Decoupling Loss Differentiation and Loss Recovery to Ensure Security and Performance

Venkatesh Obanaik, Zhao Hang, A L Ananda
Communications and Internet Research Lab, School of Computing
National University of Singapore, Singapore - 117543
Email: {venka,zhaohang,ananda}@comp.nus.edu.sg

***Abstract*— Protocols such as TCP and congestion control mechanisms like TFRC suffer in a hybrid wired-cum-wireless scenario where losses can occur for reasons other than congestion, viz. due to biterrors in the wireless link. The existing solutions to address the problem either misclassify the losses, violate end-to-end semantics, do not co-exist with IPSec or are TCP-specific. We have previously proposed an innovative mechanism, Secure Performance Enhancing Proxy (SPEP) which preserves end-to-end semantics, ensures end-to-end security and enhances performance. In this paper we show that the decoupling of loss differentiation from loss recovery mechanism enables SPEP to serve as a generic loss differentiation mechanism at the network layer. We show by means of test bed experiments that SPEP work in conjunction with any transport layer and protocols such as UDP with TFRC.**

***Keywords*: Performance Enhancing Proxy, Heterogeneous Network, End-to-End Security, TCP, TFRC**

I. Introduction

The inability of protocols such as TCP and schemes like TFRC to differentiate between congestion and corruption losses results in poor performance in a hybrid wired-cum-wireless scenario. RFC 3135 [3] identifies some of existing solutions to address this problem and implications of using such solutions. SPEP [1] was designed to address all implications identified in [3]. In this paper, we show how these features of SPEP can be made available to other protocols by decoupling loss differentiation from loss recovery and by having a generic loss differentiation mechanism at the network layer. The decoupling not only ensures security at the network layer but also enhances performance at the transport layer.

II. Related Work

Considerable amount of research work has been done to improve the performance of TCP over the wireless networks. However, we see that the existing solutions for TCP such as Snoop, W-TCP, I-TCP are either too TCP-specific are designed oblivious of the security considerations or do not suit all kinds of applications, for instance link layer solutions like Forward Error Correction (FEC) result in varying Round Trip Times (RTT) which is not suitable for delay sensitive applications.

III. Decoupling Loss Differentiation and Loss Recovery

The SPEP loss differentiation mechanism described in Section IV is implemented at the Internet Protocol layer. In FreeBSD, the Internet Protocol layer maintains a data structure called Internet Protocol Control Block (INPCB) for every connection. The INPCB contains among other things, the information about the connection end point identifiers and a pointer to the TCP Control Block (TCPCB) which contains TCP state information. All the information necessary for UDP is available in the INPCB. Whenever a socket is created, the corresponding underlying transport layer protocol creates INPCB and attaches it to the socket structure. Our motivation is to design a generic loss differentiation mechanism accessible to all transport and application layer protocols. Therefore, we have implemented the SPEP loss differentiation mechanism at the network layer. The SPEP mobile node component detects the nature of packet loss and stores the information for each lost packet in the INPCB structure, providing a generic interface to all higher layer protocols.

The loss recovery is done by the corresponding transport layer or application layer protocol by using the information about the nature of loss obtained from loss differentiation mechanism at the network layer. In the case of TCP as explained in [1], TCP layer accesses INPCB to retrieve the information about the nature of loss and performs loss recovery using a mechanism like ELN. However, TFRC is an equation based congestion control mechanism [2], where the sender estimates the sending rate based on the information about the network conditions conveyed by the receiver. The TFRC receiver generates receiver reports which contain crucial loss rate information and the necessary information to calculate Round Trip Time (RTT). The packet losses that occur due to biterrors have to be discounted in the loss event calculation. SPEP loss differentiation mechanism provides TFRC the capability to identify such losses so that it can react accordingly.

TCP and TFRC differ in congestion control mechanisms, react differently to packet losses and have protocol specific loss recovery mechanisms. However, all such protocols require a loss differentiation mechanism when operating in a heterogeneous wired-wireless network. We see that a generic implementation of loss differentiation mechanism at the network layer as shown in Fig. 1 is beneficial to all such protocols rather than a mechanism that is tightly coupled to a particular protocol. Moreover, using the existing security architecture to implement the loss differentiation mechanism ensures security at the network layer.

Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)
0742-1303/04 $ 20.00 IEEE

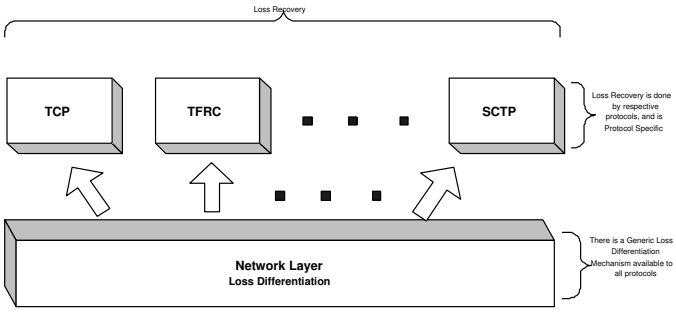


Fig. 1. Decoupling Loss Differentiation from Loss Recovery

IV. Performance Evaluation

A controlled network environment as shown in Fig. 2 was set up to carry out the tests. We have conducted experiments to show the behavior of TCP with SPEP scheme and reported the results in [1]. In this paper, we have carried out tests to study the achieved performance improvement in presence of congestion as well as corruption. All the test runs were conducted by making a bulk data transfer from sender to receiver for 100s.

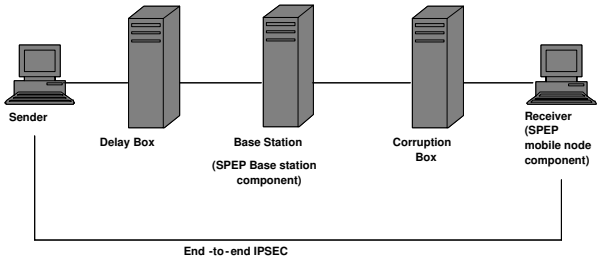
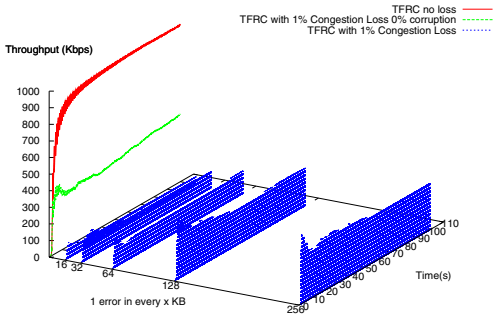
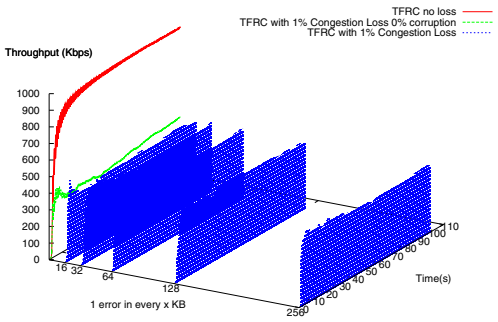


Fig. 2. SPEP Test Configuration

The Fig. 3 shows the 3-dimension plots that represent average throughput of a TFRC connection over time for different levels of corruption. The points on the X-axis represents 1 error in every 'x' KB of data. The Y-axis represents the running time of experiments and the Z-axis represents the average throughput of the connection at any instant of time. Figs. 3(a) and 3(b) depict the behavior of TFRC when there is both congestion and corruption. Fig. 3(a) depicts the throughput of the standard TFRC connection in the presence of 1% congestion loss and for various levels of corruption. There are two throughput graphs at point '0' on the X-axis. The throughput graph that obtains a throughput of close to 950Kbps, indicates the throughput of TFRC when there is no packet loss (neither congestion loss nor corruption loss). The other throughput graph at point '0' on the X-axis indicates the throughput of TFRC when there is 1% congestion loss and no corruption loss and the throughput is around 410Kbps. The throughput graphs at various levels of corruption on the X-axis show that standard TFRC reacts to both congestion and corruption losses and performs poorly. The throughput degrades close to 40Kbps in presence of 1% congestion loss



(a) TFRC without SPEP



(b) TFRC without SPEP

Fig. 3. Performance of TFRC for various levels of corruption loss and 1% congestion loss

and a corruption of 1 error in every 16KB. When TFRC is used in conjunction with SPEP we would expect it to distinguish between congestion and corruption losses and react only to congestion losses. This behavior can be seen in Fig. 3(b) which depicts the throughput of the TFRC used in conjunction with SPEP in the presence of 1% congestion loss and for various levels of corruption. We see that irrespective of levels of corruption, TFRC used with SPEP obtains the throughput close to that of standard TFRC with 1% congestion loss and no corruption loss.

V. Conclusion

The SPEP approach described in this paper, offers a unique solution at the network layer which can be readily used by any transport or application layer protocol. We have shown that, by decoupling loss differentiation from loss recovery, we can achieve both performance improvement and ensure security.

References

[1] V. Obanaik, L. Jacob and A. L. Ananda *SPEP: A Secure and Efficient Scheme for Bulk Data Transfer over Wireless Networks*, in Proceedings of IEEE Wireless Communications and Networking Conference, March 2004.

[2] S. Floyd, M. Handley, J. Padhye, J. Widmer *Equation-Based Congestion Control for Unicast Applications*, in Proceedings of SIGCOMM 2000, August 2000.

[3] J. Border, M. Kojo, J. Griner, Z. Shelby *Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*, RFC 3135, June 2001.