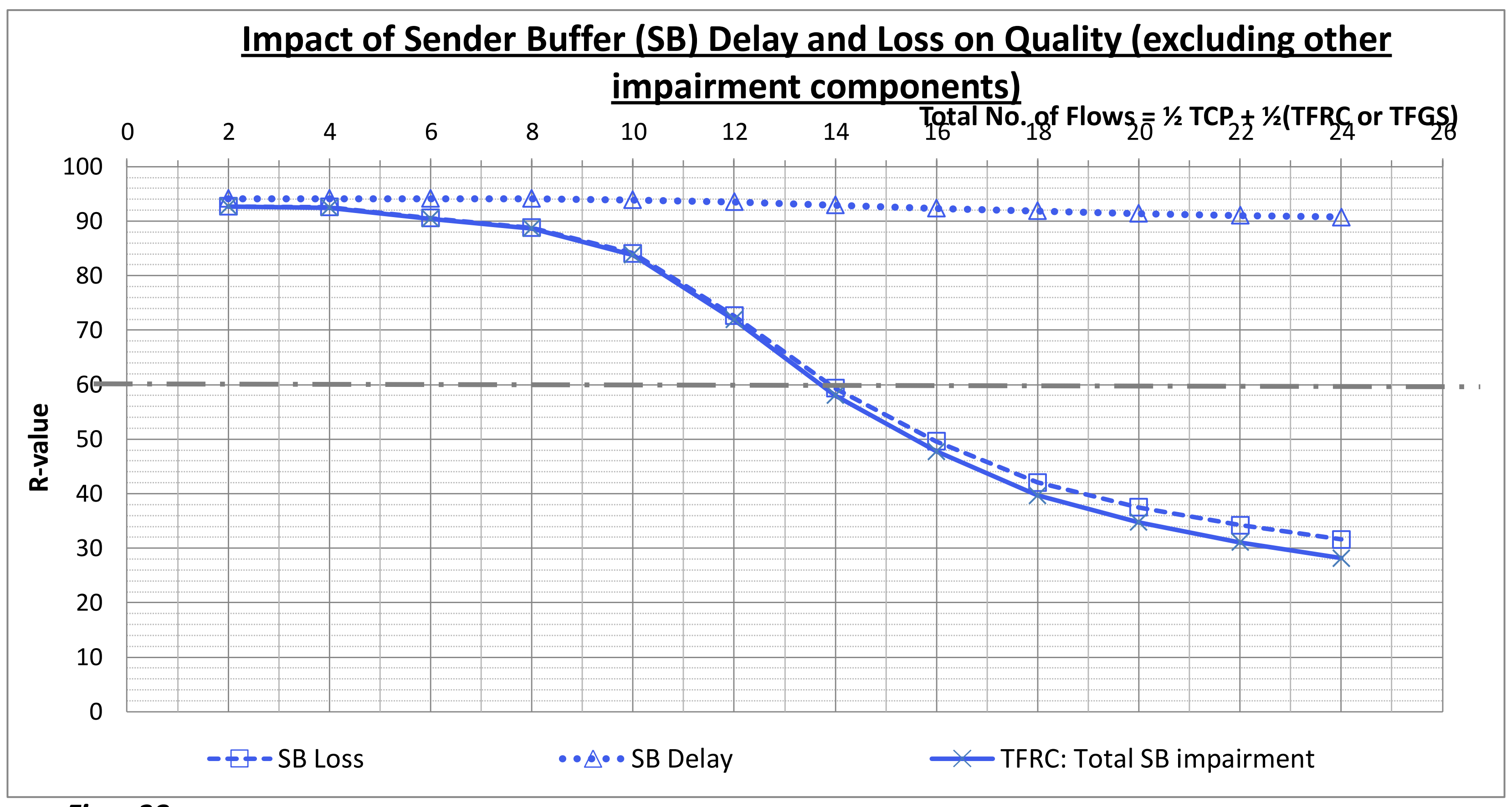


Fig. 38

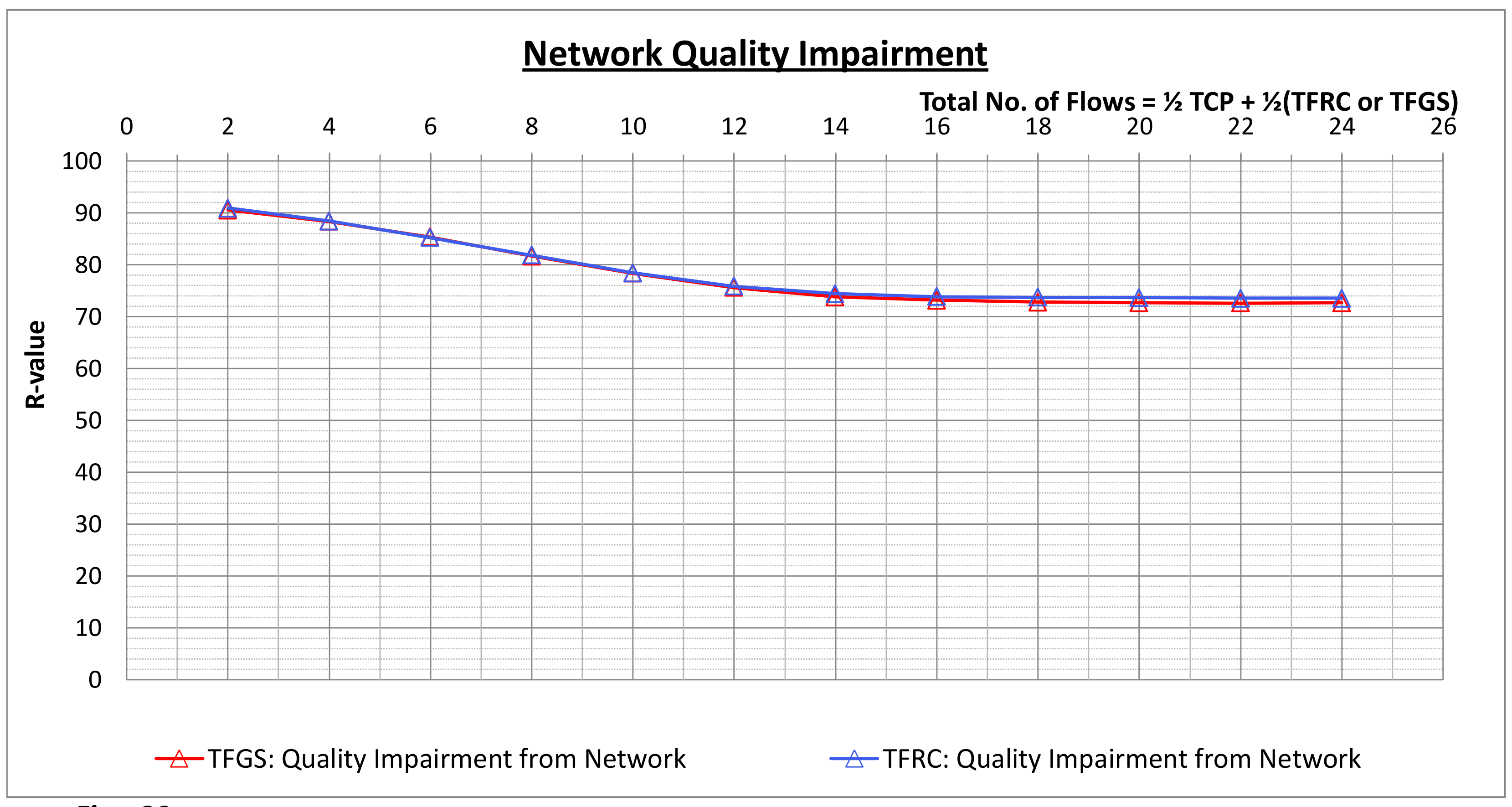


Fig. 39

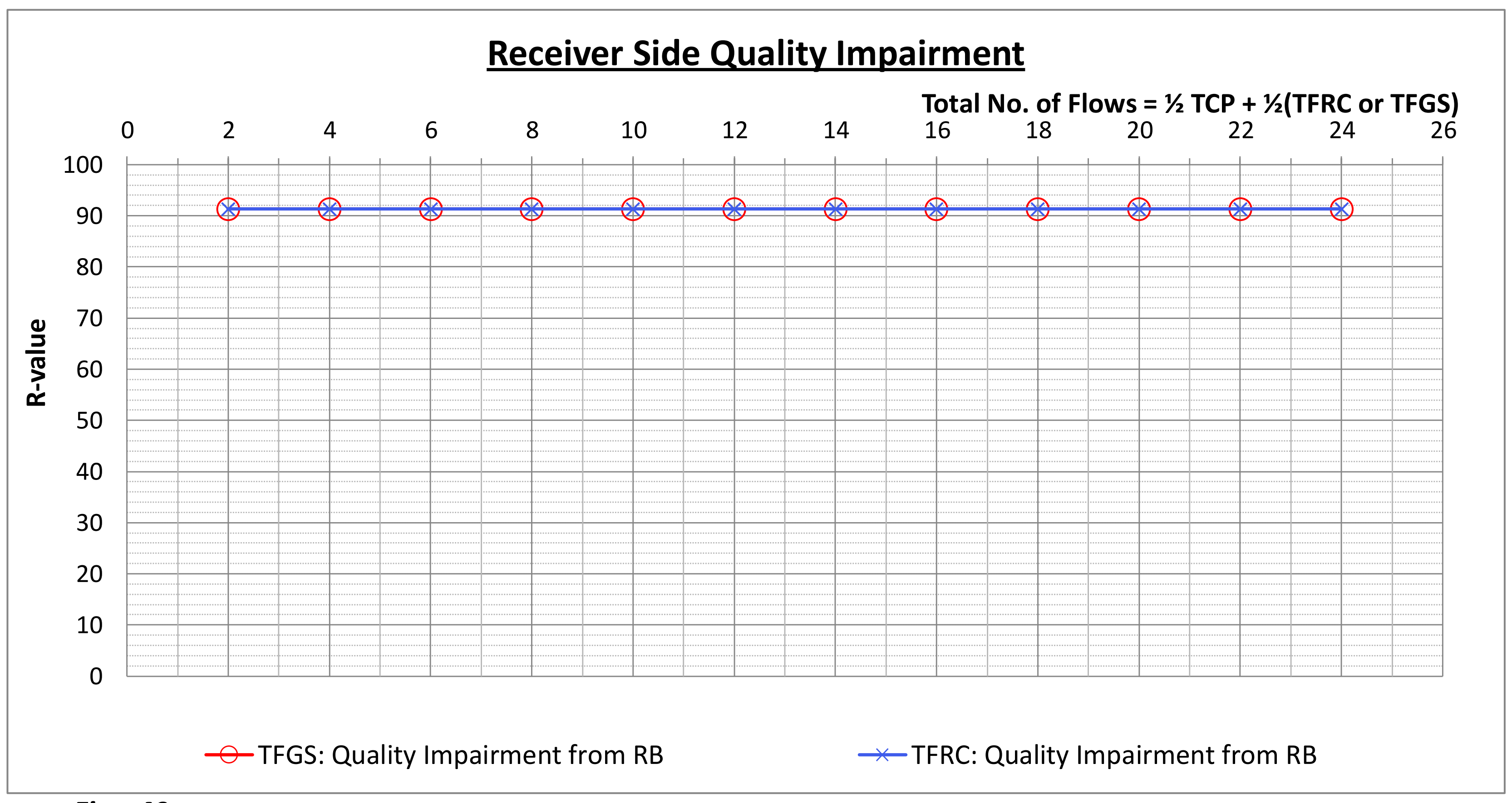


Fig. 40

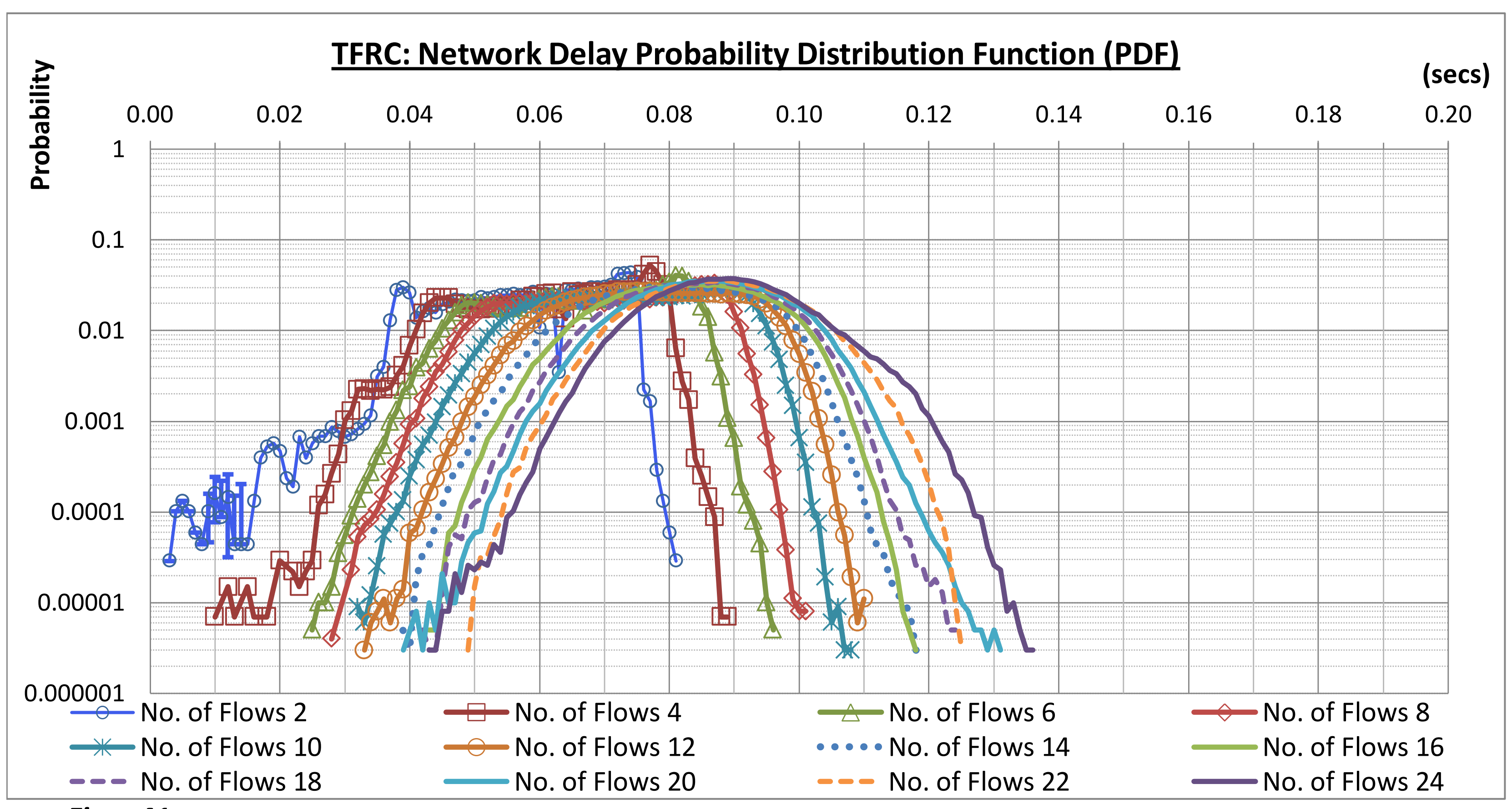


Fig. 41

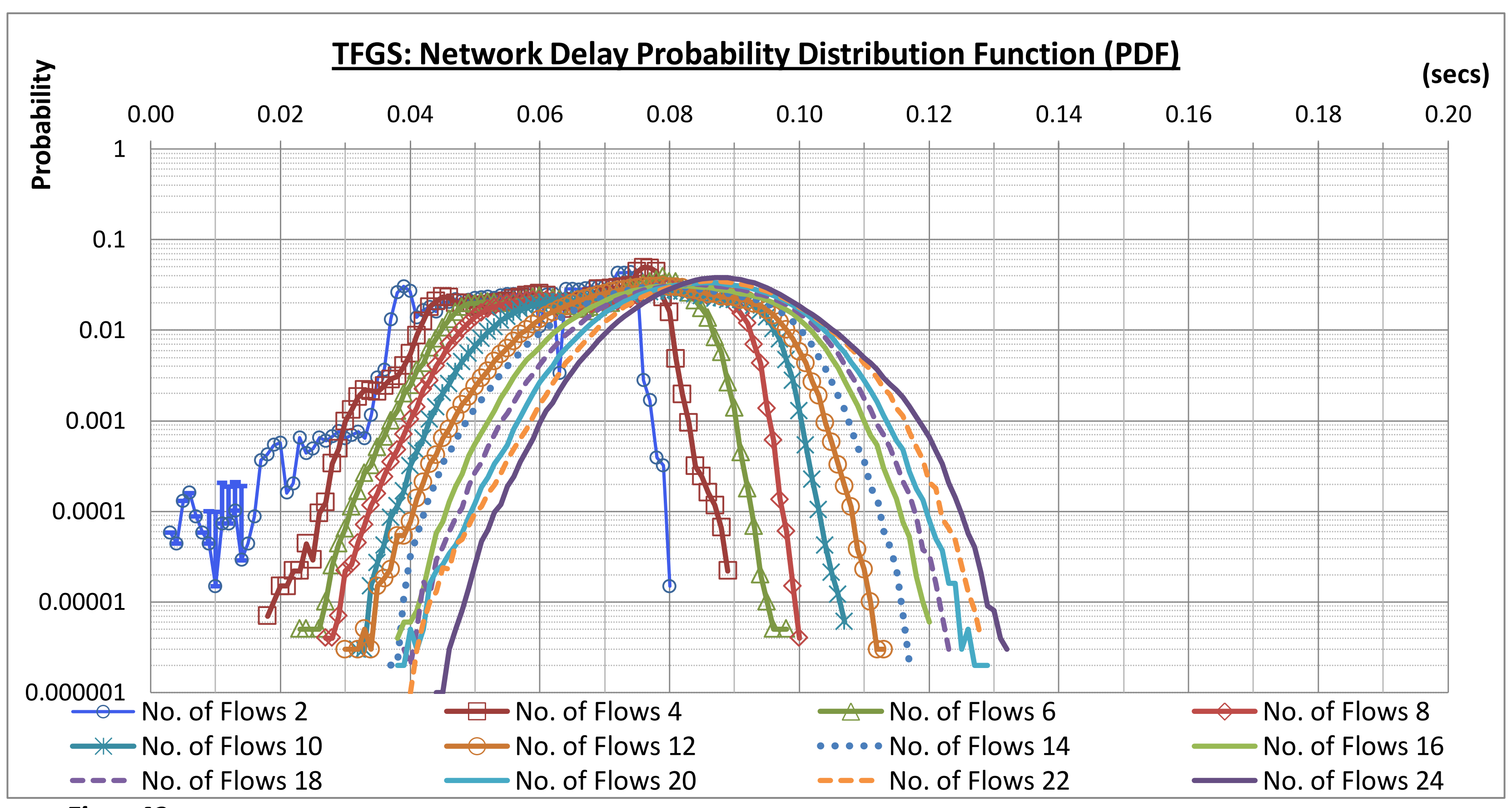


Fig. 42

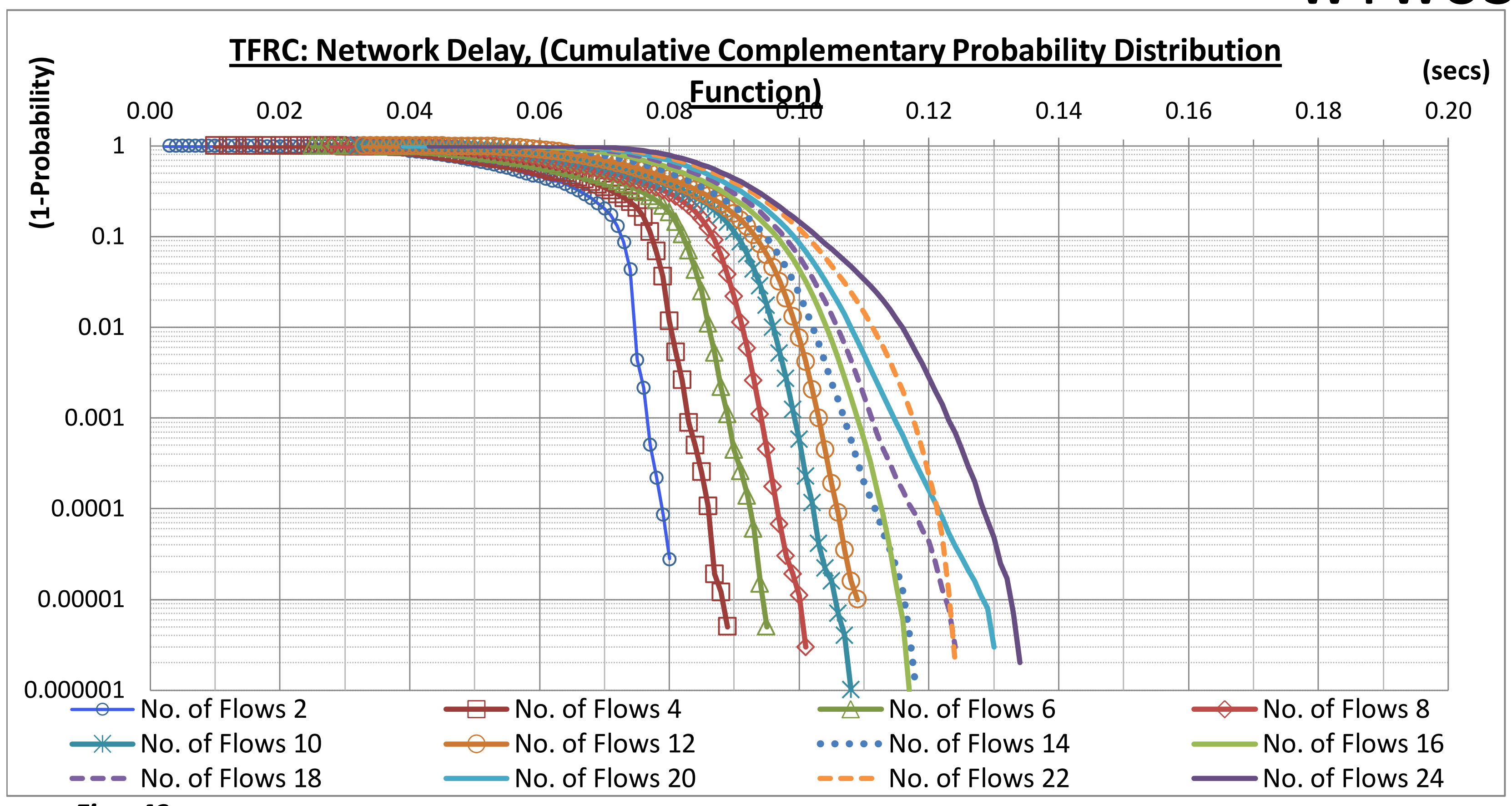


Fig. 43

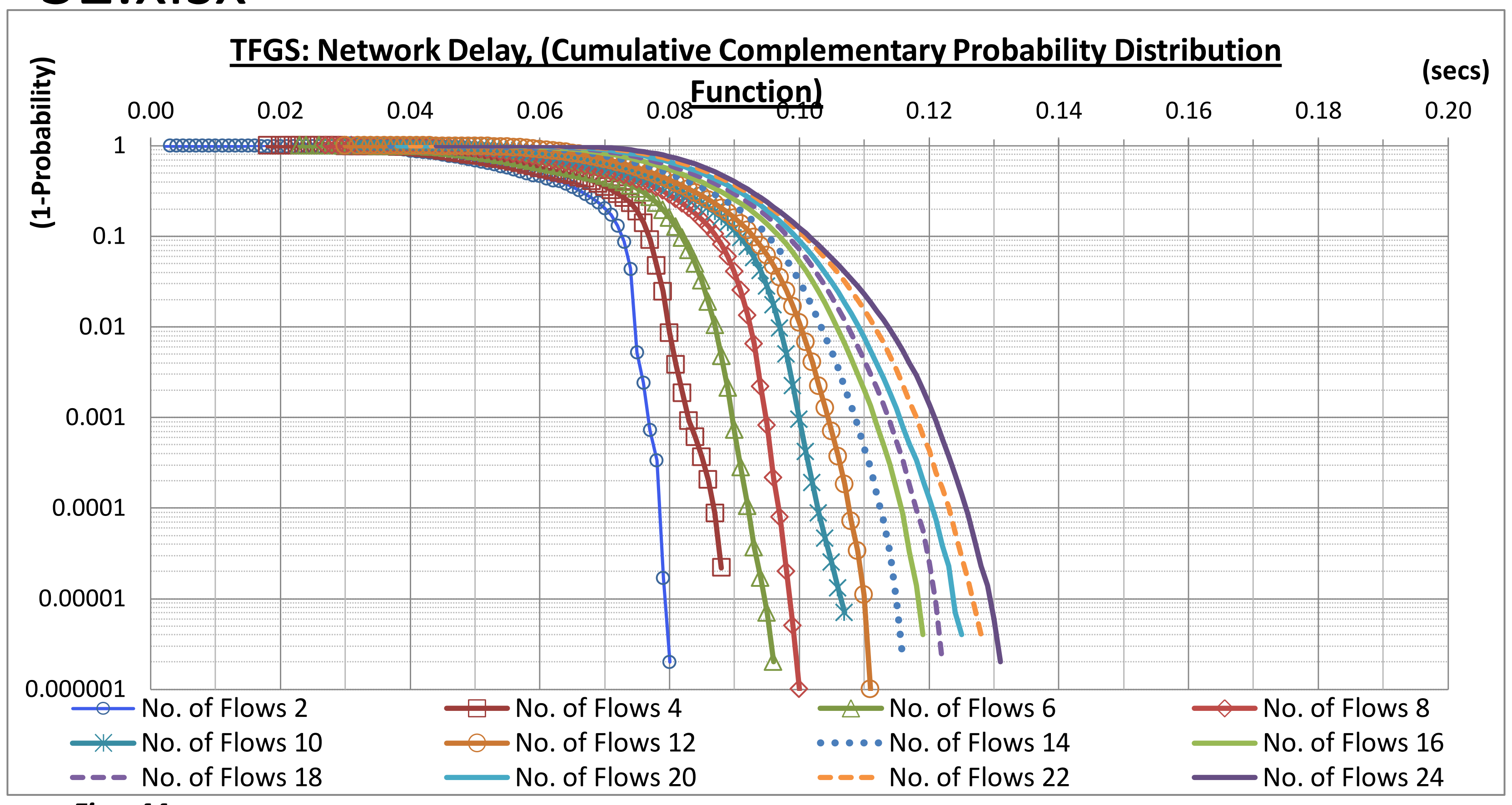


Fig. 44

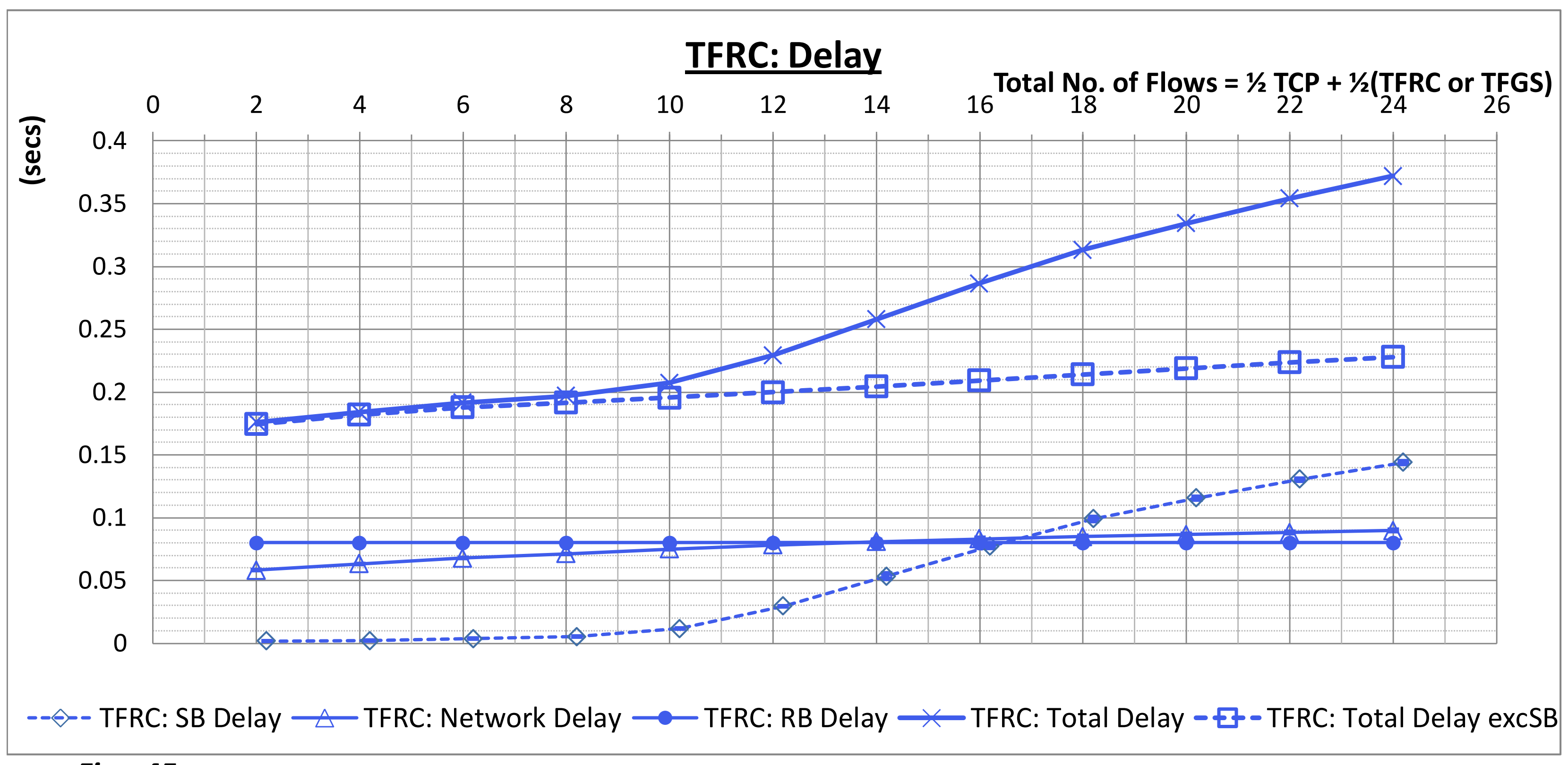


Fig. 45

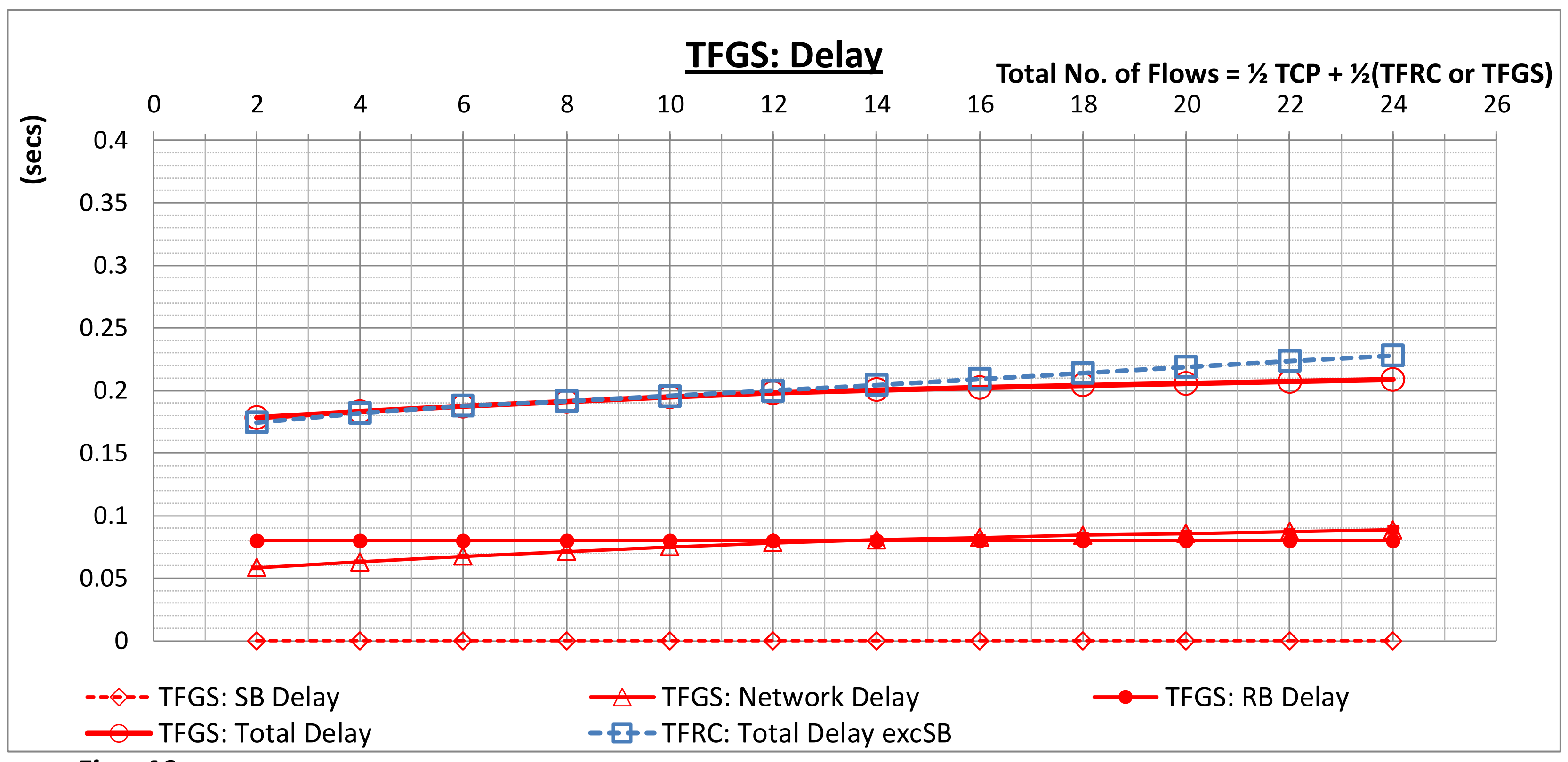


Fig. 46

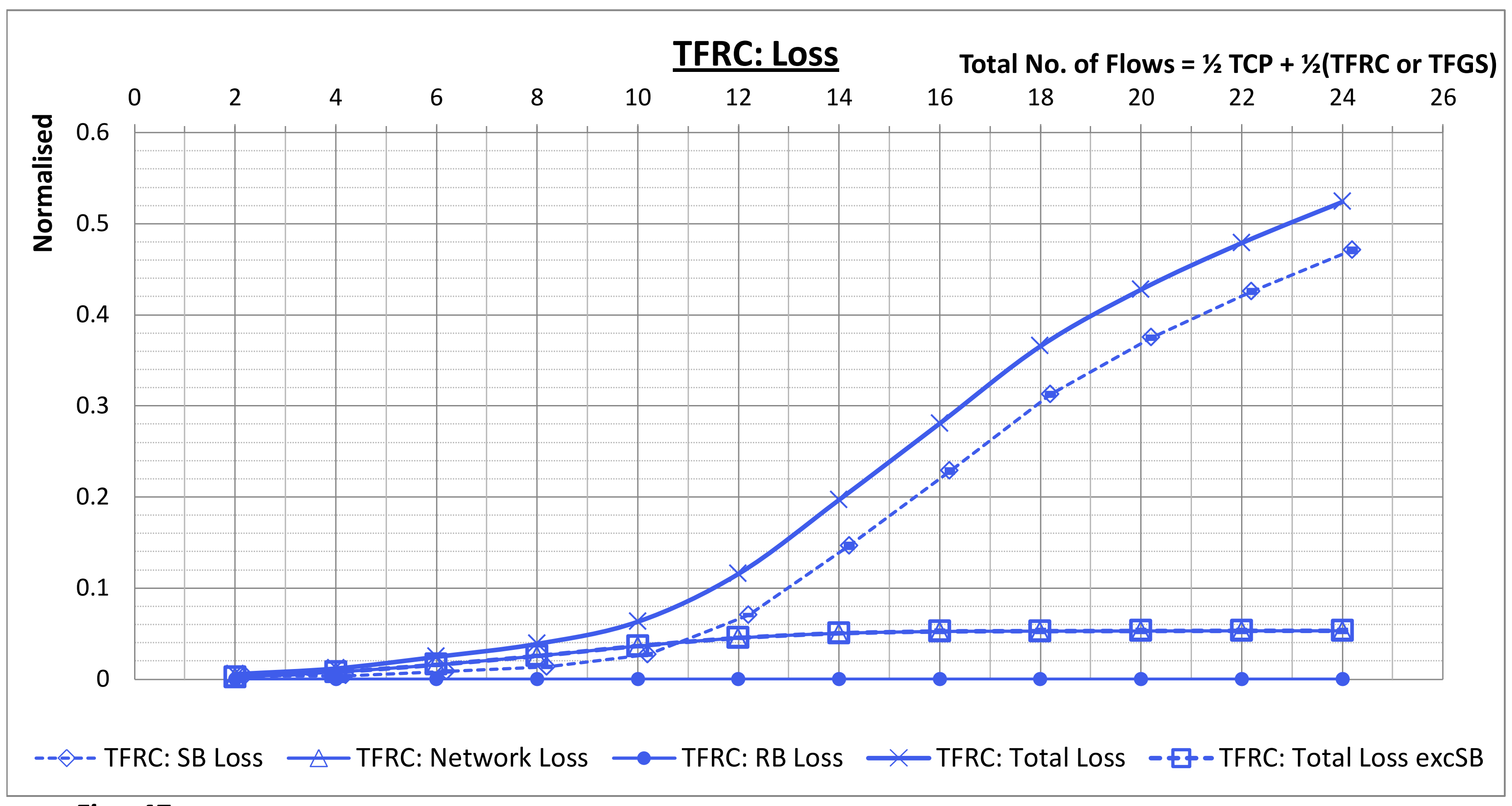


Fig. 47

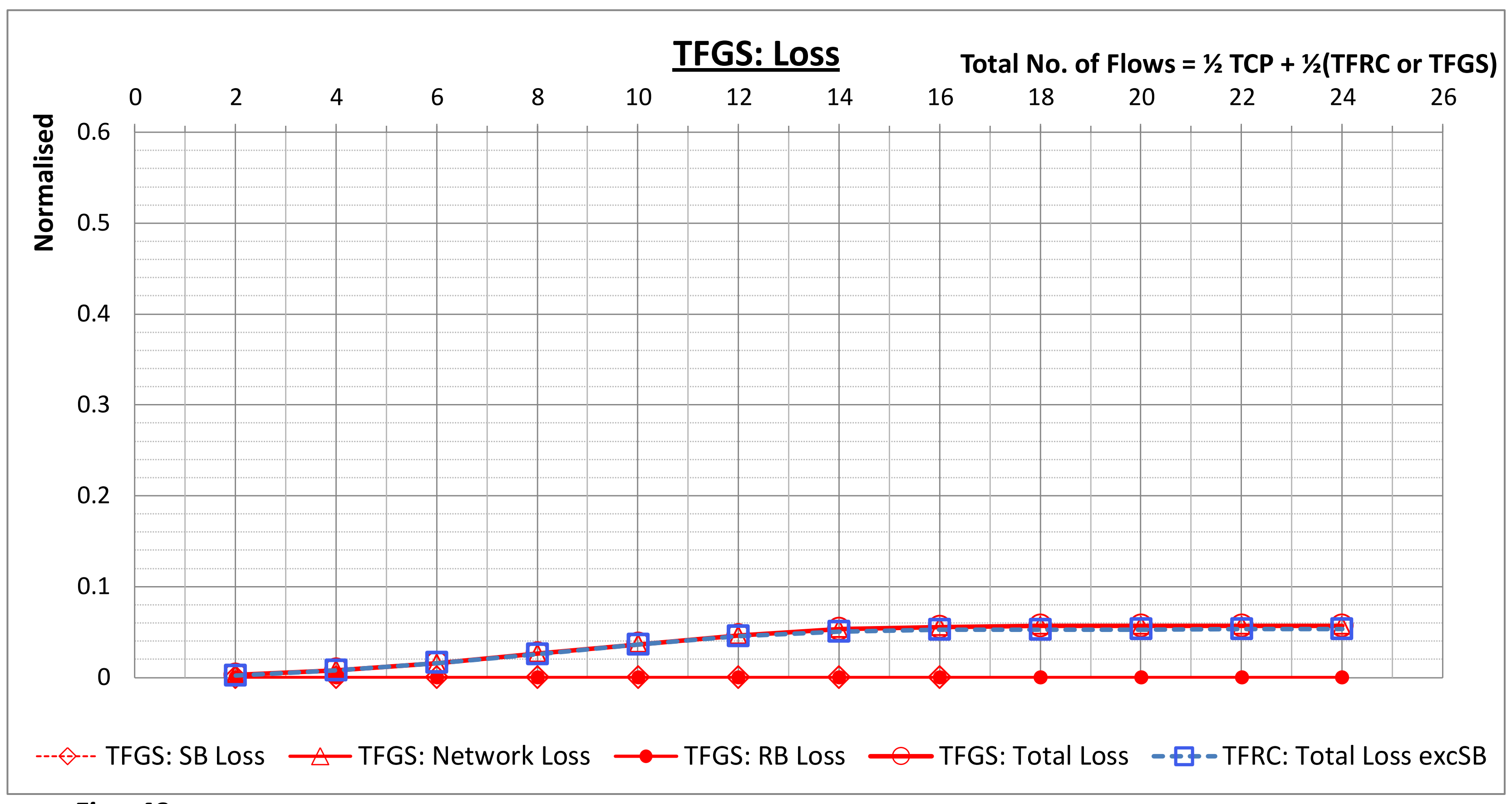


Fig. 48

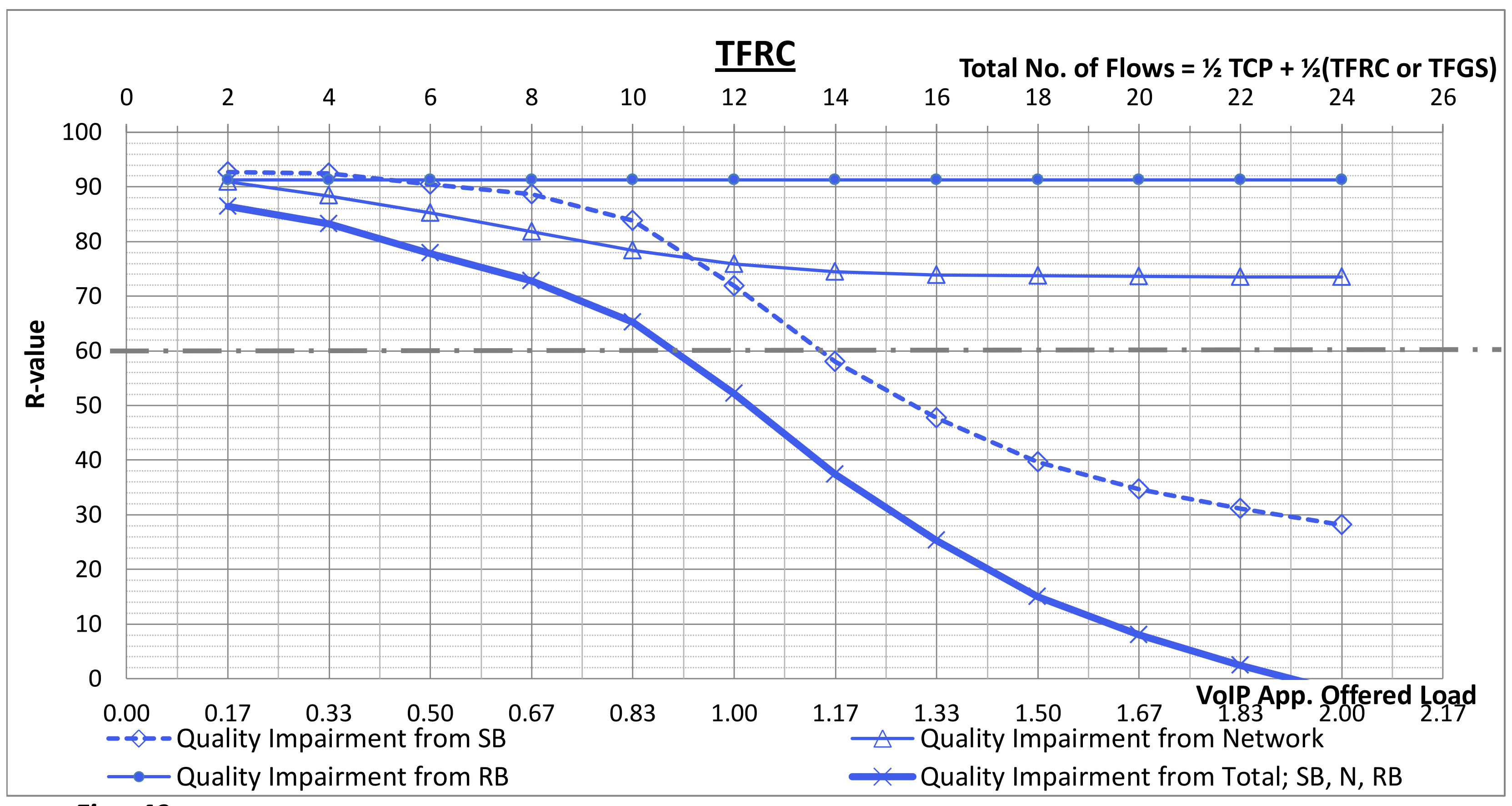


Fig. 49

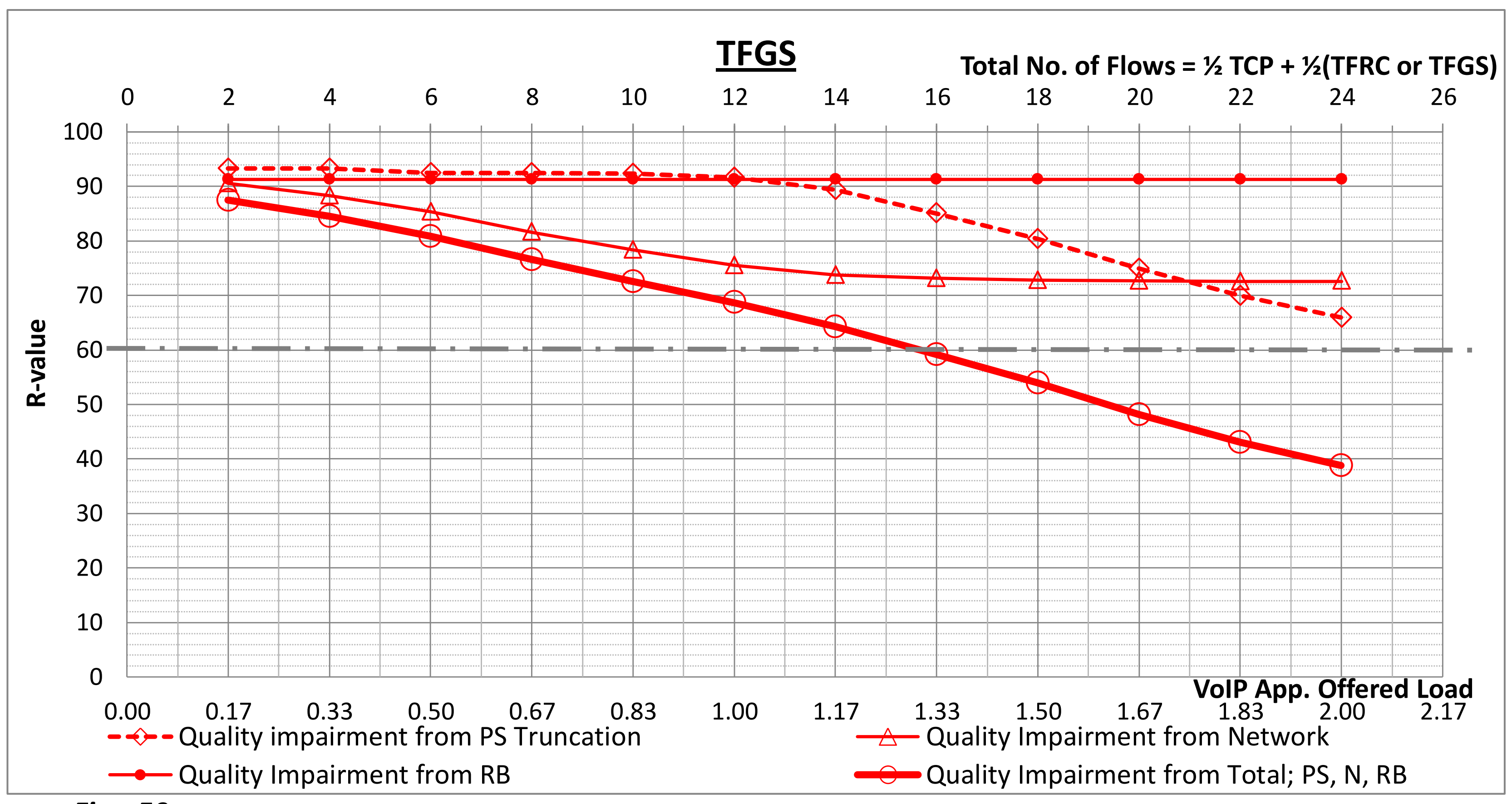


Fig. 50

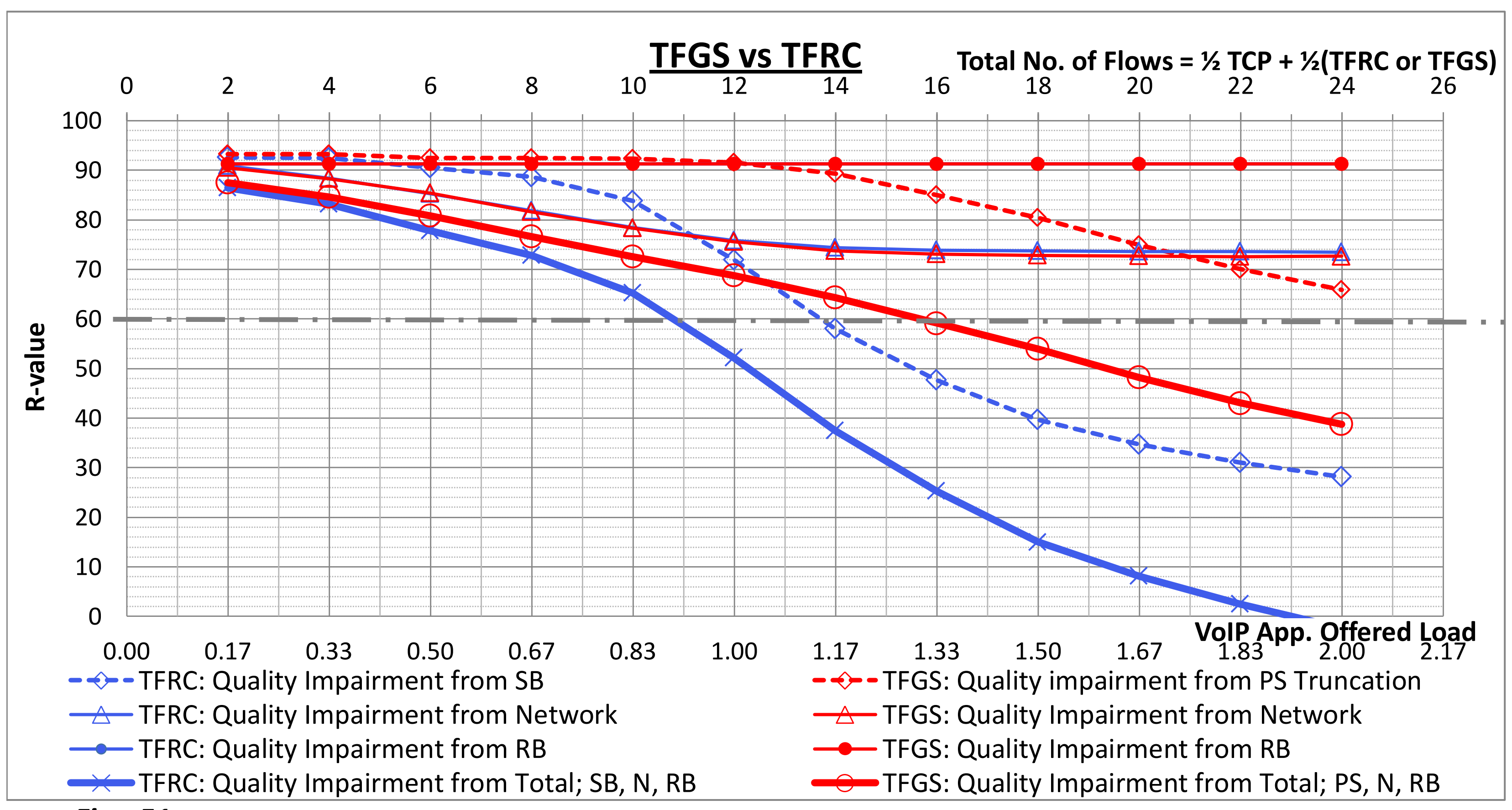


Fig. 51

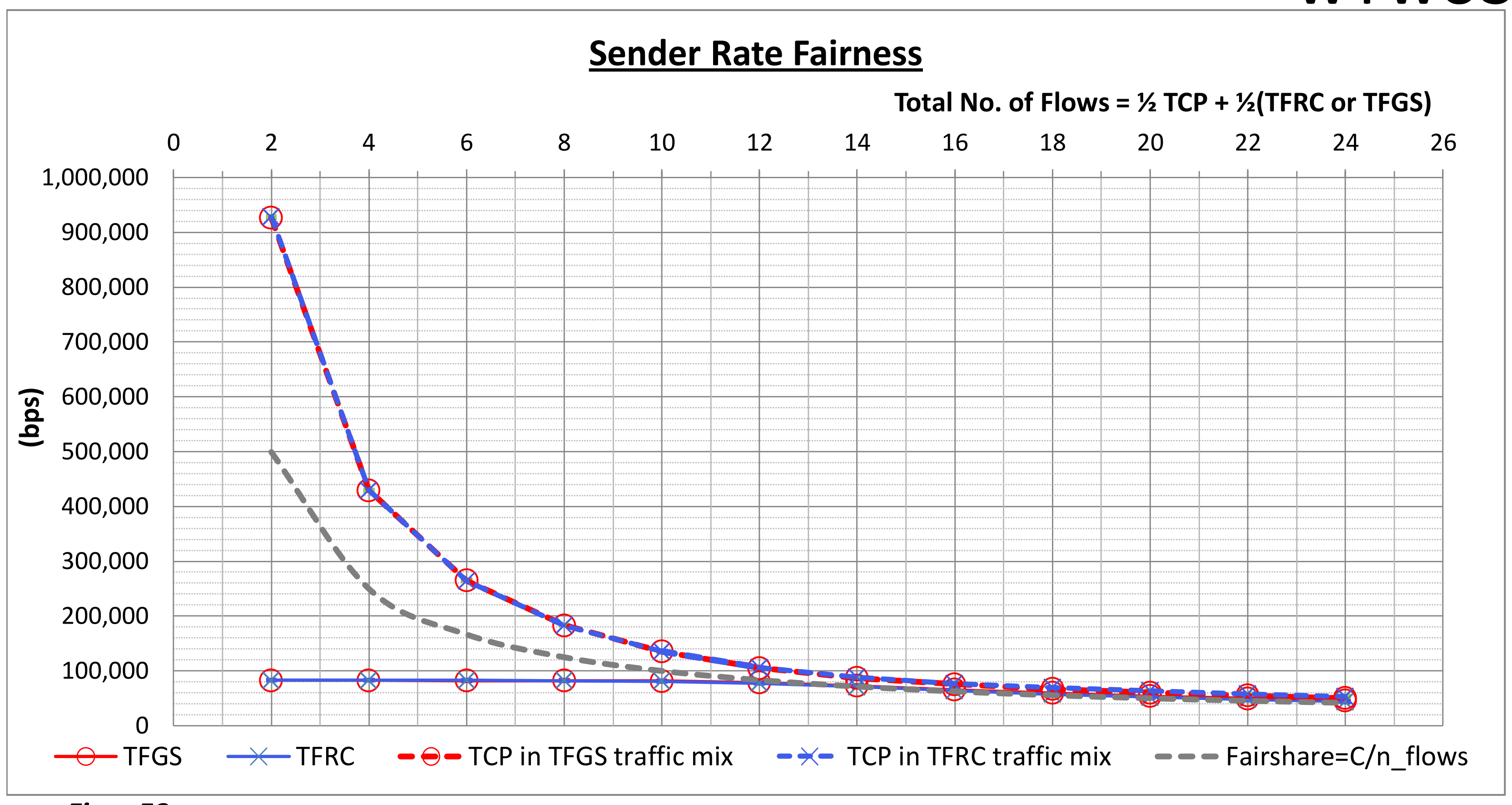


Fig. 52

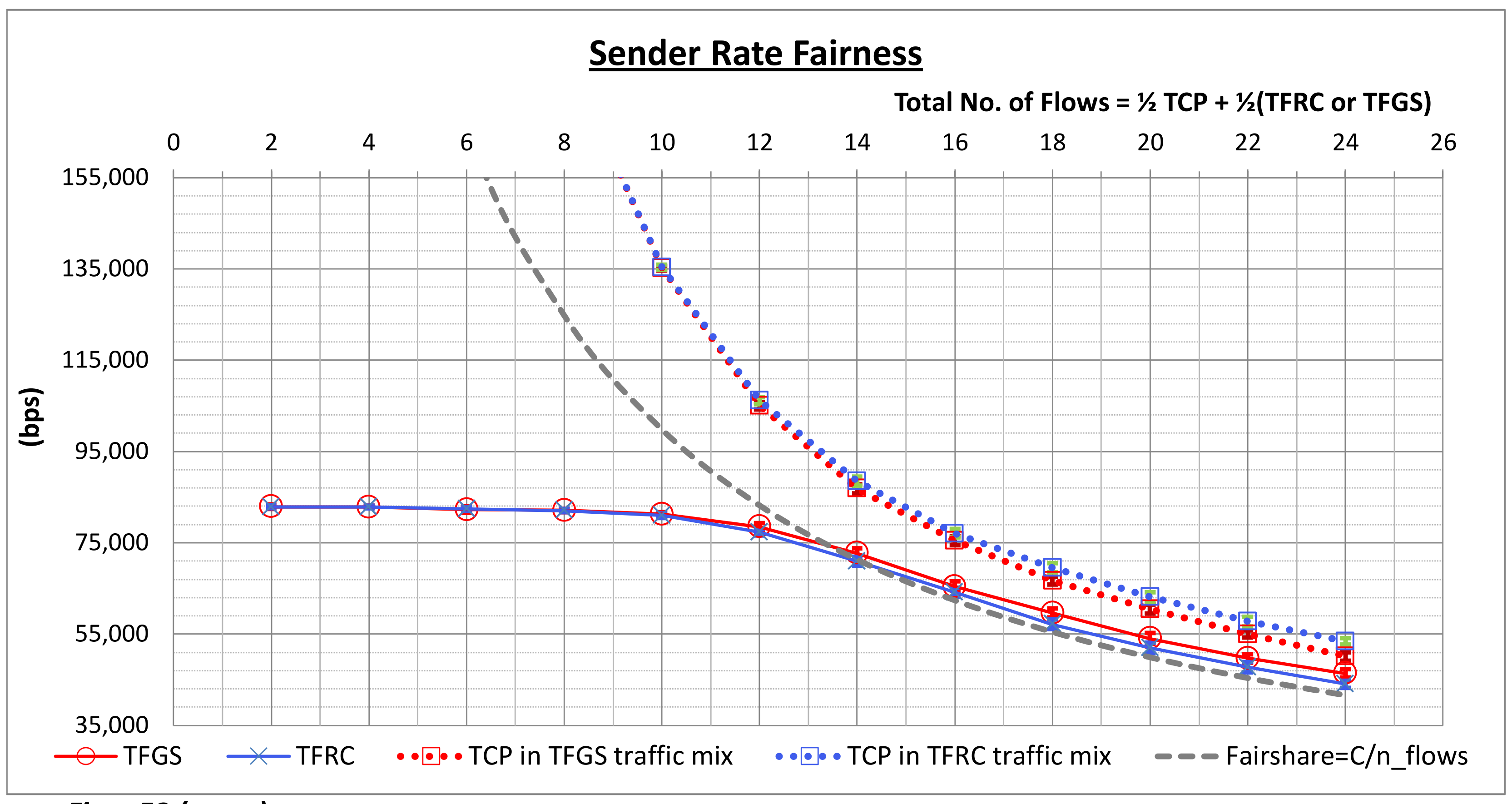


Fig. 52 (zoom)

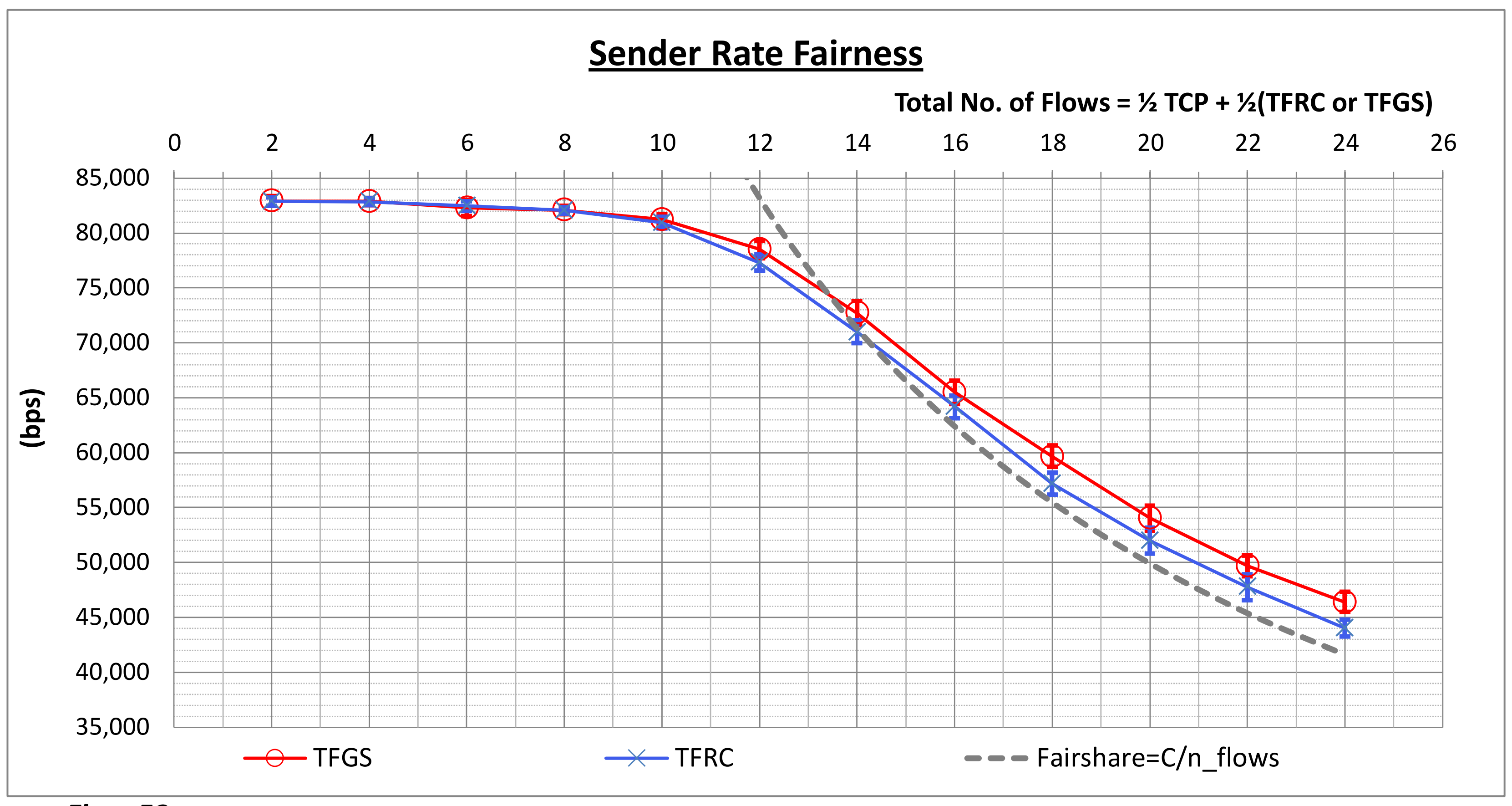


Fig. 53

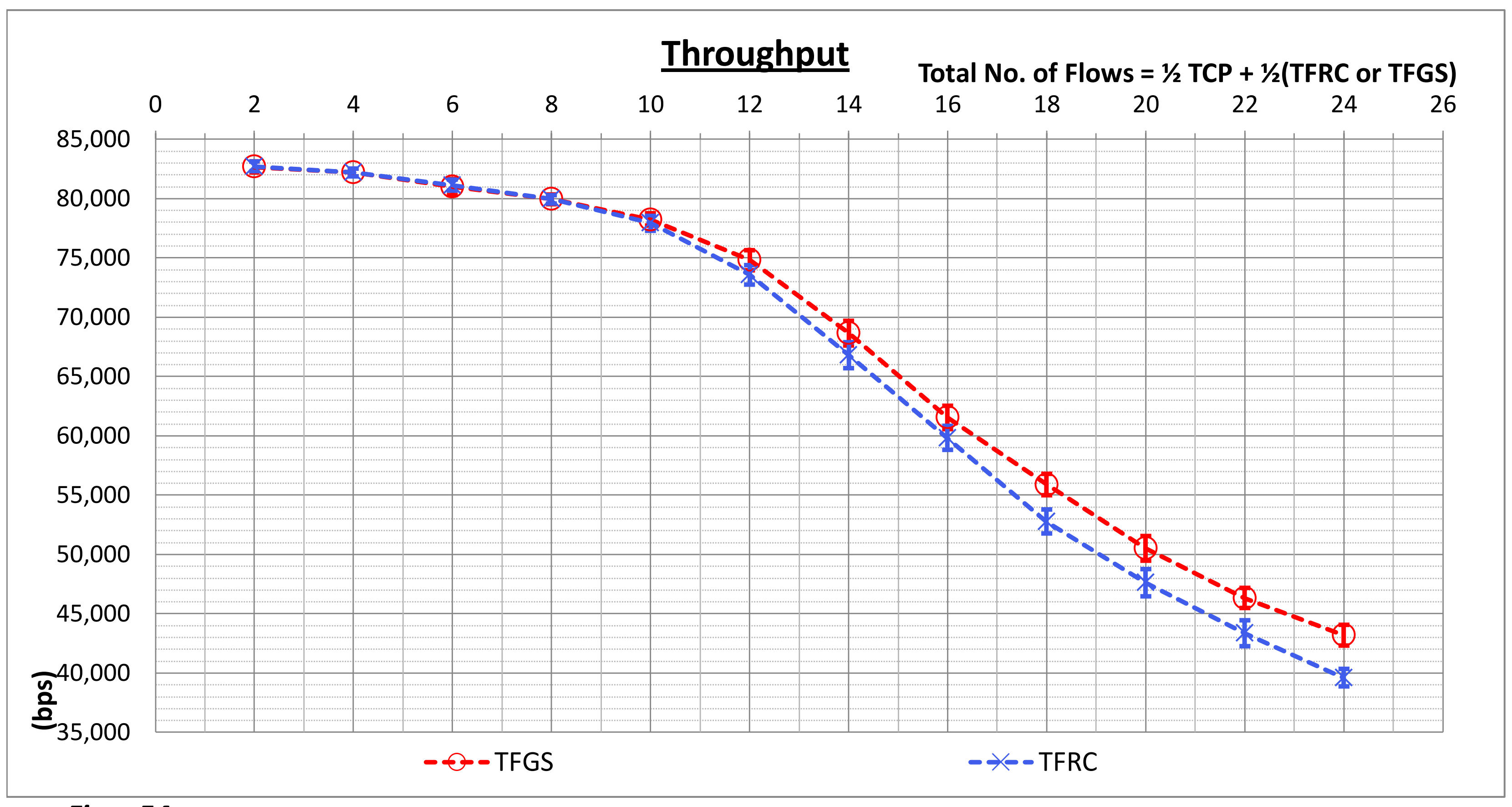


Fig. 54

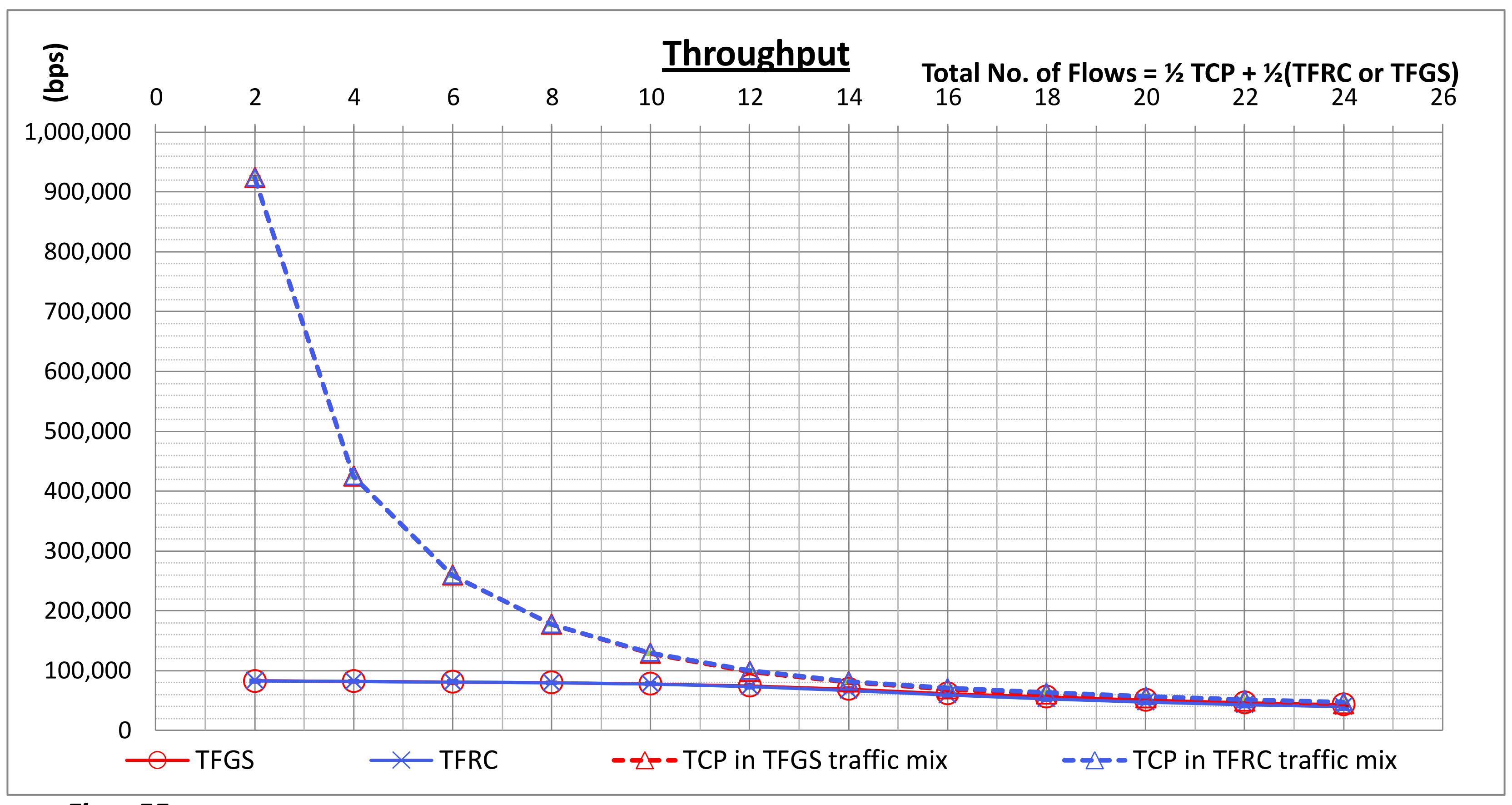


Fig. 55

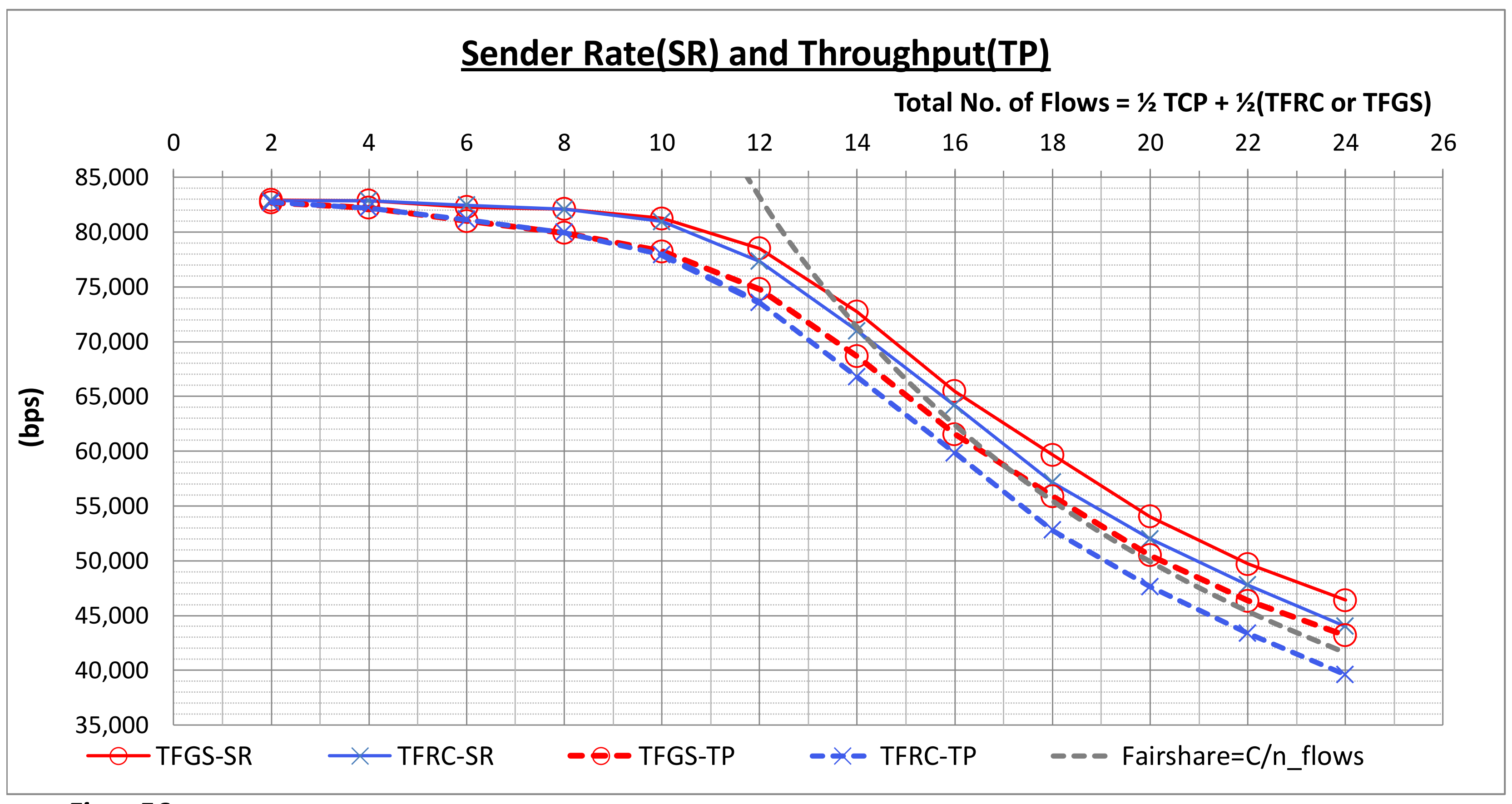


Fig. 56

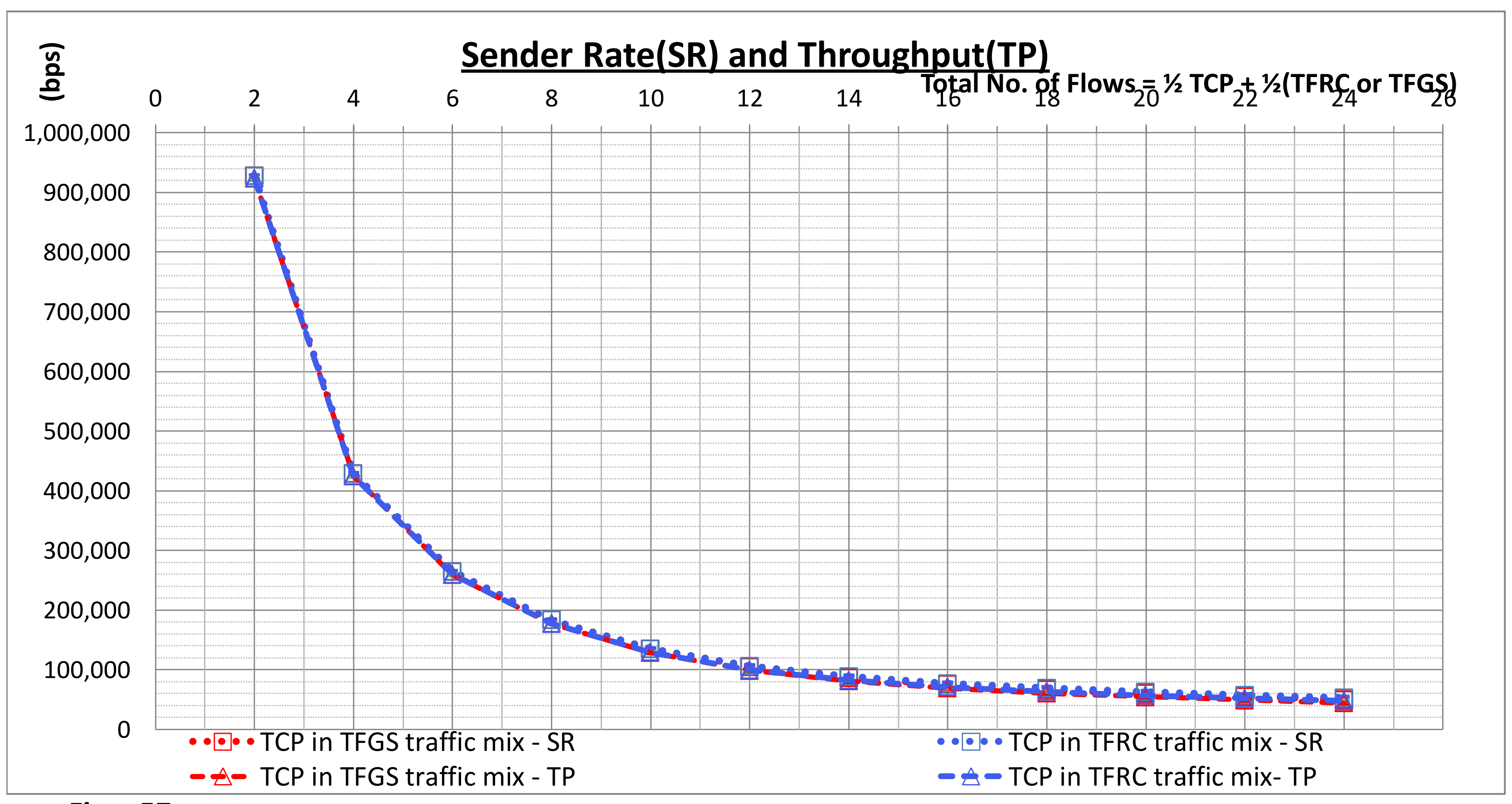


Fig. 57

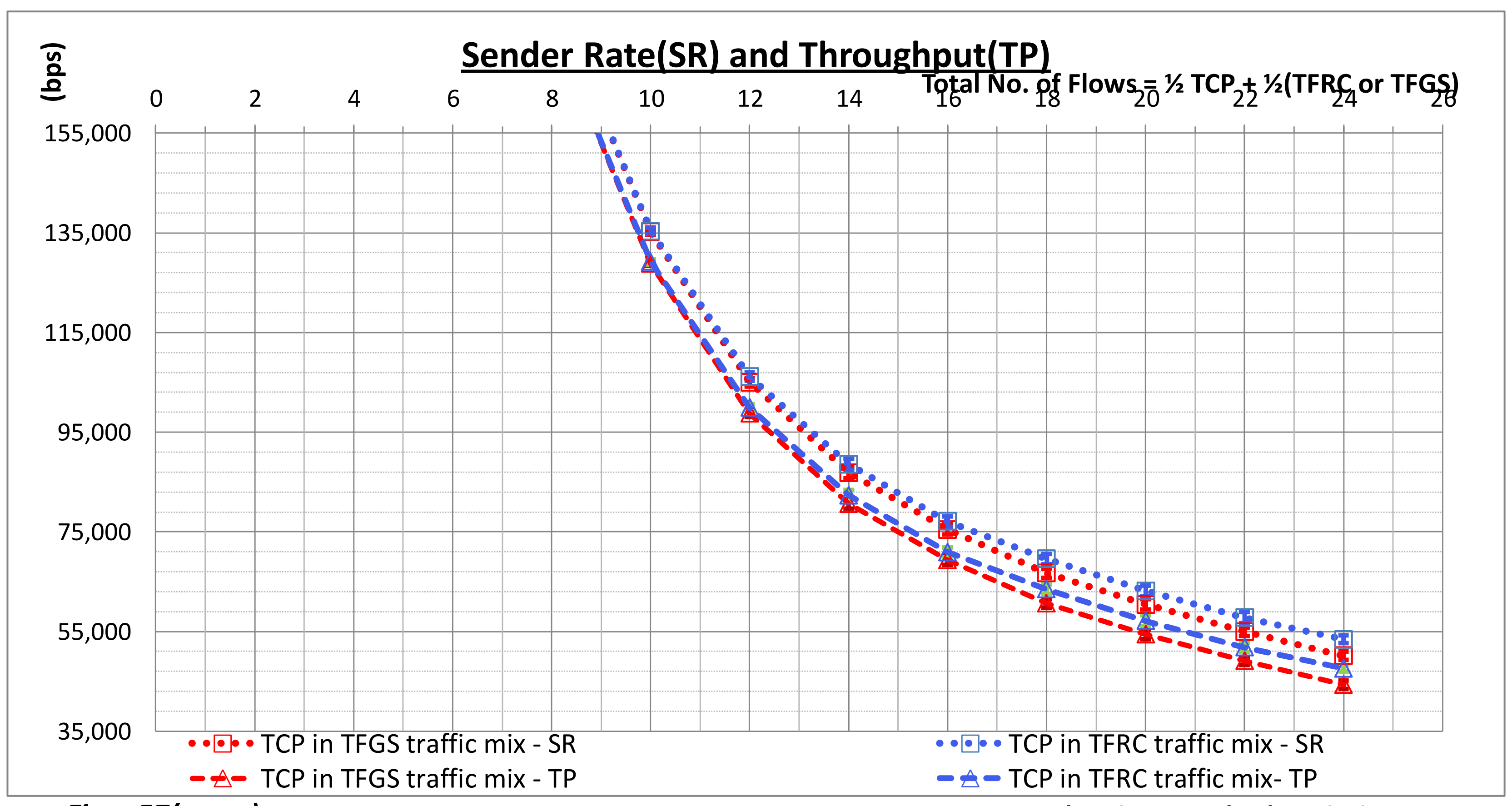


Fig. 57(zoom)

Bars showing Standard Deviation

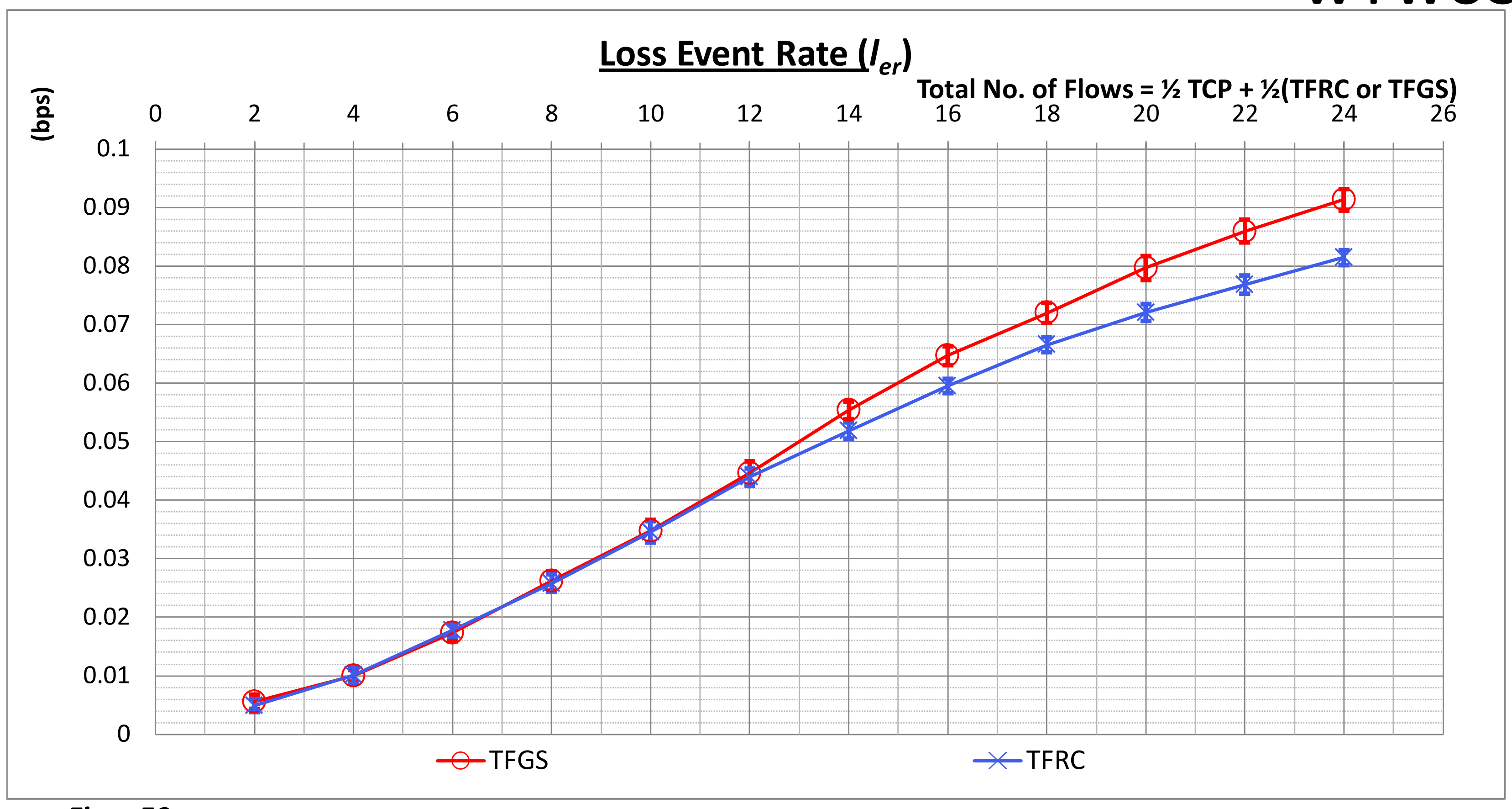


Fig. 59

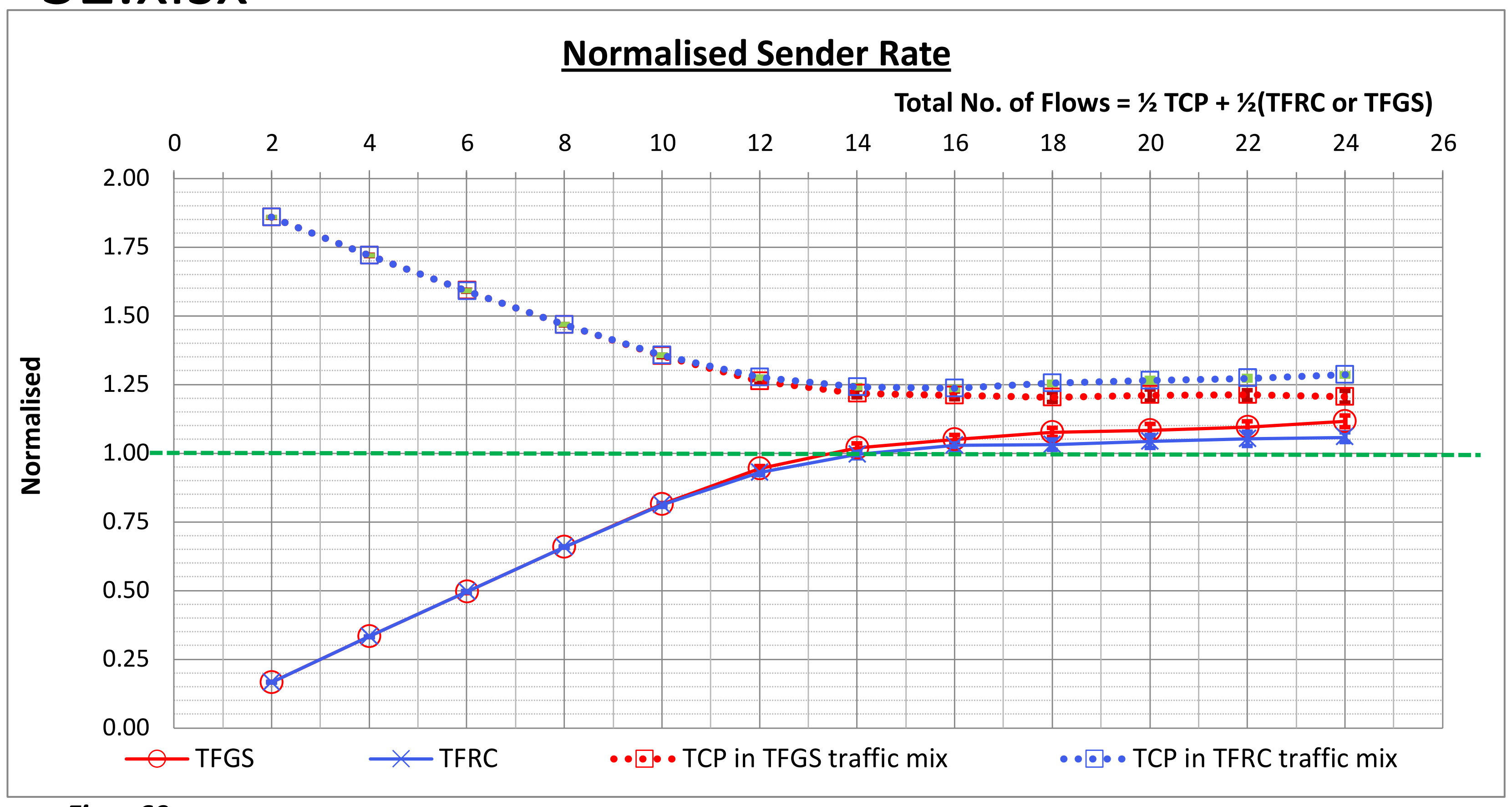


Fig. 60

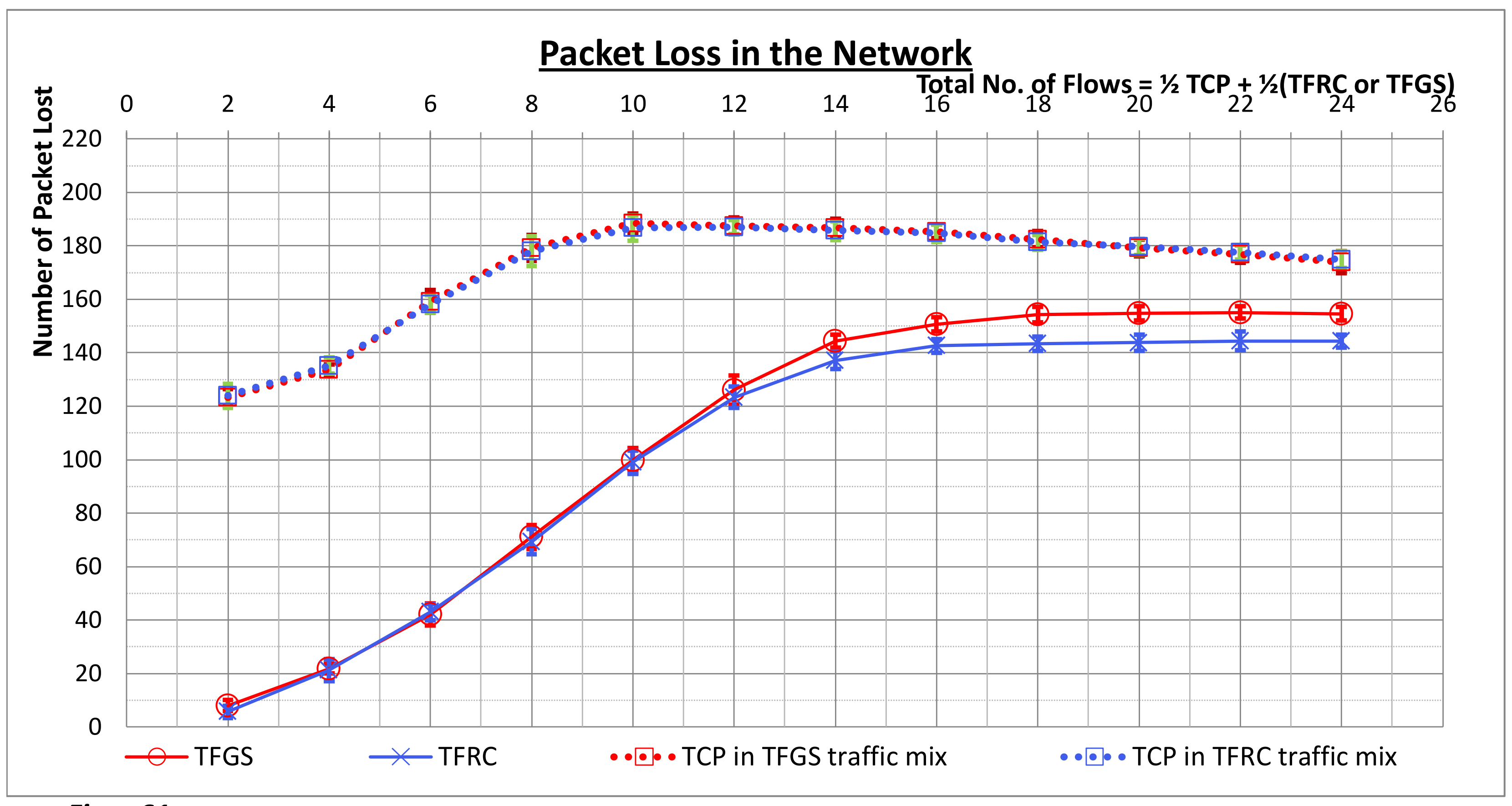


Fig. 61

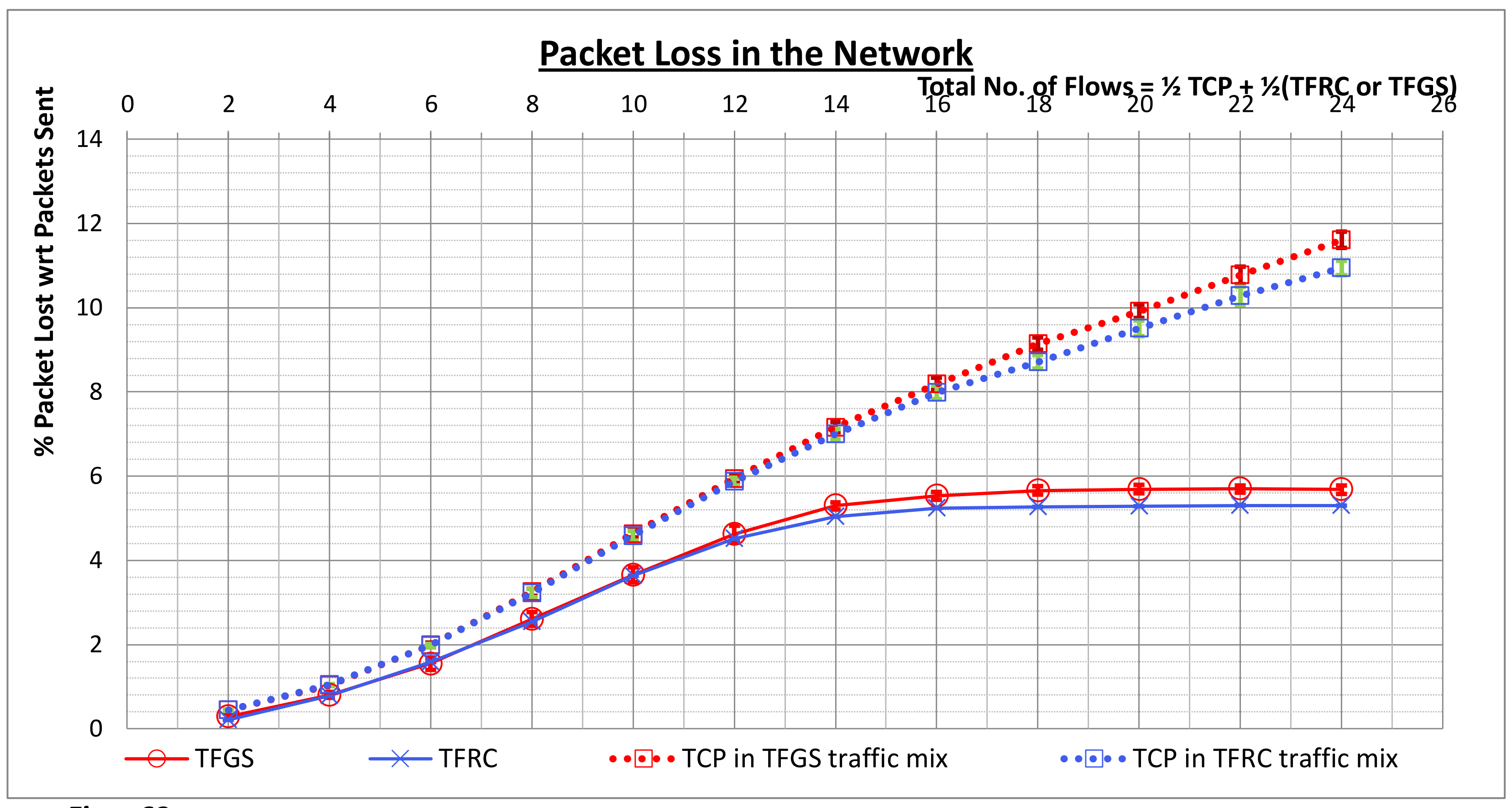


Fig. 62

11. Appendix III, Simulation Runs

11.1. Calculating mean[10] Delay

For example calculating the mean Network Delay, $\widehat{N_{D_S}}$, across 'n' simulation runs, and taking into account all packets[11] is shown in the equations below. This method is illustrated in Figure 11.1.

$$\widehat{\boldsymbol{N}_{\boldsymbol{D}_{f,k}}} = \frac{\mathbf{1}}{\boldsymbol{n}_f}\sum_{P=1}^{P=n_f} \boldsymbol{N}_{\boldsymbol{D}_{\boldsymbol{P}_{f,k}}} \qquad \textit{Equation 11.1}$$

$$\widehat{\boldsymbol{N}}_{\boldsymbol{D}_{s,k}} = \frac{\mathbf{1}}{\boldsymbol{m}}\sum_{f=1}^{f=m} \widehat{\boldsymbol{N}_{\boldsymbol{D}_{f,k}}} \qquad \textit{Equation 11.2}$$

$$\widehat{\boldsymbol{N}_{\boldsymbol{D}_s}} = \frac{\mathbf{1}}{\boldsymbol{r}}\sum_{s=1}^{s=r} \widehat{\boldsymbol{N}}_{\boldsymbol{D}_{s,k}} \qquad \textit{Equation 11.3}$$

Where:

n = total number of packets *f = flow*

P = packet *k = flow set*

s = simulation number

r = total number of simulation runs

m = total number of flows in the flow set

[10] This mean is also referred to as the 'Batch Mean'.

[11] Note: To give an idea of the number of packets generated over a simulation study where the total number of flows were 30, made up of 15 TFGS flows and 15 TCP flows. And this simulation repeated 25 times, resulted in a total of 1.2 million packets generated, see Table 11.6.

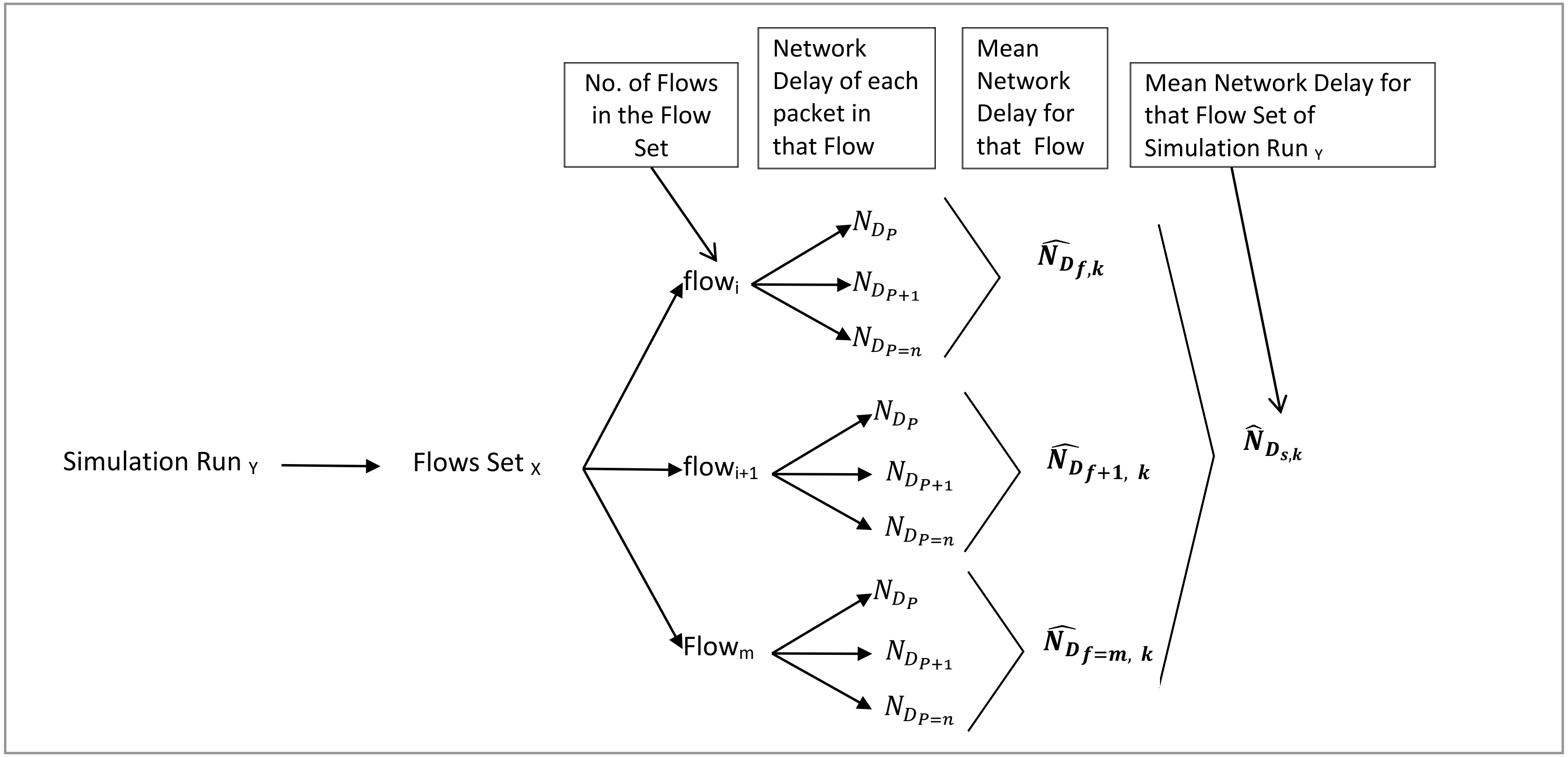


Figure 11.1, Calculating the mean Network Delay across ‘n’ simulation sets

11.2. Calculating mean[12] Loss

For example calculating the mean Network Loss, N_L, across 'n' simulation runs, and taking into account all packets is shown in the equations below. This method is illustrated below in Figure 11.2.

$\boldsymbol{N_{L_f}}$ *= No. of Packets Received* **–** *No. of Packets Sent* *from Equation 5.4*

$$\widehat{\boldsymbol{N}}_{\boldsymbol{L_{s,k}}} = \frac{\boldsymbol{1}}{\boldsymbol{m}} \sum_{f=1}^{f=m} \boldsymbol{N_{L_f}} \qquad \textit{Equation 11.4}$$

$$\widehat{\boldsymbol{N}_{\boldsymbol{L_s}}} = \frac{\boldsymbol{1}}{\boldsymbol{r}} \sum_{s=1}^{s=r} \widehat{\boldsymbol{N}}_{\boldsymbol{L_{s,k}}} \qquad \textit{Equation 11.5}$$

Where:

L = Loss *f = flow* *k = flow set*

m = total number of flows in the flow set

s = simulation number

r = total number of simulation runs

[12] This mean is also referred to as the 'Batch Mean'.

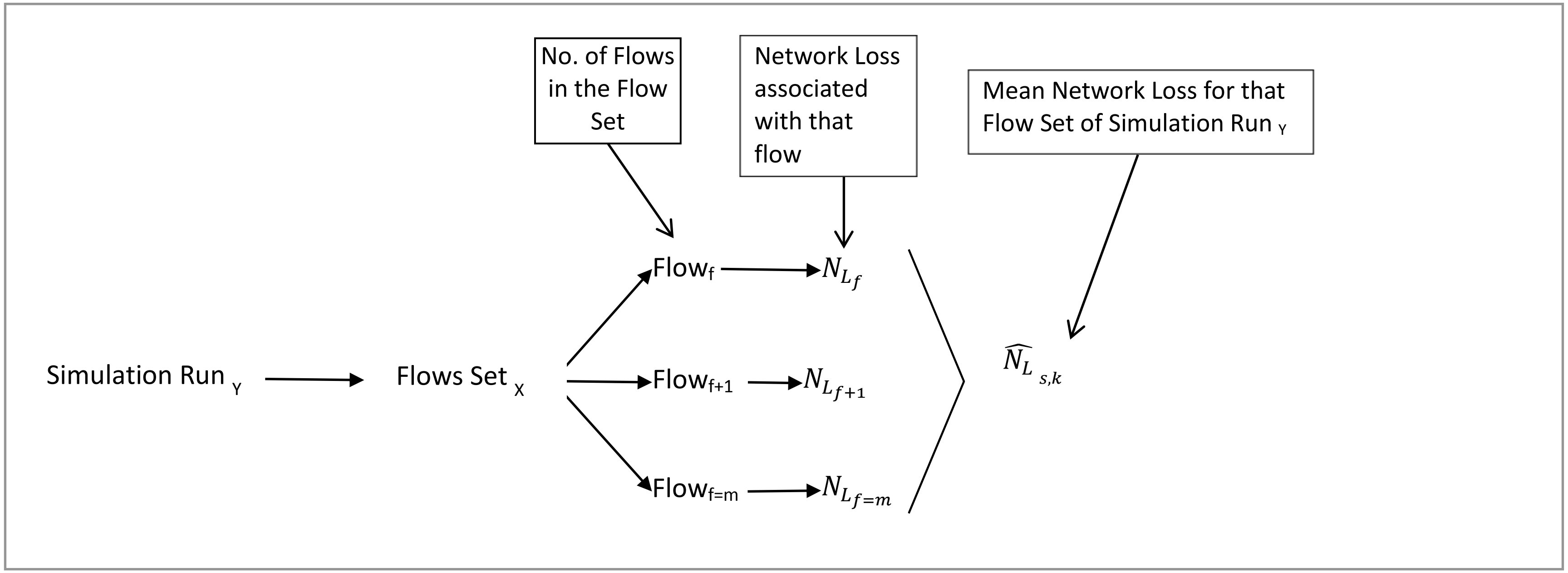


Figure 11.2, Calculating the mean Network Loss across 'n' simulation sets

11.3. Calculating Standard Deviation (STDEV) and Confidence Interval (CI)

This simulation was repeated 25 times with a randomised seed in order to give a batch mean of each simulation, X_Y. These batch means were used to calculate:

a) The standard deviation (STDEV), i.e. a measure of how far all the batch means (of each simulation run) deviate from the overall batch mean of all Simulation Runs, $\hat{X}$.

b) Confidence interval (CI), i.e. a measure of what percentage of the data set is within a given distance from the overall batch mean, $\hat{X}$.

The **STDEV**, σ, is calculated as follows:

$$\boldsymbol{\sigma = \sqrt{\frac{1}{n-1}\sum_{Y=1}^{n}(X_Y - \hat{X})^2}} \qquad \textit{Equation 11.6}$$

A **95 percent CI** of the mean, $\hat{X}$, value is calculated as follows:

$$\boldsymbol{CI_{95\%} = \hat{X} \pm 1.96\frac{\sigma}{\sqrt{n}}} \qquad \textit{Equation 11.7}$$

Where:

n = number of simulation runs

X_Y *= mean of Simulation* $_Y$*, e.g.* $\boldsymbol{\hat{N}_{D_{s,k}}}$

$\hat{X}$ *= mean of all Simulation Runs, e.g.* $\boldsymbol{N_{D_s}}$

TFRC							
	Units	**Total No. of Flows**	**No. of Simulation Runs**	**(overall Batch) Mean[1]**	**Effective Sample Size of (overall Batch) Mean**	**Standard Deviation[2]**	**95% Confidence Interval**
$IPGS_S$	secs	8	25	2.56E-02	468,042	8.4E-05	3.3E-05
P_S	Bytes	8	25	208.00	468,042	0.00	0.00
SR	Bits per sec (bps)	8	25	65,309.16	468,042	132.40	51.90
Queue Size	Bytes	8	25	4,580.251	25	1,273.872	4.758
	(equivalent) Packets	8	25	22.020	25	6.124	0.023

Table 11.1, TFRC: Batch Means, STDEV, CI

TFGS							
	Units	**Total No. of Flows**	**No. of Simulation Runs**	**(overall Batch) Mean[1]**	**Effective Sample Size of (overall Batch) Mean**	**Standard Deviation[1]**	**95% Confidence Interval**
$IPGS_S$	secs	8	25	0.02	596,100	0.00	0.00
P_S	Bytes	8	25	164.326	596,100	0.291	0.114
SR	Bits per sec (bps)	8	25	65,737.05	596,100	115.98	45.46
Queue Size	Bytes	8	25	4,591.390	25	1,497.533	3.909
	(equivalent) Packets	8	25	22.074	25	7.200	0.019

Table 11.2, TFGS: Batch Means, STDEV, CI

[1] $\hat{X}$ *= mean of all Simulation Runs, e.g.* $\boldsymbol{N_{D_s}}$

[2] The Standard Deviation is calculated from the set of Batch Mean values from 'N' Simulation Runs. The Total Number of Batch Mean values are 25, equal to the number of Simulation Runs. Further details on how the Standard Deviation is calculated can be seen in section 11.3.

11.4. Number of Packets[13] Generated by the Simulator

11.4.1. Homogenous Traffic Scenario

TFRC Flows				
No. of Simulations Runs	**Flow Set**	**No. of TFRC Flows**	**VoIP Offered Load**	**Sample Size Across all Simulation Runs**
25	1	1	0.17	74,950
25	2	2	0.33	149,775
25	3	3	0.50	224,475
25	4	4	0.67	299,050
25	5	5	0.83	373,500
25	6	6	1.00	447,825
25	7	7	1.17	461,588
25	8	8	1.33	468,042
25	9	9	1.50	476,599
25	10	10	1.67	481,660
25	11	11	1.83	488,876
25	12	12	2.00	499,447

Table 11.3, TFRC flows: Sample Size in Homogenous Traffic Scenario

[13] This is also referred to as the 'Sample Size'; indicating the number of samples from which the overall Batch Mean is calculated from.

TFGS Flows				
No. of Simulations Runs	Flow Set	No. of TFGS Flows	VoIP Offered Load	Sample Size Across all Simulation Runs
25	1	1	0.17	74,950
25	2	2	0.33	149,775
25	3	3	0.50	224,475
25	4	4	0.67	299,050
25	5	5	0.83	373,500
25	6	6	1.00	447,825
25	7	7	1.17	522,025
25	8	8	1.33	596,100
25	9	9	1.50	670,050
25	10	10	1.67	743,875
25	11	11	1.83	818,700
25	12	12	2.00	893,525

Table 11.4, TFGS flows: Sample Size in Homogenous Traffic Scenario

11.4.2. Heterogeneous Traffic Scenario

Table 11.5, TFRC & TCP flows: Sample Size in Heterogeneous Traffic Scenario

TFRC Flows					TCP flows in the TFRC mix					TFRC and TCP Heterogeneous Traffic Mix
No. of Simulations Runs	Flow Set	No. of TFRC Flows	VoIP Offered Load	Sample Size Across all Simulation Runs	No. of Simulations Runs	Flow Set	No. of TFRC Flows	VoIP Offered Load	Sample Size Across all Simulation Runs	Total No. of Packets Generated by the Simulator for each Flow Set
25	2	1	0.17	68,465	25	2	1	0.17	698,203	766,668
25	4	2	0.33	136,753	25	4	2	0.33	641,743	778,496
25	6	3	0.50	203,944	25	6	3	0.50	587,303	791,247
25	8	4	0.67	270,417	25	8	4	0.67	534,714	805,131
25	10	5	0.83	333,108	25	10	5	0.83	486,687	819,795
25	12	6	1.00	381,435	25	12	6	1.00	451,364	832,799
25	14	7	1.17	408,168	25	14	7	1.17	433,620	841,788
25	16	8	1.33	421,362	25	16	8	1.33	426,430	847,792
25	18	9	1.50	421,793	25	18	9	1.50	429,159	850,952
25	20	10	1.67	425,911	25	20	10	1.67	428,783	854,694
25	22	11	1.83	430,564	25	22	11	1.83	428,010	858,574
25	24	12	2.00	433,050	25	24	12	2.00	429,059	862,109

Table 11.6, TFGS & TCP flows: Sample Size in Heterogeneous Traffic Scenario

TFGS Flows					TCP flows in the TFGS mix					TFGS and TCP Hetrogenous Traffic Mix
No. of Simulations Runs	Flow Set	No. of TFGS Flows	VoIP Offered Load	Sample Size Across all Simulation Runs	No. of Simulations Runs	Flow Set	No. of TFGS Flows	VoIP Offered Load	Sample Size Across all Simulation Runs	Total No. of Packets Generated by the Simulator for each Flow Set
25	2	1	0.17	68,700	25	2	1	0.17	698,038	766,738
25	4	2	0.33	137,275	25	4	2	0.33	641,748	779,023
25	6	3	0.50	205,725	25	6	3	0.50	587,491	793,216
25	8	4	0.67	274,050	25	8	4	0.67	534,673	808,723
25	10	5	0.83	342,250	25	10	5	0.83	485,611	827,861
25	12	6	1.00	410,325	25	12	6	1.00	446,192	856,517
25	14	7	1.17	478,275	25	14	7	1.17	424,634	902,909
25	16	8	1.33	546,100	25	16	8	1.33	416,795	962,895
25	18	9	1.50	613,800	25	18	9	1.50	409,808	1,023,608
25	20	10	1.67	681,375	25	20	10	1.67	408,822	1,090,197
25	22	11	1.83	749,950	25	22	11	1.83	405,417	1,155,367
25	24	12	2.00	818,525	25	24	12	2.00	399,436	1,217,961

11.5. Random Number Generators (RNG)

The RNG used in *ns*2 is the combined multiple recursive generator called MRG32k3a proposed by L'Ecuyer [73]. MRG32k3a is known to have satisfactory uniformity in the random numbers generated [74, 75]. The period is 3.1 × 1057 which can provide ample values (without repeating sequences) during multiple simulation runs [76].

In *ns*2, it is not normally necessary to explicitly set the seeds as this is done automatically, further details can be found in section 24 in the *ns*2 documentation [77].

References

[1] S. Floyd, J. Kemp, "IAB Concerns Regarding Congestion Control for Voice Traffic in the Internet", March 2004, Request for Comments: 3714, pages 11,14,15

[2] S. Floyd, “Congestion Control Principles”, Best Current Practice BCP 41 (RFC 2914) IETF, September 2000

[3] Sally Floyd, Kevin Fall, "Promoting the Use of End-to-End Congestion Control", IEEE/ACM Transactions on Networking, Volume 7, May 1999

[4] ISO/IEC International Standard 13818 (MPEG-2), Information technology - Generic coding of moving pictures and associated audio information: Systems, 2007

[5] Reza Rejaie, Mark Handley, Deborah Estrin, "Layered Quality Adaptation for Internet Video Streaming", IEEE Journal, Vol 18, No 12, pp. 2530–2543, Dec 2000.

[6] Jérôme Viéron and Christine Guillemot, "Real-Time Constrained TCP-Compatible Rate Control for Video Over the Internet", IEEE Transactions on Multimedia, Vol. 6, No. 4, Aug 2004

[7] Mark Sandler & Dawn Black, “Scalable Audio Coding for Compression and Loss Resilient Streaming”, IEE Proc. Vision, Image & Signal Processing 2005, Queen Mary University of London, Vol. 153, No. 3, June 2006, pp331-339

[8] H. Schulzrinne, S. Casner, “RTP Profile for Audio and Video Conferences with Minimal Control”, RFC 3551, IETF, July 2003

[9] S. Floyd, M. Handley, J. Padhye, and J. Widmer. “Equation-Based Congestion Control for Unicast Applications: the Extended Version”, International Computer Science Institute tech report TR-00-003, March 2000 www.aciri.org/tfrc/

[10] Sally Floyd, “Metrics for the Evaluation of Congestion Control Mechanisms”, IETF, RFC 5166, March 2008

[11] M. Handley, S. Floyd, J. Padhye and J. Widmer, "TCP Friendly Rate Control (TRFC): Protocol Specification", January 2003, IETF, RFC 3448.

[12] R. Rejaie, M. Handley, and D. Estrin, “RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet,” in Proc. IEEE Infocom’99, March 1999, vol 3, pp. 1337 – 1345, New York.

[13] ITU-T Recommendation G.107, "The E-model, a computational model for use in transmission planning", December 1998

[14] Uyless Black , Voice Over IP, Prentice Hall, 1 edition, 4 Aug 1999, ISBN 0130224634, pp. 25-26, 39-41

[15] Antony Oodan, Keith Ward, Catherine Savolaine, Mahmoud Daneshmand, Peter Hoath, “Telecommunications Quality of Service Management: From Legacy to Emerging Services”, Publisher: The Institution of Engineering and Technology; [New edition (December 1, 2002), pp. 180-90, ISBN-10: 0852964242

[16] “Methods for subjective determination of transmission quality,” ITU-T Recommendation P.800, Aug. 1996.

[17] "Definition of categories of speech transmission quality," ITU-T Recommendation G.109, Sept. 1999.

[18] R.G. Cole, J.H. Rosenbluth, Voice over IP performance monitoring, ACM SIGCOMM Computer Communication Review, Volume 31, Issue 2, AT&T Laboratories, 2001

[19] A. Watson and M. Sasse, "Measuring perceived quality of speech and video in multimedia conferencing applications," in Proc. ACM Multimedia Conference, pp. 55-60, 1998, Bristol, UK, Sept. 12–16, 1998.

[20] Perkins, C. and O. Hodson, "Options for Repair of Streaming media", RFC 2354, June 1998.

[21] S. Floyd, M. Handley, and J. Padhye, "A Comparison of Equation-based and AIMD Congestion Control", May 2000, pages 1-3

[22] J. Padhye, J. Kurose, D. Towsley, and R. Koodli. "A Model Based TCP-Friendly Rate Control Protocol". In Proceedings of NOSSDAV 1999.

[23] Eddie Kohler, Mark Handley, Sally Floyd, "Datagram Congestion Control Protocol (DCCP)", IETF, RFC 4340, March 2006

[24] Sally Floyd, Eddie Kohler, Jitendra Padhye, ICIR, UCLA, " Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control", IETF, RFC 4341, March 2006.

[25] The Datagram Congestion Control Protocol (DCCP) Working Group at the IETF, http://datatracker.ietf.org/wg/dccp/charter

[26] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, L. Zhang, " Recommendations on Queue Management and Congestion Avoidance in the Internet", IETF RFC 2309, April 1998

[27] S. Floyd, "Deployment of RED", www.icir.org/floyd/red.html#experiences

[28] W. Willinger, M. S. Taqqu, R. Sherman, D. V. Wilson, Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level, ACM SIGCOMM '95, pp. 100- 113, August 1995.

[29] Jacobson V., Congestion Avoidance and Control, Proceedings of SIGCOMM August 1988, pp.314-329

[30] "The TCP-Friendly Web Page". www.psc.edu/networking/tcp friendly.html

[31] M. Mathis, J. Semke, J. Mahdavi, and T. Ott., "The macroscopic behaviour of the TCP congestion avoidance algorithm". Computer Communication Review, 27(3), ACM SIGCOMM Computer Communication Review, v.27 n.3, p.67-82, July 1997

[32] Olivier Hersent, IP Telephony: Packet Based Multimedia Communications Systems, Publisher: Addison Wesley; 1 edition (6 Jan 2000), ISBN-10: 0201619105

[33] J-C. Bolot and Andreas Vega-Garcia, "Control Mechanisms for Packet Audio in the Internet", Proceedings of IEEE Infocom 1996, pp 232-239, ftp://ftp-sop.inria.fr/rodeo/bolot/96.Audio_ctl.ps.gz

[34] C. Perkins, O. Hodson and V. Hardman, “A Survey of Packet Loss Recovery Techniques for Streaming Audio”, IEEE Network Magazine, Sept/Oct 1998, pp 40-47

[35] N. Shacham, P. McKenney, “Packet Recovery in High-Speed Networks using Coding and Buffer Management”, Proc. IEEE Infocom 1990

[36] Jim Kurose, Keith Ross, “Computer Networking, a top down approach featuring the Internet”, Third Edition, 2005, Pearson/Addison Wesley Publications, ch. 7, pp. 565-652, ISBN 0321269764

[37] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, “Adaptive playout mechanisms for packetized audio applications in wide-area networks,” in Proc. IEEE INFOCOM, pp. 680 – 688, vol.2, Toronto, Canada, June 1994

[38] M. Ghanbari , Video Coding on Introduction to standard codes, The IEE Book Series, ISBN 085296 762 4, p131

[39] ISO/IEC International Standard 14496 (MPEG-4), Information technology - Coding of audio-visual objects, January 2000

[40] Marta Mrak, Mislav Grgic, Sonja Grgic, "Scalable Video Coding in Network Applications", IEEE Region 8 International Symposium on Video / Image Processing and Multimedia Communications, 16-19 June 2002, Zadar, Croatia

[41] Chris Dunn, "Efficient Audio Coding with Fine-Grain Scalability", Audio Engineering Society, Presented at the 111th Convention, 21-24 September 2001, New York USA, Scala Technology, London, UK

[42] Ben Leslie, Mark Sandler, "Packet Loss Resilient, Scalable Audio Compression and Streaming for IP Networks", 3G Mobile Communications

Technologies, 26-28 March 2001, IEE Conference Publication no. 477, pages 120, 122

[43] N. Sprljan, M. Mrak, G.C.K. Abhayaratne, E. Izquierdo, “A Scalable Coding Framework for Efficient Video Adaptation”, Proc. Int. Work. on Image Analysis for Multimedia Interactive Services (WIAMIS), April 2005

[44] Richard Akester & Mark Handley, “Z24: Multimedia Systems”, 2003/2004 Lecture notes, www.cs.ucl.ac.uk/teaching/Z24

[45] Jonathan Davidson, James Peters, Manoj Bhatia, Satish Kalidindi, Sudipto Mukherjee, "Voice Over IP Fundamentals", Ch. 7, pp. 146-167 ISBN-10: 1587052571, Publisher: Cisco Press; 2 edition (July 2006)

[46] Olivier Hersent, "IP Telephony: Packet Based Multimedia Communications Systems", Ch. 4, 210-227, ISBN-10: 0201619105, Publisher: Addison Wesley; 1 edition (6 Jan 2000)

[47] J.C. Mogul, S.E. Deering, "Path MTU discovery", RFC 1191, Nov 1990

[48] S. Hanks, T. Li, D. Farinacci, P. Traina, "Generic Routing Encapsulation over IPv4 networks", RFC1702, October 1994

[49] Prashant Anantha Krishnan, Sunil Kumar Derasriya, Quality of Service Management for Voice Over IP Networks, 12 April 2010, http://solomon.ipv6.club.tw/VoIP/Measurement/calculate_mos.ppt

[50] T. Friedman, Ed., R. Caceres, Ed., A. Clark, Ed., "RTP Control Protocol Extended Reports (RTCP XR)", RFC3611, Nov 2003

[51] T. Phelan, “Strategies for Streaming Media Applications Using TCP-Friendly Rate Control”, Sonus Networks, Internet Draft, October 2005

[52] M. Mathis, J. Semke, J. Madhavi, and T. Ott, “Macroscopic behavior of TCP congestion avoidance algorithm,” Computer Communication Review, vol. 27, no. 3, July 1997.

[53] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP throughput: A simple model and its empirical validation,” Proc. ACM SIGCOMM’98, pp. 1998.

[54] Mark Sandler, Dawn Black, “Loss Resilient, Scalable Audio Compression and Streaming”, European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies (EWIMT), 25 November 2004, London

[55] C. Demichelis, P. Chimento, IP Packet Delay Variation Metric for IP Performance Metrics, RFC 3393, Telecomitalia Lab, Ericsson IPI, IETF (2002)

[56] J.D. Neale, E.V. Jones, J.T. Zubrzycki, "MPEG Coded News Dispatches over GSM HSCSD", BBC R&D, University of Essex, IEEE GLOBECOM, pp. 1443 – 1448, vol.3, Nov 1998

[57] The network simulator, *ns2*, version 2.29, available at http://www.isi.edu/nsnam/ns

[58] P Vasallo, "Variable Packet Size Equation-Based Congestion Control", International Computer Science Institute Technical (ICSI), Technical Report TR-00-008, Report, May 2000

[59] S. Floyd, E. Kohler, “TCP Friendly Rate Control (TFRC): The Small-Packet (SP) Variant”, IETF RFC 4828, April 2007

[60] J. Widmer, C. Boutremans and J. Boudec, "End to end congestion control for flows with Variable Packet Sizes", ACM SIGCOMM Computer Communication Review Archive, Volume 34 , Issue 2, April 2004

[61] T. Phelan, "Datagram Congestion Control Protocol (DCCP) User Guide", April 2005, Internet Draft, Sonus Networks.

[62] B. Grill, “A Bit Rate Scalable Perceptual Coder for MPEG-4 Audio”, Preprint no. 4620, 103rd AES Convention, New York, 1997

[63] Understanding Delay in Packet Voice Networks, CISCO Voice Quality, Document ID: 5125, http://www.cisco.com/en/US/tech/tk652/tk698/technologies_white_paper09186a00800a8993.shtml

[64] VoIP Performance Management: Impact of Delay in Voice over IP Services January 2006, Telchemy Application Notes, http://www.telchemy.com/appnotes/Telchemy_Impact_of_Delay.pdf

[65] Sally Floyd, Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, Volume 1 Issue 4, pp. 397 – 413, August 1993

[66] Reza Rejaie, Mark Handley, Deborah Estrin, “RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet” – Technical Report, University of Southern California, July 1998, page 10.

[67] S. Floyd, "RED: Discussions of Setting Parameters", Nov 1997, www.icir.org/floyd/REDparameters.txt

[68] W. Eddy, M. Allman, “A Comparison of RED's Byte and Packet Modes”, Computer Networks Journel, volume 42, pp. 261-280, June 2003.

[69] Skype Fast Facts Q4 2008, http://ebayinkblog.com/wp-content/uploads/2009/01/skype-fast-facts-q4-08.pdf

[70] Luca De Cicco, Saverio Mascolo, Vittorio Palmisano, An Experimental Investigation of the Congestion Control used by Skype VoIP, WWIC Conference 2007, Coimbra, Portugal, May, 2007

[71] T. Friedman, R. Caceres, A. Clark, "RTP Control Protocol Extended Reports (RTCP XR)", RFC 3611, Nov 2003

[72] L. Zhang, S. Berson, S. Herzog, S. Jamin, Resource ReSerVation Protocol (RSVP), RFC 2205, IETF, September 1997

[73] P. L'Ecuyer, "Good parameters and implementations for combined multiple recursive random number generators," Operations Research, vol. 47, pp. 59-164, 1999.

[74] K. Pawlikowski, H. D. J. Jeong, and J. S. R. Lee, "On credibility of simulation studies of telecommunication networks," in Communications Magazine. vol. 40: IEEE, 2002, pp. 132-139.

[75] Intel, "Intel® Math Kernel Library," 9 ed, 2006.

[76] P. L'Ecuyer, R. Simard, E. J. Chen, and W. D. Kelton, "An object-oriented random number package with many long streams and substreams," Operations Research, vol. 50, pp. 1073 - 1075, 2002.

[77] Kevin Fall, Kannan Varadhan, "The ns Manual (formerly ns Notes and Documentation)", June 2006, http://www.isi.edu/nsnam/ns/ns-documentation.html

-[78] Sally Floyd, Eddie Kohler, Jitendra Padhye, ICIR, UCLA, Microsoft Research, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", IETF, RFC 4342, March 2006.

[79] A-L. Burness, A. Smith, P. Eardley, "Interactive Applications using DCCP", 8 July 2005, IETF, Internet Draft, BT, Ipswich, UK

[80] ITU-T: Series G: Transmission Systems and Media, Digital Systems and Networks, Recommendation G.114, One-way Transmission Time, May 2000

[81] Joerg Widmer, Mark Handley, "Extending Equation-based Congestion Control to Multicast Applications", Proc. ACM SIGCOMM (San Diego, CA), Aug. 2001

[82] Private Communication with Marta Mrak with respect to AceMedia, 2004

[83] M. Luby, V. Goyal, "Wave and Equation Based Rate Control (WEBRC) Building Block", RFC 3738, Digital Fountain and M.I.T., April 2004

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

[85] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999

[86] Joerg Widmer, "Equation-Based Congestion Control", Diploma (Master) Thesis, Ch. 3, pp. 17-33, February 2000, www.icir.org/tfrc

[87] Eddie Kohler, Sally Floyd, " Faster Restart for TCP Friendly Rate Control (TFRC)", 14 July 2008, IETF, Internet Draft

[88] Jörg Widmer, Martin Mauve, and Jan Peter Damm, "Probabilistic Congestion Control for Non-Adaptable Flows", Technical Report 3/2001, Department of Mathematics and Computer Science, University of Mannheim.

[89] C. Boutremans, J. Y. Le Boudec, "Adaptive Delay Aware Error Control for Internet Telephony", Proceedings of 2nd IP-Telephony workshop, Columbia University, New York, April 2001, pp. 81-92.

[90] J. Mahdavi and S. Floyd, “TCP-Friendly unicast rate-based flow control,” in Draft posted on end2end mailing list, January 1997, [http://www.psc.edu/networking/papers/tcp friendly.html](http://www.psc.edu/networking/papers/tcp_friendly.html)

[91] I. Rhee, V. Ozdemir, and Y. Yi, “TEAR: TCP emulation at receivers - flow control for multimedia streaming,” Technical report, Department of Computer Science, NCSU, April 2000.

[92] John Day, "Patterns in Network Architecture", Ch. 6, pp. 185-192, ISBN-10: 0132252422, Publisher: Prentice Hall; 1 edition (27 Dec 2007)

[93] D. Tan and A. Zakhor “Real-time Internet Video Using Error Resilient Scalable Compression and TCP friendly Transport Protocol”. IEEE Transactions on Multimedia, May 1999.

[94] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", July 2003, Request for Comments: 3550, pages 67-69

[95] N. Sprljan, PhD Thesis, “A scalable video coding framework using novel rate-distortion optimisation methods”, supervisor Prof. Ebroul Izquierdo, Queen Mary University of London, 2006

[96] M. Mrak, PhD Thesis, “Motion Scalability for Video Coding with Flexible Spatio-Temporal Decompositions”, supervisor Prof. Ebroul Izquierdo, Queen Mary University of London, December 2006

[97] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction", December 1999, Request for Comments: 2733

[98] Athina P. Markopoulou, Fouad A. Tobagi, J. Karam,"Assessing the Quality of Voice Communications Over Internet Backbones" IEEE/ACM Transactions on networking, vol. 11, no. 5, October 2003

[99] Nokia N83 and N73 mobile devices, Video Recording Frame Rates, www.nokia.co.uk/phones

[100] I. Rhee, V. Ozdemir, and Y. Yi, "TEAR: TCP emulation at receivers - flow control for multimedia streaming," Technical report, Department of Computer Science, NCSU, April 2000.

[101] Doxygen, *ns*2 Documentation, www.doxygen.org , for *ns*2 version 2..27, 2004

[102] Qi Li, Di Chen, "Analysis and Improvement of TFRC Congestion Control Mechanism", School of Information Science and Engineering, Shandong University, China, pp. 1149 – 1153, IEEE Conference 2005

[103] Cisco - Voice Over IP - Per Call Bandwidth Consumption, Document ID: 7934, http://www.cisco.com/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml

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

[105] K. Kilkki, Differentiated Services for the Internet, Macmillan Technical Publishing, Indianapolis, USA, June 1999

[106] "Freephone: Why use the Plain Old Telephone when you can get so much better on the Internet?" www.sop.inria.fr/rodeo/fphone/

[107] Robust Audio Tool, April 2003, Version: 4.2.25, www-mice.cs.ucl.ac.uk/multimedia/software/rat/

[108] J. Mahdavi and S. Floyd. "TCP-friendly Unicast Rate-based Flow Control". Note sent to end2end-interest mailing list, Jan. 1997.

[109] X. Wang, J. Pitts, "Using the RED mechanism to regulate the delay and jitter performance of inelastic IP traffic with real-time QoS requirements", IET Letters, 2005

[110] S. Floyd, "RED: Discussions of Byte and Packet Modes", March 1997. With additional comments from January 1998 and October 2000. www.icir.org/floyd/REDaveraging.txt