Ethernet Services Transport Protocol with Configurable-QoS Attributes for Carrier Ethernet

Claudio Estevez[1], Georgios Ellinas[2], and Gee-Kung Chang[1]

[1] School of Electrical and Computer Engineering,
Georgia Institute of Technology, Atlanta, USA
{estevez,geekung.chang}@ece.gatech.edu
[2] Department of Electrical and Computer Engineering,
University of Cyprus, Nicosia, Cyprus
gellinas@ucy.ac.cy

Abstract. Carrier-grade Ethernet has evolved greatly since its creation. Its simplicity, scalability, and inexpensiveness have made it an attractive technology for enterprises to interconnect their metro area networks (MANs). Today, Carrier Ethernet can span over 10,000 km. Ethernet Services were created to enable companies to distribute their own services through the carrier-grade network. All the data that enters the Carrier Ethernet network (CEN) is policed by the CEN provider based on the service level agreement (SLA). Among the QoS attributes that are agreed upon is the packet loss rate (PLR) guarantee, committed information rate (CIR) and excess information rate (EIR). In this work we study the relation between PLR and throughput performance. This relationship can then be used to increase the throughput performance of the Ethernet Services transport protocol (ESTP). It achieves this by configuring an ESTP parameter that allows the CEN provider to increase the user's throughput at the cost of PLR or vice versa. This increase is done while main maintaining the throughput above the CIR and below the EIR using only transport-layer algorithms.

Keywords: Carrier Ethernet, Ethernet Services, packet loss, QoS, throughput performance.

1 Introduction

Ethernet networks started as a means to interconnect multiple devices, and it began covering only a small office area. Since then they have transitioned to much larger areas like building, campus, and metropolitan areas. This interconnection of nodes is expected to keep growing. Metropolitan Ethernet networks integrated network-to-network interfaces that connected multiple MEN domains through high capacity lines such as fiber optic links. With the increasing size of the networks the need for privacy measures increased as well. The Ethernet virtual connection (EVC) was created to resolve these issues. EVCs were designed to engage connections between end-users while isolating them from other user's connections using the same physical lines.

I. Tomkos et al. (Eds.): BROADNETS 2010, LNICST 66, pp. 69–83, 2012.

The combination of multi-domain interconnections and EVCs enabled the launch of carrier-grade Ethernet networks, and along with this Ethernet Services.

The metro Ethernet forum (MEF) is an organization that has been designing the standards for carrier Ethernet networks, releasing the first publications in 2004 and kept adding and updating them since. Among the services provided are point-to-point connections called Ethernet virtual private lines (EVPLs), which is the most prevalent service that CEN providers offer. The details of the EVPL service for specific customers is agreed upon and documented in an SLA. Some common parameters that are included in this agreement are the committed information rate, excess information rate, and the packet loss rate guarantee. These QoS parameters are controlled by a traffic policer at the network edge.

One of the main limitations of CENs is that they are high-bandwidth-delay-product networks. This creates a bottleneck at the transport layer, which is dominated by TCP and its variants. TCP relies on the acknowledgement of datagrams to successfully confirm arrival and order of the data and preserve the integrity of the original information. Because the server waits for acknowledgement (ACK) datagrams, long delays will severely degrade the overall throughput performance. For this reason a protocol called Ethernet Services transport protocol or ESTP was designed [1]. This protocol was designed to not only deal with throughput issues of high-bandwidth-delay-product networks but to also provide QoS at the transport layer, which aids the performance of CENs by relieving the transport layer from TCP's limitations and shaping the traffic such that it is compliant with the SLA requirements and therefore little information will be lost due to traffic policing.

This work is organized in the following way. The limitations of Traditional TCP and other transport layer protocols are presented in section 2. The details of ESTP are presented in section 3. In section 4, the relationship between throughput and PLR is presented followed by (section 5) the congestion avoidance algorithm derived from this relationship. The channel model described in section 6 is the basis for the simulation scenario shown in section 7. The results are found in Section 8 and the concluding remarks in section 9.

2 Limitations of Traditional TCP and Other Prominent Transport Protocols Paper Preparation

Transport Control Protocol (TCP) started as a request for comment (RFC) memorandum in 1974, with several additions in subsequent years. TCP provides congestion avoidance, delivery guarantee, and data sequence organization, making the Internet Protocol (IP) best-effort network a reliable one. The trade-off for reliability is delay. TCP is not well-suited for time-sensitive applications, such as voice over IP (VoIP). With a new generation of networks arising, the link capacity is increased and the networks extend to cover greater distances (i.e., increasing delay). These large bandwidth-delay product (BDP) networks will pose a problem to TCP. The BDP is a measure of how much data is in-flight at one point in time. The in-flight data in TCP is controlled by the congestion window (cwnd) size. The cwnd size is determined by the additive-increase multiplicative-decrease (AIMD) algorithm of TCP. This algorithm decreases the cwnd aggressively in the event of a packet loss and increases the cwnd very conservatively, making it difficult to reach the full available bandwidth. It is necessary to make modifications to the standard type to be able to fully utilize the provided bandwidth in newer versions of the transport protocol. TCP is composed of four intertwined congestion-control algorithms: slow-start, congestion avoidance, fast retransmit, and fast recovery [9]. TCP went through several versions like Tahoe, Reno, New Reno, and Sack. A simulation-based comparison of the performance of these different versions is compared in earlier work [10]. TCP Tahoe, the most native of these four versions, had the slow-start, congestion avoidance and fast retransmission algorithms. In TCP Reno, fast recovery was integrated to the TCP Tahoe congestion-control algorithms. TCP New Reno is very similar to Reno, with the exception that the retransmit wait time, after multiple consecutive packet losses, was eliminated. TCP Sack is built on top of TCP New Reno with an added feature called selective acknowledgement, hence the name Sack. In previous implementations of TCP, the receiver acknowledges the last successful segment received. However, other non-contiguous segments could have been received but not notified to the sender. If segments time out, this forces the sender to retransmit all segments after the last successful segment. With selective acknowledgement the sender acknowledges multiple non-contiguous segments such that the receiver only has to retransmit the lost segments. The analytical expression for TCP-Sack throughput is as derived in [4].

$$B = \frac{MSS\sqrt{3}}{RTT\sqrt{2bp}} \tag{1}$$

MSS corresponds to the maximum segment size, RTT is round trip time, b is the number of packets acknowledged per ACK and p is the packet loss rate. Note that MSS is usually specified in bytes and throughput in bits (per second) so a trivial unit conversion is needed.

TCP Sack became the root from which many other protocols branched out, including HighSpeed TCP [8], Scalable TCP [6], Explicit Control TCP (XCP) [7], and SLA-aware Transport Protocol [3]. The proposed protocol is built on top of the SLA-aware Transport Protocol, which is a shifted-lower-bound version of TCP Sack. Like the proposed work, the SLA-aware Transport Protocol assumes a reserved bandwidth so it shifts the lower bound of TCP Sack to match the data rate of the reserved bandwidth. HighSpeed TCP tweaks the AIMD parameters of TCP to perform better in high-bandwidth-delay networks; the parameters are obtained empirically. Scalable TCP runs a multiplicative-increase multiplicative-decrease algorithm to compute the cwnd size. The main problem with this technique is that it is very aggressive in both directions (increase and decrease), and for networks with long delays this protocol could be unstable. In XCP a congestion header is added to the packet format. Routers communicate with the transport protocol through this header field to obtain congestion feedback. The problem with this protocol is that this capability is not available in TCP so current networks are not designed to have this feature; and replacing or reprogramming all the routers in the network to support this feature might not be a feasible solution. Another problem is that routers are designed to route by reading layer 2 or 3 information, and for the router to write congestion feedback in the layer 4 packet format requires further decapsulation of the packet and adds more delay, which defeats the purpose.

3 Ethernet Service Transport Protocol

In Ethernet Services the QoS parameters specified by the SLA offer valuable information. One of the goals of ESTP is to incorporate this information into the transport protocol to have better control of the data flow over a network that employs Ethernet Services, and therefore this protocol is referred to as the Ethernet Services Transport Protocol.

ESTP can be summarized in three parts: network congestion detection mechanism, mapping function, and the congestion avoidance control algorithm. These are explained below.

3.1 Network Congestion Detection Mechanism

The principle of this method is based on the estimation of the congestion level of the network. The level of congestion is related to the amount of packets successfully delivered between two packet losses. The instantaneous amount of packets delivered between two packet losses is defined here with the parameter α. The value of α is mapped into an exponential profile to determine a less strenuous multiplicative-decrease factor defined as *map*(α). The purpose of choosing an exponential profile is to match the exponential probability distribution exhibited by the variable of the distance between two packet losses. By having the congestion window dependent on the network congestion level, the size of the congestion window can be controlled more efficiently.

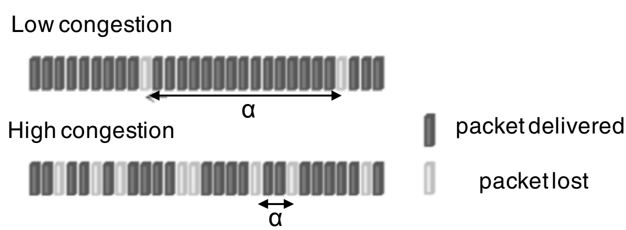


Fig. 1. Value of α is the amount of successfully transmitted packets in-between two packet losses plus one

The exact definition of α is the amount of successfully delivered packets found between two packets losses plus one. This can be easily seen by considering the metric distance to be one packet. Then, the distance between two lost packets is the amount of successfully delivered packets found in-between these two losses plus one (the lost packet). The value of α will range from 1 to ∞, which implies that the extreme cases are two consecutive losses and a lossless transmission. The value of α is mapped to the value of *map*(α), which ranges from 2 to 1. This means that the proposed protocol can decrease the congestion window size by a factor no greater than two (inclusive), and no smaller than one (exclusive). Since a factor of 1 is selected for $\alpha = \infty$ (lossless transmission), it is obvious that the protocol will never choose the exact factor of *map*(α) = 1. This mapping scheme is desirable because if α takes a small value, it is assumed to be the result of a highly congested network and therefore the congestion window size is reduced more aggressively. Inversely, if the value of α is large, it is assumed that the network is not experiencing high levels of congestion (Fig. 1) and the congestion window is only slightly reduced in size. To achieve this mapping and at the same time match the traffic loss probability distribution, an exponential function is chosen. The details of how to obtain the mapping function are found in the next section.

3.2 Mapping Function

To obtain the mapping function of ESTP it is necessary to become familiar with the probability density functions (pdf) of the period of time between two packet losses. The probability of a packet loss has a Bernoulli distribution. If we assume that a packet loss is a successful event, then the number of successful events in a fixed period of time has a Poisson distribution, i.e., the packet loss rate can be modeled as a Poisson distribution. The time between two Poisson distributed events can then be modeled as an exponential distributed process.

The pdf of an exponential distribution is:

$$f\left(\alpha;\tau\right)=\begin{cases}\frac{1}{\tau}e^{-\alpha/\tau} & \alpha\geq 0\\ 0 & \alpha<0\end{cases} \tag{2}$$

were τ is the $E[\alpha]$.

Since the main objective is to have effective congestion control, while increasing performance, the mapping function is chosen to match the pdf of the metric distance between two packets. The multiplicative decrease factor of Traditional TCP is 1/2. The metric distance unit is defined as a single packet independent of its length in bytes as this work focuses on the transport layer. Under this definition the minimum distance between two packets is one, since the measurement is taken from the start of one packet to the start of the subsequent packet. In this work the mapping function is placed in the denominator of the multiplicative decrease factor, so for the proposed protocol to be lower-bounded by the performance of Traditional TCP one of the boundary conditions has to map one metric distance to a value of 2, rather than 1/2. This will cause two consecutive packet losses to reduce the *cwnd* by a factor of 1/2, which is the behavior of Traditional TCP. If we never had a loss, such that the distance is infinite, the desired mapping factor is then 1. In summary, our boundary conditions are: *map*(1) = 2 and *map*(∞) = 1.

As discussed previously, we want to match the mapping function to the pdf of the distance between two events. Thus, since exp(∞) = 0 then one has to be added to the exponential term. This yields *map*(α) = exp($-\alpha/\tau$) + 1. To satisfy *map*(1) = 2 we need to shift the expression by 1 (see Fig. 2), which yields:

$$map(\alpha) = e^{-\frac{\alpha-1}{\tau_\alpha}} + 1 \tag{3}$$

The *cwnd* expression is then:

$$cwnd_{n+1} = cwnd_n \left(e^{-\frac{\alpha-1}{\tau_\alpha}} + 1 \right)^{-1} \tag{4}$$

This excludes the effects of $cwnd_{MIN}$, which is explained in Section 3.3.

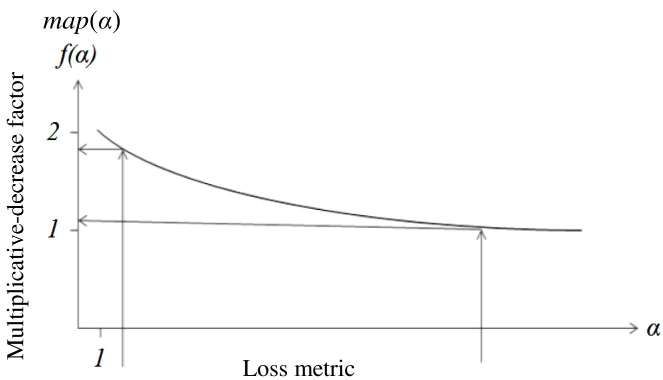


Fig. 2. Proposed protocol mapping of the distance between two packet losses to the multiplicative decrease factor mapping

Notice that for the extreme case of two consecutive packet losses ($\alpha = 1$), the proposed protocol behaves like the SLA-aware protocol [3], which means that the proposed protocol is lower-bounded by the SLA-aware throughput performance. If the SLA information is not taken into account, such that $cwnd_{MIN} = 0$, then for $\alpha = 1$ the proposed protocol behaves like Traditional TCP.

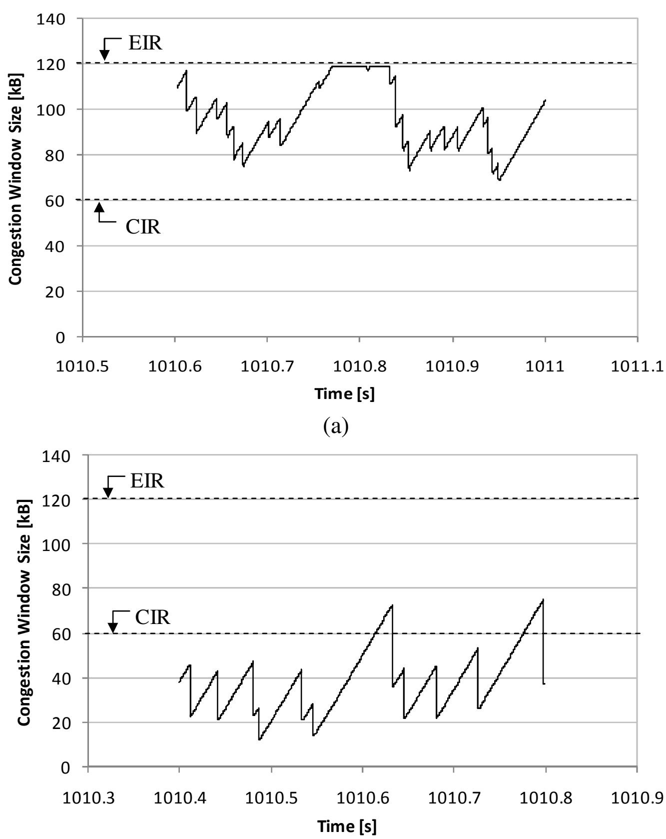
3.3 Congestion Avoidance Control

By incorporating Ethernet Services information, in addition to the dynamic multiplicative-decrease factor mapping, ESTP can further show improvement in terms of throughput performance, since these techniques are independent of each other. Combining the congestion control mechanisms discussed in the previous subsection with the SLA-aware mechanisms, the following multiplicative decrease expression can be derived for congestion control:

$$cwnd_{n+1} = \frac{cwnd_n - cwnd_{MIN}}{map(\alpha)} + cwnd_{MIN} \tag{5}$$

$$cwnd_{n+1} = \left(cwnd_n - cwnd_{MIN}\right)\left(e^{-\frac{\alpha-1}{\tau_\alpha}} + 1\right)^{-1} + cwnd_{MIN} \tag{6}$$

This technique takes full advantage of the bandwidth provided to the subscriber by maintaining the throughput above the CIR (see Fig. 3a). The $cwnd_{MIN}$ is computed in the following way: $cwnd_{MIN} = RTT{\cdot}CIR$. The throughput can be further improved by utilizing the EIR information. $cwnd_{MAX}$ can be obtained similarly to $cwnd_{MIN}$ but using EIR rather than CIR: $cwnd_{MAX} = RTT{\cdot}EIR$. With this value, the upper bound of the congestion window can be controlled. An upper bound is set because once the *cwnd* exceeds $cwnd_{MAX}$, the throughput will exceed EIR and the traffic policing enforced at the lower layers will discard packets exceeding this rate. Once that happens, the protocol at layer 4 will initiate its congestion control mechanism and decrease the congestion window. This is unnecessary since there is no congestion in the network. By maintaining the *cwnd* at its maximum value, the throughput will not be reduced by the protocol's congestion control and the subscriber will get maximum throughput until a random loss (e.g. link loss) or congestion loss occurs.



(b)

Fig. 3. Congestion window behavior of *(a)* the proposed protocol and *(b)* Traditional TCP

The additive increase properties of AIMD of the proposed protocol will then be expressed as:

$$cwnd_{n+1} = \min\left(cwnd_{MAX}, cwnd_n + 1 / cwnd_n\right) \tag{7}$$

Traditional TCP is then a special case when $cwnd_{MAX} = \infty$. The traffic-loss profile matching and the *cwnd* boundary mechanism implemented by the additive-increase multiplicative-decrease techniques distinguishes ESTP from other transport protocols and makes its implementation desirable for Ethernet Services. In Fig. 3a it can be seen how ESTP successfully maintains the congestion window within the designed limits, and the contrasting behavior of Traditional TCP can be observed in Fig. 3b.

4 Throughput and Packet Loss Rate Relationship

For connection-oriented transport protocols, such as TCP, the throughput is greatly affected by the packet loss rate. As seen in (1), the throughput of Traditional TCP decreases as the packet loss rate increases by a factor of its square root. One method to improve the throughput is to study the loss pattern of data traffic and use this information to design a more efficient congestion avoidance algorithm.

In section 3.1, a parameter α is defined as the distance between two packet losses. This parameter is the key to determine the congestion level of the network, but it is also the link between the PLR and the throughput. It is clear that the α term is related to the packet loss rate, since both can be computed from packet losses. What is less intuitive is that the average α can be used to estimate the throughput of ESTP, since ESTP uses this parameter to adjust the congestion window size, and hence the throughput. By linking the throughput to the PLR, we can adjust the congestion window size such that we provide higher throughputs without increasing significantly the PLR. The details of this principle follow below.

Packet loss can be categorized into two types: intrinsic and extrinsic. Intrinsic losses are the losses that cannot be avoided (by the transport layer protocol) because they are embedded in the system. An example of intrinsic loss is those caused by

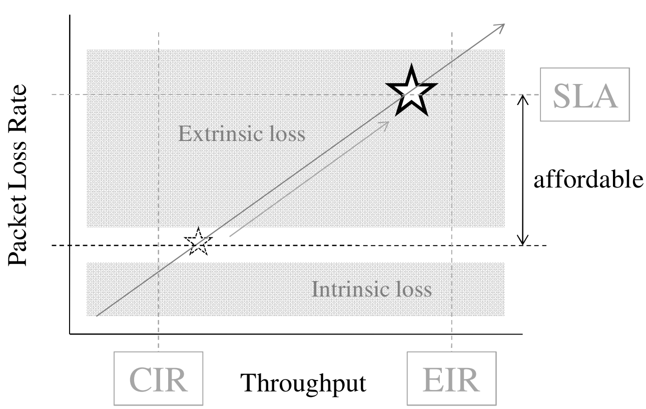


Fig. 4. Illustration of the throughput limited by PLR guarantee principle

faulty transmission lines. In contrast, extrinsic losses are the losses caused by external reasons such as congestion discarding or traffic policing, which can be avoided with appropriate control mechanisms. If the throughput were to increase the intrinsic packet loss rate will not change, because it is a fixed loss rate embedded in the system, but the extrinsic packet loss rate would increase because the risk of congestion and losses due to traffic policing will increase as the data rate increases. In CEN, the customer is allowed to customize the QoS parameters to suit the needs of their networks. As discussed in the introductory section, one the QoS services users can purchase is a PLR guarantee. This loss rate is chosen according the amount of loss that the applications running on the network can tolerate. The main idea (depicted on Fig. 4) is to increase the throughput to the highest possible value in which the PLR specification is not exceeded, and doing this in an innate manner (naturally embedded in the algorithm) so that no additional policing is required at the transport layer.

5 Congestion Avoidance Control with Configurable-QoS

The congestion avoidance control with configurable-QoS is a closed-loop feedback control system, were the input parameter is α and the output parameter is the congestion window size. To provide a PLR boundary, we compute what the average α (i.e. $E[\alpha]$) needs to be for this specific PLR and make sure the average does not exceed a predefined threshold. To do this we use the assumptions used to find the mapping function described in section 3.2. These assumptions are: the packet loss event has Bernoulli distribution and the distance between two Bernoulli distributed events has exponential distribution with average $E[\alpha] = \tau_\alpha$. The PLR is translated to the $E[\alpha]$ using the following equation (see Fig. 5):

$$\tau_\alpha = \frac{average\ packet\ transmission\ period}{PLR} \tag{8}$$

and the resulting value of τ_α is used in (6).

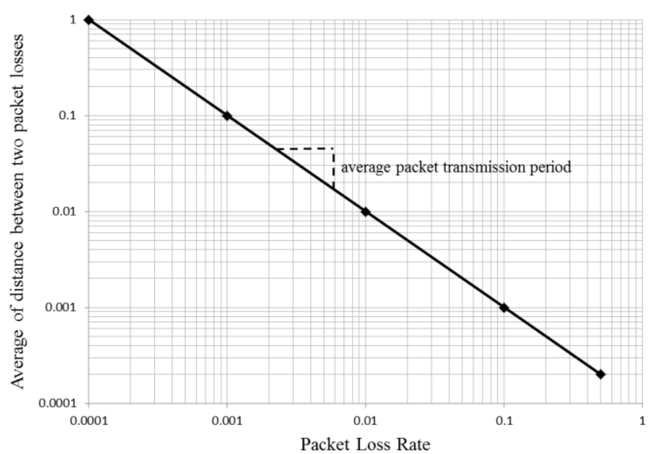


Fig. 5. Relationship between $E[\alpha]$ and *PLR*

It should be emphasized that this is only true under the assumption that the PLR increases if the throughput is increased. This is a fair assumption and is true for most, if not all, networks. To visualize the mechanics of this algorithm assume the system has encountered two losses, and the transport protocol has computed its respective α and this value is much smaller than $E[\alpha]$. The protocol assumes the network is congested and will reduce the congestion window size aggressively (see section 3 for details). This will cause the throughput to reduce drastically, hence reducing the congestion in the network and consequently next value of α will probably be higher than $E[\alpha]$. This iterative process will generate values of α that surround $E[\alpha]$ and therefore maintaining a relatively constant PLR. So the guarantee is an indirect effect of the congestion avoidance mechanism. Since the average packet transmission period is inversely proportional to throughput from (8) is it clear that increasing τ_α will lower the throughput but advantageously will also decrease the PLR; inversely, increasing the τ_α value will increase the throughput but at the cost of increasing the PLR. So τ_α is a design parameter of ESTP and the main focus of this work.

6 Channel Model

According to information theory, the capacity of a channel, which is defined as the amount of information transmitted between sender and receiver, is the upper bound of the data rate that can be delivered through the channel without error. Since Internet is known to be composed mainly by TCP transmissions and In order to evaluate the channel capacity of a CEN line, an erasure channel model is proposed. As shown in Fig. 6, the traffic from the sender is policed according to the SLA at the egress point. The packet that falls into the CIR profile will be marked with high priority and delivered with guaranteed QoS. A packet that falls out of the CIR profile, but still in the EIR profile, will be marked with low priority and delivered in a best effort manner. A packet that falls out of the EIR profile will be discarded immediately. Within the transport network, a high priority packet suffers a loss probability of p_1, which is defined in the SLA. A best-effort packet suffers a loss probability of p_2, which depends on background traffic, and can be any value between 0 and 1. For a prioritized channel as shown in Fig. 6, the capacity can be expressed as:

$$C = CIR \cdot (1 - p_1) + EIR \cdot (1 - p_2) \tag{9}$$

Given that $0 \le p_2 \le 1$, the lower bound of the channel capacity can be derived as

$$C = CIR \cdot (1 - p_1) \tag{10}$$

which is the minimum capacity of a CEN channel.

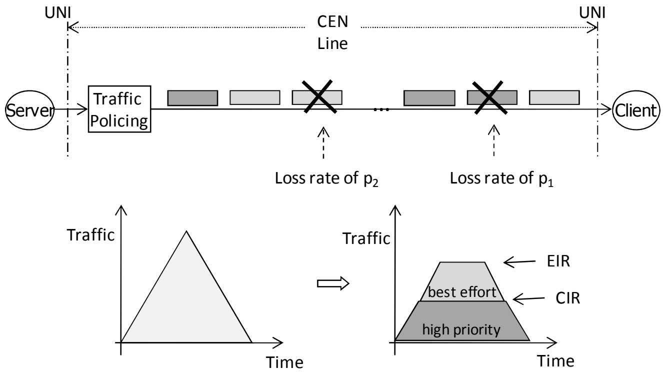


Fig. 6. Channel model for a Carrier Ethernet Network Line

7 Simulation Scenario

The simulations were realized in OPNET Modeler 16.0 based on the channel model described in the previous section. A server transmits a 2 GB file to the client using the QoS and ESTP parameters specified in Table 1. The channel has an extrinsic loss rate that is dependent on the throughput, and modeled by the empirical formula described by (11).

$$p_{chLoss} = p_1 \left(\frac{p_2}{p_1} \right)^{\left(\frac{Throughput - CIR}{EIR - CIR} \right)} \tag{11}$$

This only applies to the best-effort traffic; the CIR traffic is only subjected to the intrinsic loss rate. The first simulation is to prove the concept of throughput increase by sacrificing PLR. To prove concept it is necessary to accurately know the packet loss model of the CEN line, which is the reason for inducing a controlled loss. The realness or practicality of the CEN line loss model is a secondary issue, but as long as the loss is monotonically increasing with increasing throughput the proof of concept can be realized with any CEN line model. The simulation parameters used for Simulation I are found below (Table 1).

Table 1. Simulation I Parameters

Parameter description	Value
CIR	0 Mbps (no CIR)
EIR	500 Mbps
p_1	0.0000001
p_2	0.01
τ_α	[0.01, 0.0001]
Round-trip propagation delay	1 ms
Receiver buffer size	1 MB
Maximum segment size	1460 bytes
Max ACKed packets (b value)	2

The value of τ_α is varied to see the throughput improvement, while keeping a log of successfully delivered packets and packet losses. The throughput versus the PLR is found and analyzed in the following section.

The second simulation consists on showing the CIR and EIR functionalities of the protocol. In this case the RTT is varied from 1 to 10ms and the throughput is recorded for each event. The τ_α is left constant. The throughput versus RTT is analyzed. The simulation parameters for this case are found in Table 2.

Table 2. Simulation II Parameters

Parameter description	Value
CIR	200 Mbps
EIR	400 Mbps
p_1	0.0000001
p_2	0.01
τ_α	0.01
Round-trip propagation delay	[1, 10] ms
Receiver buffer size	1 MB
Maximum segment size	1460 bytes
Max ACKed packets (b value)	2

Also the pdf of the distribution of α in the simulation is provided to show the similitude to the theoretical case, the plot is provided in the results section.

8 Results

The PLR versus throughput is plotted in Fig. 7. It can be seen that for the same RTT the throughput can be further increased by varying the value of τ_α. The throughput is

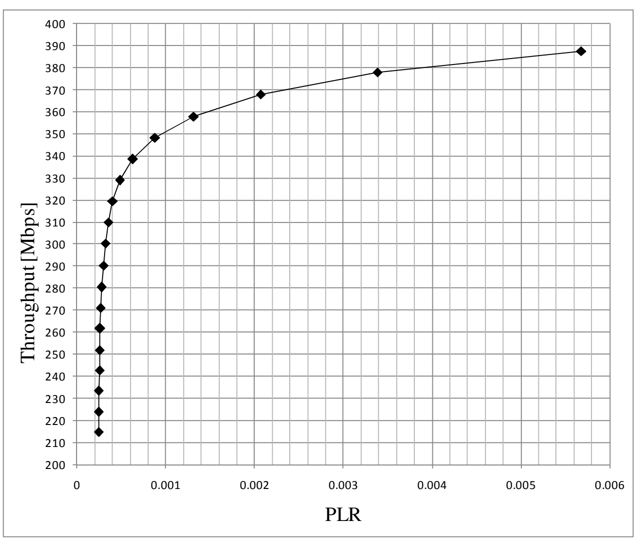


Fig. 7. Throughput versus *PLR*

more sensitive to this change at the lower PLR, as the PLR increases the throughput becomes less sensitive to this change.

This plot shows the ability of the protocol to reconfigure its parameters to satisfy the changing needs of the customer. If the applications that are been ran in the network need to prioritize throughput or PLR this can easy be achieved by changing the ESTP's parameters accordingly.

For the second simulation case the main interest is to show the CIR and EIR boundaries. Fig. 8 shows the results of the throughput versus RTT.

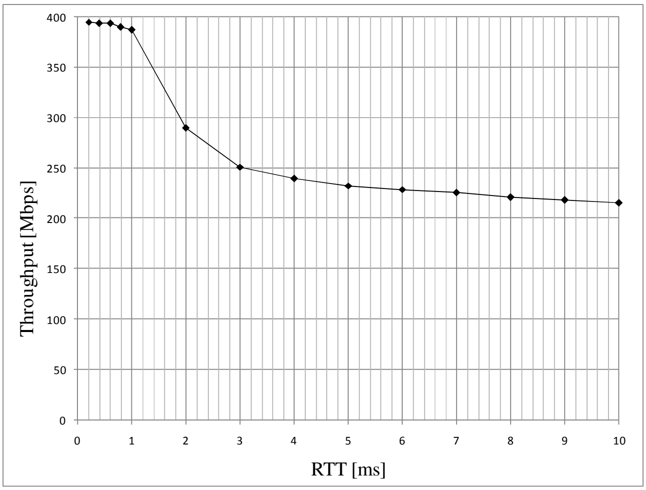


Fig. 8. Throughput versus *RTT*. Shows CIR advantage and EIR enforcement

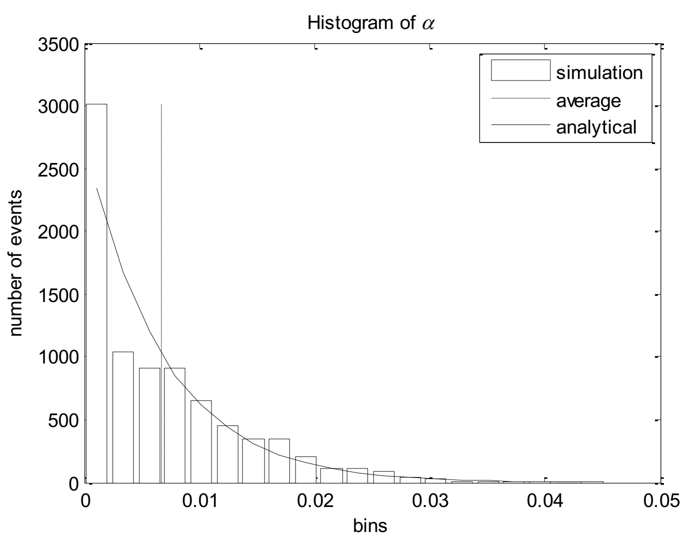


Fig. 9. Simulation histogram of the probability distribution of α

It can be seen that all the best-effort traffic is found between these boundaries. Because the CIR is bandwidth that is guaranteed to the customer ESTP can lower bound the congestion window size, which will maintain the throughput above the CIR. If the customer required more or less bandwidth the provider can reserve the adjusted bandwidth for the customer and change the ESTP's parameters accordingly so that the transport layer performs more efficiently based on the available resources and customer's needs.

The pdf of α is provided for work completeness. It is interesting to see the behavior of the loss pattern. Our initial assumption was that the behavior is exponential and Fig. 9 shows that the simulation behavior is in agreement with the hypothesis.

9 Summary

A summary of all the QoS-configurable parameters in ESTP follows (Table 3). These parameters can be computed from the specifications established in the SLA and legacy parameters of Traditional TCP, available at the transport layer.

Table 3. Configurable QoS Parameters of ESTP

Parameter	Parameter description
$cwnd_{MAX}$	Upper bounds the congestion window. It prevents the throughput from exceeding the EIR and therefore will prevent packet loss due to policing, and therefore congestion window will not be penalized and the throughput will remain closer to the EIR. $$cwnd_{MAX} = RTT \cdot EIR$$
$cwnd_{MIN}$	Lower bounds the congestion window. It allows the throughput to remain at or higher than the CIR. This will allow the protocol to fully utilize the available bandwidth even in the event of a packet loss. $$cwnd_{MIN} = RTT \cdot CIR$$
τ_{α}	Establishes a packet loss rate guarantee. A high value will lower the PLR, but at the cost of lowering the throughput. It should be calculated based on the packet loss rate that the applications of the network can tolerate.

10 Conclusions

A broadband data transport protocol was designed for Carrier Ethernet Networks. The protocol has a unique congestion avoidance algorithm and has many configurable parameters to suit the reconfiguration needs of CEN providers and can fulfill the throughput needs of the CEN customers. ESTP can fully utilize the reserved bandwidth provided to the customer by lower bounding the congestion window size, the end result of this bounding technique is that the throughput will be lower bounded to the CIR value, as agreed on the SLA. The ability to reconfigure is attractive to CEN providers and this protocol can support these QoS attributes using only resources available at the transport layer. Its compatibility with CEN traffic policing makes it highly suitable for carrier-grade applications running on Carrier Ethernet networks.

References

1. Estevez, C., Ellinas, G., Chang, G.-K.: Broadband Data Transport Protocol Designed for Ethernet Services in Metro Ethernet Networks. In: IEEE GLOBECOM 2008, New Orleans, LA (November 2008)
2. Xiao, C., Estevez, C., Ellinas, G., Chang, G.-K.: A Resilient Transport Control Scheme for Metro Ethernet Services Based on Hypothesis Test. In: Proc. IEEE GLOBECOM 2007, Washington, DC (November 2007)
3. Xiao, C., Chang, G.-K., Bing, B.: An SLA-aware Transport Protocol for High Throughput Wide Area Ethernet Services. In: Proc. IEEE GLOBECOM 2006, San Francisco, CA (November 2006)
4. Padhye, J., Firoiu, V., Towsley, D., Kurose., J.: Modeling TCP Throughput: A Simple Model and its Empirical Validation. In: Proceedings of ACM SIGCOMM 1998, Vancouver, B.C. (September 1998)
5. Yang, Y.R., Lam, S.S.: General AIMD Congestion Control. Technical Report TR-2000-09, The University of Texas at Austin (May 2000)
6. Kelly, T.: Scalable TCP: Improving Performance in Highspeed Wide Area Networks. Computer Communication Review 33(2), 83–91 (2003)
7. Zhang, Y., Henderson, T.R.: An Implementation and Experimental Study of the eXplicit Control Protocol (XCP). In: Proc. IEEE Infocom 2005, Miami, FL (March 2005)
8. Floyd, S.: HighSpeed TCP for Large Congestion Windows. IETF RFC 3649 (December 2003)
9. Allman, M., Paxson, V., Stevens, W.: TCP congestion control. IEFT RFC 2581 (April 1999)
10. Fall, K., Floyd, S.: Simulation-based comparisons of Tahoe, Reno and Sack TCP. Computer Communication Review (July 1996)
11. Vertical Systems Group: Business Ethernet Expands 43% in 2008. Vertical Systems News (February 2009)
12. Vertical Systems Group: Worldwide Business Ethernet Services Market Rises to $38.9 Billion by 2013. Vertical Systems News (March 2009), http://www.verticalsystems.com/news.html
13. Rejaie, R., Handley, M., Estrin, D.: RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet. In: Proceedings of IEEE INFOCOM 1999, vol. 3 (March 1999)
14. Sisalem, D., Schulzrinne, H.: The Loss-delay based Adjustment Algorithm: A TCP-friendly Adaptation Scheme. In: Proceedings of NOSSDAV 1998, Cambridge, UK (July 1998)
15. Kurose, J., Ross, K.: Computer Networking: A Top-Down Approach Featuring the Internet, 3rd edn. Pearson Addison-Wesley (2005)