PACKET PAIRING FOR TCP IN HETEROGENEOUS NETWORKS

*Biljana STOJCEVSKA, **Oliver B. POPOV
*Institute of Informatics, Faculty of Natural Sciences and Mathematics
Saints Cyril and Methodius University, PO Box 162, MK-1000 Skopje
Republic of Macedonia, E-mail: biljanas@ii.edu.mk
**Department of Information Technology and Media, Mid Sweden University
SE - 851 70, Sundsvall, Sweden, E-mail: oliver.popov@miun.se

SUMMARY

Since the early days of the Internet, TCP has been the premier transport protocol for most if not all applications. TCP predates IP and was designed to run over any packet-switching wireline network environment where errors due to the underlining medium are an exception rather then a rule. This complex protocol, in addition to many services such as flow control, congestion management and reliability, attempts to provide both an efficient utilization and a fair share of network resources. Wireless and mobile networks generate an environment in which TCP may suffer severe performance degradation due to possible misinterpretation of random losses that come from high BER, low and variable bandwidth, and frequent disconnections. The reason is the triggering of the congestion management mechanisms in response to the losses since TCP assumes by default that a lost packet is an indication for network congestion. The results are a decrease in throughput, poor resource utilization and eventually an unacceptable performance of the applications. There is no alternative in making TCP understand heterogeneity through modification and enhancement of the algorithms that the protocol consists of. Naturally, there are options that are enumerated in the article, but in the model we propose the preservation of the E2E semantics is an imperative. In addition to an overview of the "mainstream" research area, this article present the deployment of the loss pair method as a possible solution to some congestion management issues and thus making heterogeneity transparent to TCP.

Keywords: *TCP, heterogeneous networks, congestion control, loss pairs*

1. INTRODUCTION

TCP has been the transport protocol for most if not all applications on the Internet. It was designed to run over any packet-switching wireline network where errors due to the transmission medium have become an exception rather then a rule. In addition to many services such as flow control, congestion management and reliability, TCP attempts to provide both an efficient utilization and a fair share of network resources. However, TCP in wireless and mobile networks may suffer a significant performance degradation due to the misinterpretation of random losses that come from high BER, low and variable bandwidth, and frequent disconnections.

The reason is not the retransmission of the lost packets, but the congestion control (CC) algorithms that are triggered in a response to the losses. Namely, TCP always assumes that a lost packet signals the appearance of network congestion. If the loss is indicated by a timeout, TCP enters a slow-stat phase. When the loss is detected by a receipt of three duplicate acknowledgements, then the answer is fast recovery. The net result in both cases is the reduction of the congestion window size. On the other hand, whenever a packet loss is due to a transmission error, any contraction of the window is inadequate. The final outcome of this behavior is decreased data throughput, poor utilization of the network resources and marginal performance of the applications.

It is obvious that some new modifications of TCP to accommodate the conditions imposed by wireless and mobile networks are needed. So far, there is no a single solution for making TCP operate seamlessly in heterogeneous networks (and hence on the Internet). The overview of the "mainstream" research area in the article is followed by the analysis and the evaluation of the proposed solutions, including the results of our study that are based on the loss pair approach.

2. TCP AND HETEROGENEITY

The properties of wireless channels are characterized with frequent random losses. Transmission errors can be caused by physical obstacles, interference, shadowing, signal fading etc. The bit-error rate (BER) for wireless medium is in order of 10^{-3}, which translates as packet loss rate of 12 percent [11], for packets with size of 1500 bytes. In comparison, the BER for wired medium is in the order of 10^{-6} to 10^{-8}, which equals to a packet loss rate from 1.2 to 0.012 percent. Mobility itself is also a generator for packet loss and increased packet delay. When a user migrates between cells, the connection handoff may induce packet losses.

While TCP may face a number of problems in wireless networks, random losses are the most relevant to the overall performance. The reduction of the transmission rate, which leads to throughput degradation, is due to the misinterpretation of the symptoms and consequently the application of wrong remedies, such as contraction of the congestion window. For instance, a packet loss of 2% in a TCP Reno flow leads to 45% decrease of the throughput. The negative effect of this behavior is even more exaggerated in paths with high bandwidth-delay products (BDP), such as satellite links.

The ideal TCP behavior in case of a random loss would be retransmission of the lost segment without activating congestion control. On the other hand, the ideal network corrective action would be taking care of the loss transparently to the TCP. The usual approach for solving this problem has been through a dichotomy of loyalties either to TCP or to the network.

In the last ten years, various techniques for improving TCP performance in heterogeneous networks have been proposed. The strategies can be classified in three categories: link-layer solutions, split connection and TCP variations [13], [14], [15].

a. Link-layer schemes

The link-layer schemes [2], [12], [10], [18] attempt to localize the solving of the problem by hiding the deficiencies of the wireless medium from the upper layers. The data link layer offers service with quality close to the one of a wired link and there is no need for modification of TCP.

There are mainly two mechanisms that are used in the data link protocols: local retransmissions and Forward Error Correction (FEC). FEC can accomplish error correction of small number of bits, but it adds overhead even when there are no errors present. It increases the computation complexity and the energy consumption. Local retransmissions at the data link layer are used to recover the TCP connection from loss errors.

b. Split connection

The split connection schemas [1], [6] divide the TCP connections in two parts: one going over the wired segment of the network and one going over the wireless channel. The two parts meet at the base station that connects the wired and the wireless part of the network. A protocol tailored to wireless transmission is used for the wireless part.

The base station acknowledges every received segment, without waiting for its arrival at the mobile host. The biggest deficiency of this behavior is the possibility that an acknowledgement for a segment may arrive at the sender before the segment actually arrives at the mobile receiver. This is a violation of the end-to-end semantics of TCP.

c. TCP variations

TCP variations introduce new functionalities to the basic protocol [3], [5], [8], [9], [19], [17]. They preserve TCP end-to-end (E2E) semantics and thus conform to the IP family inclusion rule - any new or modified protocol should be scalable.

The TCP problem in heterogeneous networks is interpreting random losses as indication for congestion. For all TCP variations, the key is to determine the nature of the errors that caused packet losses. If indeed there is congestion along the path, the protocol should apply standard congestion control mechanisms. Conversely, if the reason behind the loss is a transmission error, then the sender should keep the current sending rate.

The research adheres to the E2E paradigm, and hence embraces one of the three universal principles of the Internet (IP over everything and “best-effort” service model being the other two).

3. THE LOSS PAIR METHOD

A good estimation of network characteristics based on E2E measurements is important for the applications that should dynamically adapt to the network state. A common method used for this type of measurements is to send a train of successive packets and identify their round trip times (RTTs). It appears that these methods may be used in TCP congestion control also.

The round trip times of a TCP connection contain information about the underlying network conditions. The idea is to use this information for finding out why a segment is lost in the TCP connection. In a way, it is a problem of an Internet forensics, just in this case applied to network traffic models rather then security problems.

The loss pair method has been developed as a tool for determining the characteristics of a network by using E2E measurements [7]. It can find out the throughput capacity of a bottleneck link along a path and the reason for packet dropping.

A loss pair is a pair of packets that travel close to each other through the same path in the network. When one of them is dropped, the round trip time of its pair supplies information about the state of the network at the time of the drop. Formally, a loss pair is defined as pair of packets p_1 and p_2 and:

1. p_2 initially follows p_1 with time Δ;
2. exactly one of them is dropped;
3. both packets travel on the same path before one of them is dropped.

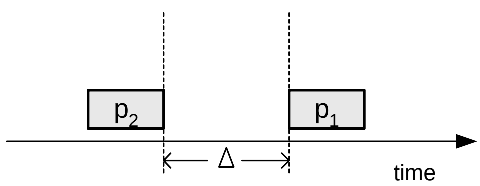

Fig. 1 “A loss pair”

A loss pair is formed whenever one of the packets gets dropped. It is irrelevant weather p_1 or p_2 is dropped – both cases give the same results [7]. The assumptions are that (1) the path in the network and the location of the bottleneck link do not change during the measurement, and (2) most of the losses take place at the bottleneck. These assumptions guarantee that the bottleneck state was similar when both of the packets passed the bottleneck node (an observational invariance). The observation of the of the packet p_2 round trip time provides the reason for dropping p_1. Namely, the round trip times of the loss pairs are tightly grouped around the time necessary to drain the bottleneck queue plus the time of the links traversal.

4. THE LOSS PAIR METHOD AND TCP CONGESTION CONTROL

The network path of a TCP connection (Figure 2) can be modeled as a graph with $n+1$ nodes and n links. The nodes $1, 2, \ldots, n$- are routers and the nodes 0 and n are the hosts where the TCP source and destination entities are located.

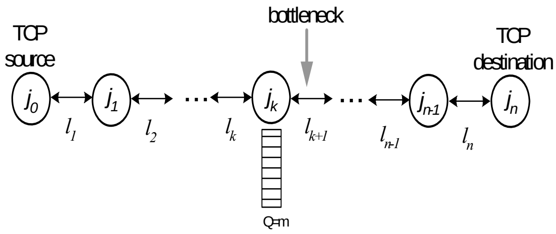

Fig. 2 Network topology

Let all of the TCP segments have the same size and let the acknowledgements travel on the same path as the TCP segments. Let link l_i *have capacity* C_i where $1 < i \leq n$ and all the queues use FIFO discipline. If $C_k > C_{k+1}$ for k, $1 < k \leq n$, then the link l_{k+1} is a bottleneck and the node j_k can drop packets when its queue get congested. We assume only one bottleneck in our model.

For the network model in Figure 1, we assume the following:

- P – size of the packets in bits
- C_i - l_i capacity
- Tq_k(s) – queuing delay for packet s
- Td_i – link traversal delay at l_i
- Tp_i processing time to forward P bits at l_i
- Link l_{k+1} is a bottleneck

The value of the round trip time RTT for a segment s can be calculated as:

$$RTT_s = 2 \times \sum_{i=1}^{n} \left(\frac{P}{C_i} + Td_i + Tp_i + Tq_i^{(s)} \right) \tag{1}$$

The values of Td_i and Tp_i are constant provided there are no changes in the network topology during the connection. Tq_i depend on the number of packets in the queues.

If all queues have infinite capacity, except the queue at j_k, the following holds:

$$RTT_s = 2 \times \sum_{i=1}^{n} \left(\frac{P}{C_i} + Td_i + Tp_i \right) + Tq_k^{(s)} \tag{2}$$

since $Tq_i(s)=0$ for $i \neq k$.

A receipt of a duplicate acknowledgement in a TCP connection means that an out of order segment arrived at the destination. It is safely to assume in all cases that the reason is a lost segment. The duplicate acknowledgements can be treated as loss pairs. When the sequence number of an acknowledgement is s that means that the segment with sequence number s is lost. The segment $s+1$ that generated the duplicate acknowledgement and the lost segment s form a loss pair where $p_1=s$ and $p_2=s+1$.

The round trip time of the duplicate acknowledgement contains considerable and valuable information. The increase of the round trip times is an indication that the bottleneck queue is congested, which is possible to use as an instrument for determining the reason behind the packet loss. Intuitively, when large round trip times follow congestion losses the activation of congestion mechanisms is the appropriate thing to do. In the opposite case, the small round trip times that follow a packet loss signal that (1) the bottleneck queue was not congested when the packet was lost, and (2) the packet was lost on some of the other links on the path.

5. THE REASON BEHIND THE LOSS

The goal of the TCP variations is to determine the nature of the errors that caused packet losses. Simply, when there is congestion on the path apply standard congestion control algorithms, otherwise if the loss is due to a transmission error, then the sender should keep the current sending rate.

The protocol belongs to the family of TCP variations. It observes the sequences of packets that precede and follow a lost packet. If RTT values are between the smallest measured RTT, $RTTmin$, and the maximum threshold $RTTmax$, then it is very likely to have congestion at the bottleneck. The main issue is the definition of $RTTmax$.

The smallest RTT can be expressed as:

$$RTT_{\min} = 2 \times \sum_{i=1}^{n} \left(\frac{P}{C_i} + Td_i + Tp_i \right) \tag{3}$$

when there is no queuing delay and $Tq_k(s)=0$.

The changes of $RTTs$ depend on the changes of Tq_k(s). We have:

$$\overline{RTT} = 2 \times \sum_{i=1}^{n} \left(\frac{P}{C_i} + Td_i + Tp_i \right) + \overline{Tq_k} \tag{4}$$

where $\overline{RTT} = \frac{\sum_{s=1}^{M} RTT_s}{M}$ is the average value of all measured round trip times RTT_s and $\overline{Tq_k} = \frac{\sum_{s=1}^{M} Tq_k^{(s)}}{M}$ is the average queuing delay at j_k for M measured values of RTT_s.

If

$$\Delta_{RTT} = RTT_s - \overline{RTT} \qquad (5)$$

then:

$$\Delta_{RTT} = Tq_k^{(s)} - \overline{Tq_k} \qquad (6)$$

We can conclude that if $\Delta RTT<0$ then $Tq_k(s) < \overline{Tq_k}$, which means that the queuing delay was bellow the average, therefore ***there was no congestion at the bottleneck when packet s was at the bottleneck*** and the congestion window should not be decreased. Finally, we can use $\overline{RTT}$ as the maximum *RTT* threshold *RTTmax*.

6. THE MODIFICATION TO TCP CC

Based on the previous analysis of the RTT behaviour after lost segments, we added two new algorithms to the existing TCP congestion control. The first one is executed every time an acknowledgement arrives at the sender and calculates the Number_of_small_RTTs of RTTs that were smaller than $\overline{RTT}$. The second algorithm runs on the receipt of three duplicate acknowledgements and modifies the fast recovery algorithm in the following manner: if the Number_of_small_RTTs is smaller than Min_small_RTTs, than assume a random loss and leave the congestion window as it is, otherwise there is a congestion and cut in half the size of the current congestion window.

The rationale is based on the observation of a sequence of consecutive packets sent from a window of segments as part of a packet train. If all of them have RTTs smaller than the average before the loss occurred, then the assumption that the queue was not congested when the segment was lost is the correct one.

7. PERFORMANCE

The ns software [16] is used as a network simulation tool. The modification is built into the TCP Reno implementation and is called PR (Prompt Recovery). Across the entire experimental set-up, and hence the experiments, the parameter Min_small_RTTs has value 6. The throughput is calculated by the formula:

$$Throughput = \frac{(Np - Nx)\cdot P \cdot 8}{t \cdot 10^6} \qquad (7)$$

where *Np* is the total number of arrived segments, *Nx* is the number of retransmitted segments, *P* is packet size and *t* is the duration of the connestion.

On Figure 3 the model of the simulated topology is given.

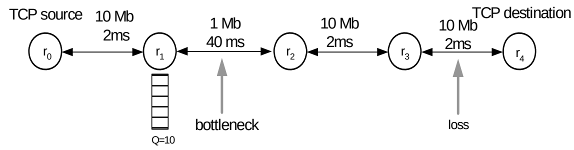

Fig. 3 Simulated network model

At the node n_0 there is a TCP source which sends packets towards the TCP sink at node *r4*. The line r_1-r_2 is a bottleneck because it has ten times smaller capacity then the line r_0-r_1. At the node r_1 there is a FIFO queue with capacity of 10 packets. At the line r_2-r_4 there is an error loss model which generates packet loss with packet loss rate of 1% starting at 10 seconds from the beginning of the simulation. The TCP source starts sending of the packets at time 0. The simulation duration is 30 seconds. The results are given in Table 1 and Table 2.

Protocol	No. of bytes sent	No. of acknowledged packets
PR	3187000	3121
Reno	2669000	2620

Tab. 1 Simulation results - 1

No. of retransmitted packets	Throughput (bps)	Improvement compared to Reno
75	814983.28	131370.24 **(19.2%)**
65	683612.04	

Tab. 2 Simulation results - 2

The sending of the packets in the simulation with Reno compared to the simulation with the modified protocol is shown on Figure 4.

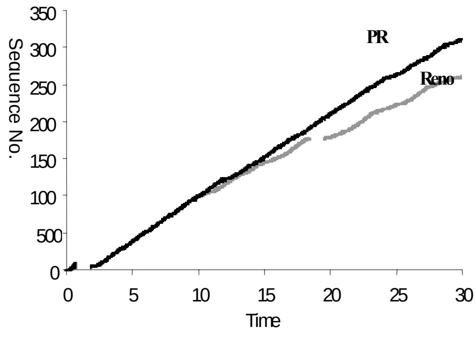

Fig. 4 Packets sending

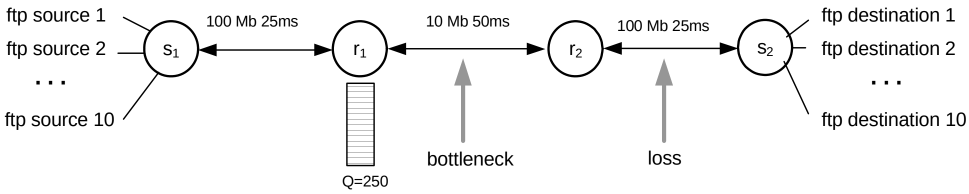

Fig. 5 "Ten connections"

Obviously, after 10 seconds, when the packets drop on the link r_3-r_4 begins, the modification sends packets at higher rate then Reno. It is unequivocal demonstration that the PR related mechanism is an improvement to the TCP congestion control.

The modified protocol was tested in variety of simulation topologies and scenarios. The results show throughput increase in all cases. The modification was also added to the TCP Newreno implementation (PR*) and tested. The improvement of both versions ranges from 10% up to 70% increase which is more than a significant improvement.

In all cases where the loss is due to congestion PR acts the same as Reno. Likewise, the performance of PR* with the absence of random losses is identical to the performance of Newreno.

a. Fairness of PR and PR*

In order to evaluate the performance of the modified protocol when there is a contention of more than one connection for the bottleneck link capacity, a special network topology is generated. The simulations were done with 10 simultaneous connections representing ftp sources. The topology is given in Figure 5.

The simulations had several scenarios with different combinations of protocols. For all of them we calculated the fairness index by the formula [4]:

$$F(x) = \frac{\left(\sum x_i\right)^2}{n\left(\sum x^2{}_i\right)} \qquad (8)$$

When all ten connections use the same protocol, the values of the fairness index are given in Table:

Reno	PR	PR*
0.670509	0.685728	0.757788

Tab. 3 "Fairness index"

On Figure the values of the fairness index during the simulation time for the three protocols are compared.

Both modifications, PR and PR* indicate a high degree of fairness. This was somewhat expected since they deploy the AIMD congestion control.

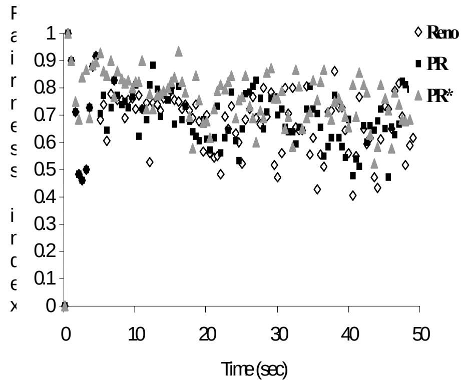

Fig 6 "Fairness with ten connections"

8. CONCLUSION

The introduction of wireless and mobile technologies places new demands to the existing Internet architecture designed mainly for a wired transmission medium.

The paper looks into a novel congestion control mechanism for improvement of TCP operation in heterogeneous networks. It is based on the loss pair method developed as a tool for network probing ands exploration based on E2E measurements. The proposed solution is a member to the TCP modifications family. Hence, it does not require changes to the other Internet layers, preserves the E2E semantics, transparency, modularity, and the structure of TCP.

The modification consists of two new algorithms. The first one constantly measures the round trip times of the received packets and calculates a round trip time threshold $\overline{RTT}$. The second one affects the fast recovery algorithm. When three duplicate acknowledgements arrive, if a number of consecutive round trip times are below the given value, a random loss is assumed and the congestion window stays the same.

Both modified Reno and Newreno have been subjects to intensive simulations with ns. The results show significant and very promising improvements concerning the throughput in the range from 10% to 70%.

The research continues both on the experimental level with various network parameters, and on the theoretical level where we are trying to provide an explicit mathematical model, which will be further studied with respect to its validation and verification.

REFERENCES

[1] Bakre, A. and Badrinath, B., “I-TCP: Indirect TCP for Mobile Hosts” *Proc. of the IEEE ICDCS ’95*, pp. 136–143, 1995

[2] Balakrishnan, H., Seshan, S., Amir, E., Katz, R., “Improving TCP/IP Performance over Wireless Networks”, *Proc. of the 1st ACM International Conference On Mobile Computing and Networking (MOBICOM),* Nov 1995

[3] Brakmo, L. S. and Peterson, L. L., “TCP Vegas: End to End Congestion Avoidance on a Global Internet”, *IEEE Journal on Selected Areas of Communication*s, 1995

[4] Chiu, D. M., Jain, R. “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks”, *Computer Networks and ISDN Systems*, Vol. 17, pp. 1-14, 1991

[5] Goff,T., Moronski, J., Phatak., D., “Freeze-TCP: A True End-to-End Enhancement Mechanism for Mobile Environments “, *In Proceedings of the INFOCOM*, 2000

[6] Haas, Z., Agrawal, P., “Mobile-TCP: An Asymmetric Transport Protocol Design for Mobile Systems”, *In Proceedings of the IEEE International Conference on Communications (ICC’97),* 1997

[7] Liu, J., Crovella, M., “ Using loss pairs to discover network properties”, *ACM SIGCOMM Internet Measurment Workshop*, San Francisco, CA, Nov 2001

[8] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M., Y. ,Wang, R., "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links", *In Proceedings of ACM Mobicom 2001*, pp 287-297, Rome, Italy, July 16-21 2001

[9] Parsa, C., Garcia-Luna-Aceves, J., J., “Improving TCP Congestion Control over Internets with Heterogeneous Transmission Media,” *Proc. IEEE ICNP ‘99*, Toronto, Oct. 1999

[10] Parsa, C., Garcia-Luna-Aceves, J., J., “ TULIP: A Link-Level Protocol for Improving TCP over Wireless Links”

[11] Pentikousis, K., “TCP in Wired-cum-Wireless Environments”, *IEEE Communications Surveys*, http://www.comsoc.org/pubs/surveys, Fourth Quarter 2000

[12] Ratnam, K. and Matta, I., “WTCP: An Efficient Mechanism for Improving TCP Performance over Wireless Links”, *Proc. of the Third IEEE Symposium on Computer and Communications (ISCC ’98*), Athens, Greece, Jun 1998

[13] Stojcevska, B. "Improving TCP performance in heterogeneous networks", Master thesis, Skopje, Macedonia, 2003

[14] Stojcevska, B., Popov, O. B., “A Loss Pair Approach towards Congestion Management”, *Proc. of the Third CiiT*, Bitola, Macedonia, Dec 2003

[15] Stojcevska, B., Popov, O. B., “Loss Pairs for Congestion Detection and Control”, Proc. of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics SCI2004, Orlando, Florida, June 2004

[16] The Network Simulator - ns-2, the VINT project , http://www.isi.edu/nsnam/ns/

[17] Tsaoussidis, V. and Badr, H., “TCP-Probing: Towards an Error Control Schema with Energy and Throughput PerformanceGains”, *Proc. of the 8th IEEE International Conference on Network Protocol*s, Osaka, Japan, 2000

[18] Vaidya, N., H., *et al.*, “Delayed Duplicate Acknowledgements: A TCP-unaware Approach to Improve Performance of TCP over Wireless,” Technical Report 99-003, Computer Science Department, Texas A&M University, Feb 1999

[19] Zhang, C., Tsaoussidis, V., “TCP Real: Improving Real-time Capabilities of TCP over Heterogeneous Networks”, *Proc. of the 11th IEEE/ACM NOSSDAV 2001,* New York 2001

BIOGRAPHIES

Biljana Stojcevska is a research and teaching fellow, as well as a PhD student, at the Department of Computer Science at the Faculty of Natural Sciences and Mathematics, Saints Cyril and Methodius University, in Skopje, Macedonia. Her research interests include TCP/IP for high performance networking, Internet traffic models, application programming, and ICT4D.

Oliver B. Popov is a professor of Computer Science in the Department for Information Technology and Media at Mid Sweden University, and an associate professor in the Computer and Systems Sciences Department at Stockholm University and KTH in Sweden. His research inclinations converge towards computer networks, Internet technology, ICT4D, research and education networking, and Information Society.