TCP IN HETEROGENEOUS ENVIRONMENTS

B. Stojčevska, O. B. Popov

Institute of Informatics, Faculty of Natural Sciences and Mathematics
Sts. Cyril and Methodius University
Arhimedova bb, P.O.Box 162, Skopje, Macedonia
biljanas@ii.edu.mk, oliver.popov@mh.se

Abstract: TCP is the premier transport protocol for a large number of applications on the Internet. The original protocol and many of its subsequent modifications were based on the wireline infrastructure. This implies, particularly with the evolution of the physical media, a nearly error free transmission. The introduction of wireless settings mostly invalidates the last statement due to the high BER, disconnections, and bandwidth limitations. Hence, TCP experiences an impaired performance mostly affected by random losses. One of the major reasons are congestion mechanisms triggered as a response to losses, and not the retransmission of lost packets, through which TCP tries to attain reliability. Clearly, innovative methods and strategies are needed for dealing with these phenomena in heterogeneous environments, which are a combination of wireline and wireless networks. In the article, we analyze the nature of the problems for TCP in heterogeneous systems, and provide an overview and evaluation of the proposed solutions.

Keywords: TCP, wireless TCP, transport protocols

1. Introduction

IP is a protocol of integration. Its potential to connect different physical networks, which makes it technology independent, robust and immensely scalable is behind the tremendous growth of the Internet. Nevertheless, the development of wireless and mobile communications has made it necessary to consider some of the components of the Internet protocol for possible modifications and adaptations in order to accommodate the reality of the heterogeneous infrastructure.

One of those members in the IP family is Transmission Control Protocol (TCP), which is a dominant transport protocol on the Internet [11]. Over the years, as a connection-oriented and reliable protocol, TCP has been widely modified and well tuned for wireline networks. The original design, with the exception of additions that improved the performance over satellite links, did not address the particular characteristics of the wireless and mobile networks. Those include, but are not limited to, frequent random errors result that impair its performance. In this case, TCP usually assumes that a packet loss indicates network congestion. Consequently, it activates congestion control algorithms that reduce the size of its congestion window and severely cut down the throughput and efficiency of resource utilization.

The confusion between the source of the error, transmission or congestion, which is indistinguishable for the transport protocol, is indeed behind the problem. Hence, if the error is of a transmission nature, any contraction of the window is inappropriate. The final outcome of this behavior is poor overall performance of the network. Hence, it is necessary to provide the protocol with new mechanisms, especially with ones that deal with random errors.

The numerous proposals to deal with the problem can be generally classified in three categories: link-layer, split connection, and TCP variations. The link-layer solutions try to hide the errors of the wireless medium from the upper layers by using local retransmissions and FEC. The main issue here is whether the link-layer protocol should be TCP-aware or not. In the split connection approach the TCP connection is divided in two parts: a wired and a wireless connection. In a way by using a protocol tailored to wireless transmission at the wireless par, using a protocol tailored to wireless transmission at the wireless par localizes the consequences of the problem. The TCP modifications introduce new mechanisms to the standard TCP protocol, leaving other layers unaffected. The main advantage is that they preserve modularity of the protocol stack and do not infringe end-to-end semantics of TCP.

2. TCP in wireless environments

2.1 TCP response to loss error

TCP assumes that a packet loss signifies congestion on the path where the connection takes place. In order to avoid further growth of the congestion it reacts to packet losses by decreasing its congestion window. TCP detects a packet loss in two ways: a timeout and duplicate acknowledgements. If the loss is indicated by a timeout, TCP sets the value of cwnd to one segment and enters a slow-start 6 phase. When the loss is detected by a receipt of three duplicate acknowledgements, then the answer is fast retransmit and fast recovery which set the value of cwnd to one half of its value. The end result is reduction in the size of the congestion window whenever a packet loss is detected. If the packet loss is due to a transmission error, any contraction of the window is inappropriate. The final outcome of this behavior is decreased data throughput, inefficient resource utilization and eventually a poor performance of the applications.

2.2 TCP in networks with wireless links

The properties of wireless channels are characterized with frequent random losses. The transmission errors can be caused by physical obstacles, interference, shadowing, signal fading etc. The bit-error rate (BER) 10 for wireless medium is in order of 10^{-3}, which translates as packet loss rate of 12 percent, for packets with size of 1500 bytes. To compare with, the BER for wired medium is in the order 10^{-6} to 10^{-8}, which translates as 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 moves from one cell to another, connection handoff is performed during which packet losses can occur.

Of all the problems TCP faces in wireless networks, its performance is mostly affected by random losses. The reason is not the retransmission of the lost packets itself, but the congestion algorithms that are triggered as a response to the losses, when there is no congestion on the path. The final outcome of this behavior is decreased data throughput, inefficient resource utilization and eventually a poor performance of the applications.The reduction of the transmission rate leads to severe throughput degradation. Introducing 2% of packet loss, on a path where a TCP Reno 7 connection takes place, can lead 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.

3. Proposed solutions

The ideal TCP behavior in case of a random loss would be retransmission of the lost segment without activating congestion control. The ideal network corrective action would be taking care of the loss transparently to the TCP. The usual approach for solving this problem is trying to accomplish the ideal behavior either of TCP or of the network.

During the past few 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 10.

3.1 Link-layer schemes

The link-layer schemes 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 also increases the computation complexity and consequently also the energy consumption. Local retransmissions at the data link layer are used to recover the TCP connection from loss errors. Figure 1 illustrates the process of data link local retransmissions in a typical wired/wireless setting.

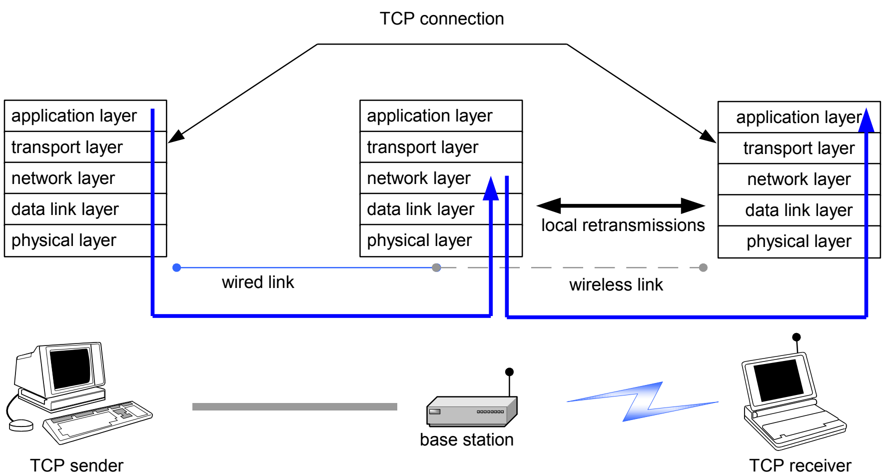

Figure 1: Local retransmissions

3.1.1 TCP aware protocols

The link-layer solutions in which the link layer protocol is aware of the TCP connection are known as TCP aware protocols. This knowledge can be used to hide the wireless errors from TCP.

One of the earliest proposals in this class is the snoop protocol 3. A snoop agent resides at the base station and the snoop protocol runs at the mobile host. The agent intercepts the segments destined for the mobile clients in its cell and buffers all unacknowledged segments. Snoop uses local retransmissions and timeout mechanism for wireless error recovery. If there is a duplicate acknowledgement from the mobile host, snoop suppresses it and retransmits the lost segment. In that way, the connection is recovered from the loss without activating fast retransmit at the sender.

A similar scheme is used in WTCP 13. This protocol employes timestamp option to estimate RTT. It buffers the segments sent to the mobile clients and records their arrival times. Then it adds the base station residence time to the timestamp to avoid enlarging RTO due to the local retransmissions.

3.1.2 TCP unaware protocols

The TCP unaware link layer protocols have no knowledge about the transport protocol that uses their service. These protocols can offer service to transport protocols other than TCP.

TULIP 9 is designed for half duplex wireless channels with limited bandwidth. The protocol requires to be informed if the ongoing transport service is reliable or not. TULIP tries to accomplish in-order delivery to prevent generating of duplicate acknowledgements and triggering fast retransmit and fast recovery. It uses ARQ and timeout mechanism for local recovery. TULIP is not TCP aware, but it violates the modularity principle because it requires information from the network protocol.

The Delayed Duplicate Acknowledgement protocol 15 is TCP unaware protocol that performs similar to snoop. For proper protocol operation, every segment has to be encapsulated in separate frame. The same is for the acknowledgements. The reason is that DDA uses its own sequence numbers. If a loss error is detected, the base station delays the sending of the third duplicate acknowledgement d seconds and retransmits the lost segment. The idea is to prevent the sender to go into fast retransmit. The authors report that the protocol performs well in the same conditions as snoop. The estimation of the parameter d is still an open problem.

3.2 Split connection

The split connection schemas 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, which connects the wired, and the wireless part of the network. A protocol tailored to wireless transmission is usually used at the wireless part. The layout of this scheme is given in Figure 2.

The base station acknowledges every received segment, without waiting for it to arrive at the mobile host. The biggest deficiency of this behavior is the possibility of the following: an acknowledgement for a segment may arrive at the sender before the segment actually arrives at the mobile receiver. In that way, the end-to-end semantics of TCP is violated.

One of the earliest split connection proposals is I-TCP 1. The base station maintains one TCP connection with the fixed host and uses another protocol for communication with the mobile host, which is designed for wireless channels. The base station sends acknowledgements for the received segments as soon as they arrive and this violates the TCP end-to-end semantics. Because of that, the authors recommend the protocol for applications that don’t use TCP acknowledgements, but have their own acknowledging mechanism.

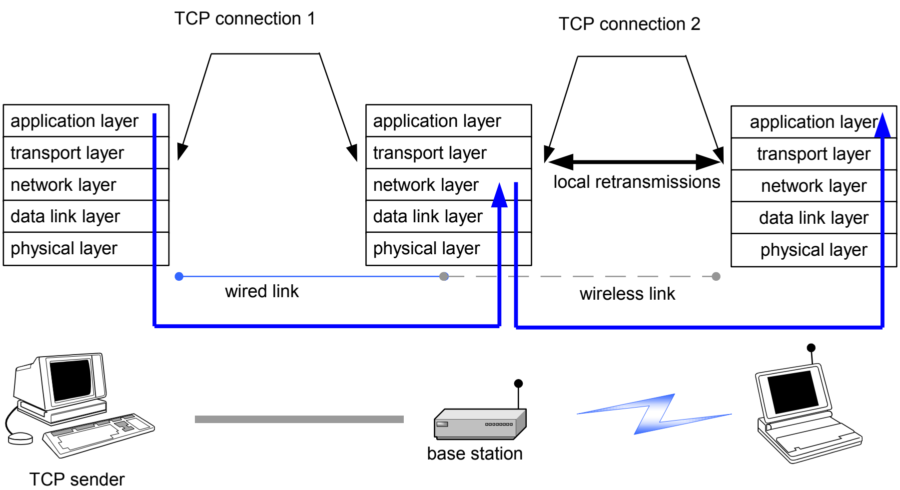

Figure 2: Split connection

Mobile-TCP 5 is an asymmetric transport protocol for mobile hosts. The main goal in its design is the energy efficiency of the protocol. The protocol at the wireless part uses selective reject from the mobile host to the base station and go-back-n in the opposite direction. It has reduced computation complexity with placing lower load at the mobile device. Compression of the header is used and to reduce the amount of the wireless transmitted data. Timeout mechanism is used only at the base station. Delaying the acknowledgements until the segments arrive at the mobile host preserves the semantics of the TCP protocol.

3.3 TCP variations

The TCP variations are in fact TCP modifications that introduce new functionality to the protocol. They always maintain TCP end-to-end semantics and are also named end-to-end protocols.

The basic problem for TCP in heterogeneous networks is interpreting random losses as indication for congestion. For that reason, the main focus for the TCP variations is finding a way to determine the nature of the errors that caused packet losses. If congestion on the path is concluded, the protocol should apply standard congestion control algorithms. If the reason for the loss is found to be a transmission error, then the sender should keep the current sending rate.

3.3.1 Implicit calculation

The protocols with implicit calculation of the reason for the packet loss typically use heuristic methods usually based on RTT measurements, congestion window sizes or packet loss patterns. These solutions require only changes to TCP and no changes to other layers.

The Tri-S 16 protocol is primarily oriented towards improved congestion control. It uses changes in the throughput as an indication for congestion. A throughput gradient is calculated as:

$$TG(W_n) = \frac{T(W_n) - T(W_{n-1})}{W_n - W_{n-1}} \quad (1)$$

where T(Wn) is the throughput for the n-th window W_n. Then, the normalized throughput gradient (NTG) is calculated as

$$NTG(W_n) = \frac{TG(W_n)}{TG(W_1)} \quad (2)$$

NTG takes values in the interval [0, 1]. Values of NTG close to 0 indicate congestion. The protocol uses two thresholds, NTGd and NTGi, to differentiate between congested and uncongested state. The slow-start and congestion avoidance are replaced with new algorithms. On connection initialization the window is increased for one basic adjustment unit (BAU) on each received acknowledgement. When the value of NTG is bigger than NTGd, the sending window is increased for BAU/W on each acknowledgement. If the value of NTG is less than NTGd, the sending window is decreased for one BAU.

TCP Vegas 4 was designed having wired network and efficient network utilization in mind. It has new more efficient retransmission mechanism, which performs faster recovery than fast retransmit. To avoid congestion, Vegas modifies Reno's slow-start and congestion avoidance algorithm. The expected throughput is calculated by:

$$Expected = \frac{WindowSize}{BaseRTT} \quad (3)$$

where BaseRTT is the smallest measured RTT value. The difference between the actual and expected throughput is compared to thresholds α and β. When the difference is smaller than α the window size is increased linearly. If the difference is bigger than β the window size is decreased linearly. In fact, in this case congestion is assumed. α and β can be interpreted as the smallest and the biggest number of buffers that the connection should occupy at the intermediate router.

TCP Westwood 8 is TCP Reno modification with new congestion control algorithm. By measuring the arrival rate of the acknowledgement Westwood calculates bandwidth estimation, BWE, and then after a packet loss, the value of the congestion threshold is determined as:

$$sstresh = \frac{BWE * RTTMin}{seg\_size} \tag{4}$$

instead of being set to half the value of the congestion window unconditionally. In that way, the protocol accomplishes recovery faster because the congestion window grows more quickly. The protocol is tested in experimental wireless test bed and in simulated environment. Compared to Reno, it shows significant throughput improvement, but not when the loss rate exceeds few percent.

3.3.2 Explicit notification

TCP doesn't use information about the state of the network from the lower layers and it relies on a network model with FIFO queues at the routers. With the adding an explicit congestion notification (ECN) to IP 12 an active queue management scheme is proposed. Routers detect congestion before the queues overflow and then they set the congestion experienced (CE) bit to the IP header. This information can be used by TCP for congestion control. If a segment is lost while there is no congestion signal, the sender can assume that there is no congestion and keep the current transmission rate.

Explicit loss notification (ELN) 2 can be added in wireless networks. When the TCP sender is explicitly informed that a segment is lost because of errors on the wireless link, it can recover without performing congestion control. In TCP ELN is simulated and perfect network knowledge about the losses is assumed. It is difficult to identify which packets are lost because of errors on a lossy link so implementing this scheme in practice may be not completely feasible. However, this scheme can be used to determine what percent of the performance degradation is due to the inappropriate congestion control invocation.

3.3.3 Network probing

Error control with energy and throughput performance gains are the primary goals in the design of TCP Probing 14. It is a TCP modification, which includes a probing device. When a segment loss is detected, the protocol goes into a probing cycle in which the sending of the segment is stopped. During the probing cycle two short probing segments are sent and their RTTs are measured. If they are both in the interval *[bestRTT, lastRTT],* it is concluded that there is no congestion and the sender performs immediate recovery. If not, it performs slow-start. The modification improves TCP throughput in presence of random errors and is energy efficient. But, when the error rate is higher than 10%, the performance drops because the probing cycles increase the connection time. Another problem is the handling of incoming data segments in a two-way data exchange during the probing cycles.

4. Comparative analysis

The biggest advantage of the link-layer protocols is that they achieve most efficient loss recovery. The biggest open question is whether the protocol should be TCP aware or not. The TCP aware solutions perform better because they can intervene in the error recovery. It is not clear which should be the choice of the error detection mechanism (timeout, sequence numbers…). A segment out of order can trigger fast retransmit and decrease of the congestion window, so link-layer reordering is recommended. There is also the question of TCP awareness of the link layer. The results indicate that the TCP aware protocols achieve better results, but the TCP unaware protocols can offer service to other transport protocols than TCP.

The split connection approach localizes the management of the wireless segment. The major problem in this approach is the violation of the TCP end-to-end-semantics. It also has bad performance when there are handoffs to other base stations.

The biggest advantage of the TCP variations is that they maintain the TCP end-to-end-semantics and the modularity of the network design. Sometimes they require support from the network layer. But, they achieve lower throughput increase than the other techniques. Very few of them have mechanisms for handling handoffs, so they don't perform well when there is high mobility.

In Table 1 the characteristics of all presented protocols are summarized.

5. Conclusion

The paper deals with the analysis and review of the TCP behavior and the myriad of modifications in a heterogeneous environment. Basically, the modifications are trying to reconcile the connection-oriented and reliable nature of TCP with the need to exhibit flexibility and diversity in response to events when wireless links and mobile devices are present on the network. The selfclocking protocol, originally conceived for wireline systems, empowered with a lot of network-friendliness, becomes almost inefficient in wireless settings.

Approach	Protocol	Mechanism	Advantages	Disadvantages
Link layer	Snoop WTCP TULIP DDA	Sniffing the packets at the data link layer and buffering at the base station; local retransmissions	Fast error recovery transparent to the end-to-end connection; localizing of the wireless segment; no changes at the transport layer	problems with encrypted transport
Split connection	I-TCP Mobile-TCP	Spliting the connection in wired and wirelless part	localizes the management of the wireless segment; maintaining the modularity	violation of the TCP end-to-end-semantics; inefficient handoff management
TCP variations	Vegas Westwood ECN ELN Probing	implicit calculation of the reason for the packet loss based on heuristics	maintain the TCP end-to-end-semantics and the modularity of the network design	Don't achieve big increase in throughput compared to other solutions

Table 1: Protocol characteristics

The problem is with the assumption the protocol makes whenever a packet is not acknowledged. Namely, in this case TCP triggers immediately congestion control mechanisms. While this might be an appropriate course of action in wired networks, it certainly not suitable for wireless links, with a higher error bit rate, frequent handovers and interference.

Most of the traditional solutions have been presented observing the commonly acceptable classification in three categories: link-layer, split connection and TCP variations. All these modifications have their pros and cons, ranging from better throughput, to semantical clarity and the preservation of the integrity of the TCP/IP communication model.

One way to resolve at least some of the problems is to give all the responsibility to the link layer whenever the BER is the reason behind packet loss. High delays and jitter are certainly in favor of mechanisms that might be application dependent. If handovers are involved then the fine and coarse granularity of TCP congestion control becomes an obstacle to dissent network performance.

Split connections, while having a syntactical simplicity, are not quite in the spirit of the Internet purity and break the end-to-end semantics. The last category that covers the variations of TCP, while may be not so efficient, are actually what TCP is all about - to resolve the problems within its framework, while maintaining the original specifications and intentions of the protocol.

The article shows that inducing the best behavior of TCP in heterogeneous networks is neither an easy problem nor a closed one. In addition, the complexity stems from the inherit diversity and functionality of the transport layer, the need for technological independence and scalability. It is also an argument why the research on the transport layer protocols, and in particular TCP, is simpliciter not only the thing of the present, but also very much of the future in networking.

6. References

1. Bakre, A. and Badrinath, B., "I-TCP: Indirect TCP for Mobile Hosts" *Proc. of the IEEE ICDCS '9*5, pp. 136–143, 1995
2. Balakrishnan, H., Katz, R., "Explicit loss notification and wireless web performance", *IEEE Globecom Internet Mini-Conference*, Sydney, Oct 1998
3. 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
4. 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
5. 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
6. Jacobson, V., "Congestion Avoidance and Control", *Proc. of the ACM SIGCOMM Symposium on Communications Architectures and Protocol*s, pp. 314–329, Stanford, USA,14-17 Aug 1988
7. Jacobson, V., "Modified TCP Congestion Avoidance Algorithm,". Technical report, 30 Apr 1990
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., " TULIP: A Link-Level Protocol for Improving TCP over Wireless Links"
10. Pentikousis, K., "TCP in Wired-cum-Wireless Environments", IEEE Communications Surveys, http://www.comsoc.org/pubs/surveys, Fourth Quarter 2000
11. Postel, J., "Transmission Control Protocol", *RFC 793*, Sep 1981
12. Ramakrishnan, K., Floyd, S., Black, D., "The Addition of Explicit Congestion Notification (ECN) to IP". RFC 3168, Sep 2001
13. 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
14. 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
15. 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
16. Wang, Z., Crowcroft, J., "A New Congestion Control Scheme: Slow Start and Search (Tri-S)", ACM Computer Communication Review, vol. 21, pp 32-43, Jan 1991