SIMPLE MODEL FOR TRANSMISSION CONTROL PROTOCOL (TCP)

Irma Aslanishvili, Tariel Khvedelidze

***Abstract**: Ad hoc Networks are complex distributed systems that consist of wireless mobile or static nodes that can freely and dynamically self-organize [Namicheishvili et al, 2011]. In this way they form arbitrary, and temporary “Ad hoc” networks topologies, allowing devices to seamlessly interconnect in areas with no pre-existing infrastructure. Introduction of new protocols such as Bluetooth, IEEE 802.11and Hyperlink are making possible the deployment of ad hoc networks for commercial purposes. TCP (Transmission Control Protocol) was designed to provide reliable end-to end delivery of data over unreliable networks. In theory, TCP should be independent of the technology of the underlying infrastructure. In particular, TCP should not care whether the Internet Protocol (IP) is running over wired or wireless connections. In practice, it does matter because most TCP deployments have been carefully designed based on assumptions that are specific to wired networks. Ignoring the properties of wireless transmission can lead to TCP implementations with poor performance. In practice, most TCP deployments have been carefully designed in the context of wired networks. In order to adapt TCP to the ad hoc environment, improvements have been proposed in the literature to help TCP to differentiate between the different types of losses. Indeed, in mobile or static ad hoc networks losses are not always due to network congestion, as it is mostly the case in wired networks. In this paper, we present model: how TCP can be affected by mobility and lower layers protocols.*

***Keywords**: Ad Hoc Networks, protocols, Routing protocols, packet, source node, Relay routing, finite memory, TCP, Simulation Metrics, layers protocols TCP Feedback, TCP-DOOR.*

Introduction

The Transmission Control Protocol (TCP) was designed to provide reliable end-to-end de-livery of data over unreliable networks. In practice, most TCP deployments have been carefully designed in the context of wired networks. Ignoring the properties of wireless ad hoc Networks [Aslanishvili, 2014] can lead to TCP implementations with poor performance. In order to adapt TCP to the ad hoc environment, improvements have been proposed in the literature to help TCP to di errant types of losses. Indeed, in mobile or static ad hoc networks losses are not always due to network congestion, as it is mostly the case in wired networks. In this Chapter, we present an overview of this issue and a detailed discussion of the major factors involved. In particular, we show how TCP can be an acted by mobility and lower layers protocols. In addition, we survey the main proposals which aim at adapting TCP to mobile and static ad hoc environments.

In ad hoc networks, the principal problem of TCP lies in performing congestion control in case of losses that are not induced by network congestion. Since bit error rates are very low in wired networks, nearly all TCP versions nowadays assume that packet losses are due to congestion. Consequently, when a packet is detected to be lost, either by timeout or by multiple duplicated ACKs, TCP slows down the sending rate by adjusting its congestion window. Unfortunately, wireless networks sure from several types of losses that are not related to congestion, making TCP not adapted to this environment. Numerous en-Han cements and optimizations have been proposed over the last few years to improve TCP performance over one-hop wireless (not necessarily ad hoc) networks.

Simple model for Transmission Control Protocol (TCP)

In the authors report, simulation results on TCP throughput in a static linear multi-hop chain, where IEEE 802.11 protocol is used. In Figure1, we display a multi-hop chain of N nodes. It is expected that, as the number of hops increases, the spatial reuse will also increase. However, simulation results indicate that TCP throughput decreases rapidly up to a point as the number of hops increases. It is argued that this decrease is due to the hidden terminals problem, which increases frames collisions. After a repeated transmission failure the MAC layer will react by two actions. First, the MAC will drop the head-of-line frame destined to the next hop we note that this type of drops is known also as drops due to contention on wireless channel. Second, the MAC will notify the upper layer about a link failure. When the routing protocol of a source node detects a routing failure, it will initiate a route re-establishment process. In general the route re-establishment duration is greater than the retransmission timer of the TCP agent; hence the TCP agent will enter the back procedure and will set its congestion window to 1. Also, as TCP sender's does not have indications on the route re-establishment event, TCP will sure from a long idle time. During this time, the network may be connected again, but TCP is still in the back state.

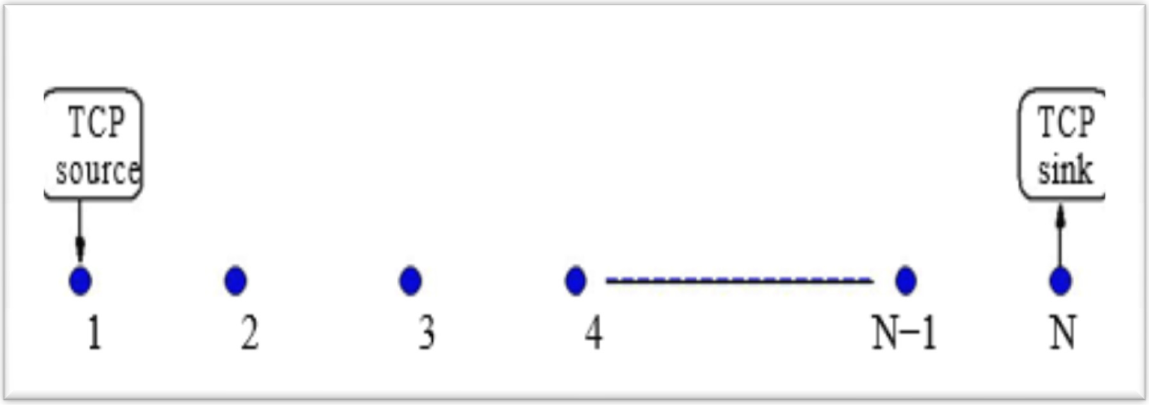

Figure 1. TCP static linear multi-hop chain

Proposals to improve TCP performance in ad hoc networks

We present the various proposals which have been made in the literature to improve the performance of TCP in ad hoc networks. We regroup these proposals to four sets according to the four problems. These four problems are:

1. TCP is unable to distinguish between losses due to route failures and those due to network congestion;
2. Frequent route failures,
3. Contention on wireless channel;
4. TCP unfairness.

We note that problems (1) and (2) are the main causes of TCP performance degradation in MANETs. However, problems (3) and (4) are the main causes of TCP performance degradation in SANETs. Figure 2 shows general classifications of each set of proposals. We classify the proposals that belong to the same set to two types: cross layer proposals, and layered proposals. The cross layer proposals rely on interactions between two layers of the Open System Interconnection (OSI) architecture. These proposals were motivated by the fact that providing lower layer information's to upper layer should help the upper layer to perform better. Thus, depending on between which two OSI layers there will be information's exchange, cross layer proposals can be further classed to four types: TCP and network, TCP and link, TCP and physical, and network and physical. Layered proposals rely on adapting OSI layers independently of other layers. Thus, depending on which layer is involved, layered proposals can be further classed to three types: TCP layer, network layer, and link layer proposals.

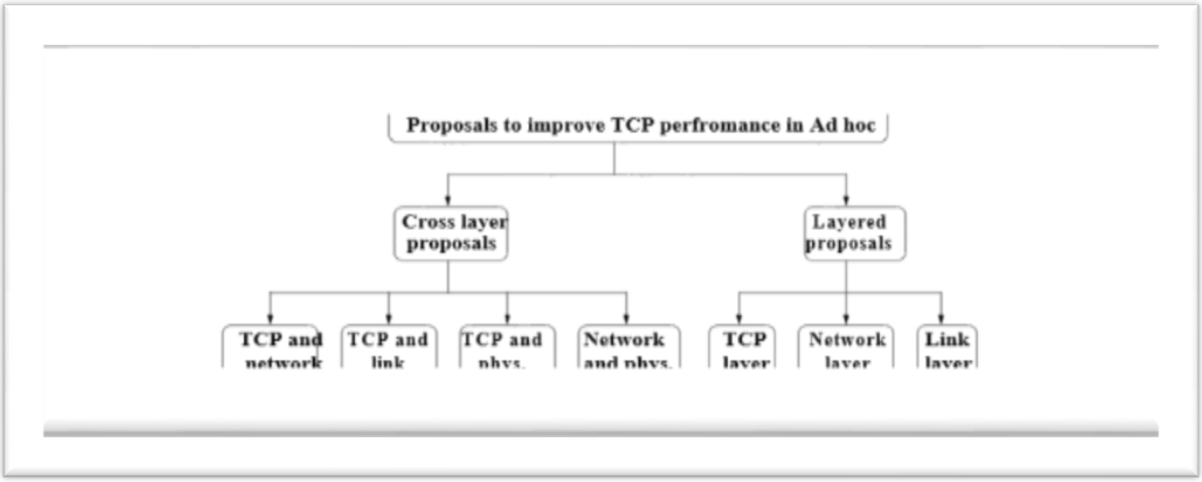

Figure 2. Classification of proposals to improve TCP performance in ad hoc networks

In general, cross layer solutions report better performance than layered solution. But layered solutions respect the concept of designing protocols in isolation, thus they are considered as long term solutions. So, to choose between cross layer and layered solutions we have to answer RST what is prior for us, performance optimization or architecture Performance optimization can lead to short term gain, while architecture is usually based on longer term considerations. In addition, cross layer solutions are more complex to implement and to design than layered solutions. The reason is that the implementation of cross layer solutions requires at least medications of two OSI layers, and their design requires that the system is considered in its entirety.

TCP and network cross layer proposals

This metric **TCP-F: (**TCP Feedback) is a feedback based approach to handle route failures in MANETs. This approach allows the TCP sender to distinguish between losses due to routes failure and those due to network congestion. When the routing agent of a node detects the disruption of a route, it explicitly sends a Route Failure Notification (RFN) packet to the source. On receiving the RFN, the source goes into a snooze state. A TCP sender in snooze state will stop sending packets, and will freeze all its variables, such as timers and congestion window size. The TCP sender remains in this snooze state until it is notified of the restoration of the route through Route Re-establishment Notification (RRN) packet. On receiving the RRN, the TCP sender will leave the snooze state and will resume transmission based on the previous sender window and timeout values. To avoid blocking in the snooze state, the TCP sender, on receiving RFN, triggers a route failure timer. When this timer expires the congestion control algorithm is invoked normally. The simulation scenario is basic and is not based on an ad hoc network. Instead, they emulate the behavior of an ad hoc network from the viewpoint of a transport layer.

ELFN-based technique: Explicit Link Failure Notification technique is similar to TCP-F. However in contrast to TCP-F, the evaluation of the proposal is based on a real interaction between TCP and the routing protocol. This interaction aims to inform the TCP agent about route failures when they occur. The authors use an ELFN message, which is piggy-backed on the route failure message sent by the routing protocol to the sender. The ELFN message is like a host unreachable Internet Control Message Protocol (ICMP) message, which contains the sender receiver addresses and ports, as well as TCP packet's sequence number. On receiving the ELFN message, the source responds by disabling its retransmission timers and enters a standby mode. During the standby period, the TCP sender probes the network to check if the route is restored. If the acknowledgment of the probe packet is received, the TCP sender leaves the standby mode, resumes its retransmission timers, and continues the normal operations. The ELFN message is like a host unreachable Internet Control Message Protocol (ICMP) message, which contains the sender receiver addresses and ports, as well as TCP packet's sequence number. On receiving the ELFN message, the source responds by disabling its retransmission timers and enters a standby mode. During the standby period, the TCP sender probes the network to check if the route is restored. If the acknowledgment of the probe packet is received, the TCP sender leaves the standby mode, resumes its retransmission timers, and continues the normal operations.

In the mentioned reference, the authors study the act of varying the time interval between probe packets. Also, they evaluate the impact of the RTO and the Congestion Window (CW) upon restoration of the route. They end that a probe interval of 2 sec. performs the best, and they suggest making this interval a function of the RTT instead of giving it an axed value. For the RTO and CW values upon route restoration, they end that using the prior values before route failure performs better than initializing CW to 1 packet and/or RTO to 6 sec., the latter value being the initial default value of RTO in TCP Reno and New Reno versions. This technique provides significant enhancements over standard TCP, but further evaluations are still needed. For instance, different routing protocols should be considered other than the reactive protocol DSR considered, especially proactive protocols such as OLSR.

ATCP: Ad hoc TCP utilizes network layer feedback too. In addition to the route failures, ATCP tries to deal with the problem of high Bit Error Rate (BER). The TCP sender can be put into persist state, congestion control state or retransmit state. A layer called ATCP is inserted between the TCP and IP layers of the TCP source nodes. ATCP listens to the network state information provided by ECN (Explicit Congestion Notification) messages and by ICMP Destination Unreachable message; then ATCP puts TCP agent into the appropriate state. On receiving a Destination Unreachable message, TCP agent enters a persist state. The TCP agent during this state is frozen and no packets are sent until a new route is found by probing the network. The ECN is used as a mechanism to explicitly notify the sender about network congestion along the route being used. Upon reception of ECN, TCP congestion control is invoked normally without waiting for a timeout event. To detect packet losses due to channel errors, ATCP monitors the received ACKs [Aslanishvili, 2012]. When ATCP sees that three duplicate ACKs have been received, it does not forward the third duplicate ACK but puts TCP in the persist state and quickly retransmits the lost packet from TCP's buyer. After receiving the next ACK, ATCP will resume TCP to the normal state. Note that ATCP allows interoperability with TCP sources or destinations that do not implement ATCP.

Split TCP: TCP connections that have a large number of hops sure from frequent route failures due to mobility. To improve the throughput of these connections and to resolve the unfairness problem, the Split TCP scheme was introduced to split long TCP connections into shorter localized segments see Figure 3. The interfacing node between two localized segments is called a proxy. The routing agent decides if its node has the role of proxy according to the inter-proxy distance parameter. The proxy intercepts TCP packets, buyer them, acknowledges their receipt to the source (or previous proxy) by sending a local acknowledgment (LACK). Also, a proxy is responsible for delivering the packets, at an appropriate rate, to the next local segment. Upon the receipt of a LACK (from the next proxy or from the destination), a proxy will purge the packet from its buyer. To ensure the source to destination reliability, an ACK is sent by the destination to the source similarly to the standard TCP. In fact, this scheme splits also the transport layer functionalities into those end-to-end reliability and congestion control. This is done by using two transmission windows at the source which are the congestion window and the end-to-end window. The congestion window is a sub-window of the end-to-end window. While the congestion window changes in accordance with the rate of arrival of LACKs from the next proxy, the end-to-end window will change in accordance with the rate of arrival of the end-to-end ACKs from the destination. At each proxy, there would be a congestion window that would govern the rate of sending between proxies.

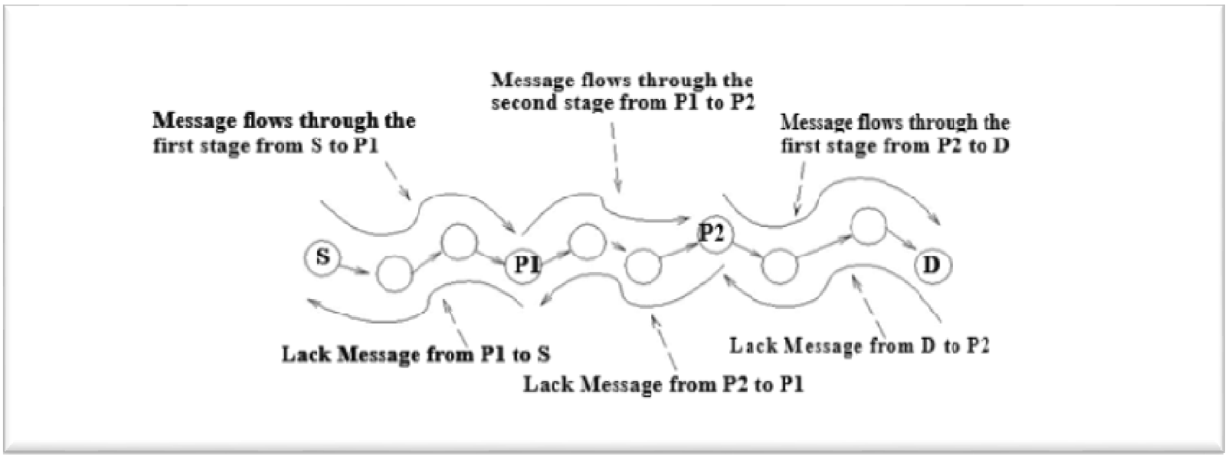

Figure 3. TCP connections are split into shorter localized segments

Simulation results indicate that an inter-proxy distance of between 3 and 5 has a good impact on both throughput and fairness. The authors report that an improvement up to 30% can be achieved in the total throughput by using Split TCP. Also, this proposal makes the role of proxy nodes more complex, as for each TCP session they have to control packet delivery to succeeding proxies.

Conclusion

We have presented the state-of-the-art of TCP over static and mobile ad hoc networks (SANETs and MANETs). The principal problem of TCP in this MANETs environment is clearly its inability to distinguish between losses induced by network congestion and other types of losses. [Aslanishvili, 2014] TCP assumes that losses are always due to network congestion. But while this assumption in most cases is valid in wired networks, it is not true in MANETs. In MANETs, there are indeed several types of losses, including losses cause by routing failures, by network partitions, and by high bit error rates. Performing congestion control in these cases, like TCP does, yields poor performance. In static multi-hop ad hoc networks the principal problem of TCP is the contention on wireless channel that induces routes failures and losses. In order to solve these problems, several proposals have been made in the literature. We classed these proposals as layered and cross layer proposals. In cross layer proposals, TCP and the underlying protocols cooperate to improve Ad hoc network performance. In layered proposals, one OSI layer is adapted. For example, TCP layer proposals require only the cooperation of the sender and receiver, like in TCP-DOOR and Fixed-RTO. However, cross layer proposals yield higher improvement than layered ones.

Bibliography

[Aslanishvili, 2012] Irma Aslanishvili, One model for two-hop relay Routing with limited Packet Lifetime the Conference for International Synergy in Energy, Environment, Tourism and contribution of Information Technology in Science, Economy, Society and Education era-7, 2012, ISSN 1791-1133, http://era.teipir.gr

[Aslanishvili, 2014] Irma Aslanishvili, Three RD Models for Two-Hop Relay Routing With Limited Packets Lifetime In Ad Hoc Networks", International Journal "Information Models and Analyses" Volume 3, Number 3, 2014

[Namicheishvili et al, 2011] Oleg Namicheishvili, Hamlet Meladze, Irma Aslanishvili, Transactions, "Two models for two-hop relay Routing with limited Packet Lifetime", Georgian Technical University, Automated Control Systems, 2011, No1(10), pp. 54-58.

Authors' Information

Irma Aslanishvili *– Iv.Javakhishvili Tbilisi State University, Faculty of Exact and Natural Sciences, teacher of Informatics and Computer Sensor Networks; e-mail: Irma.aslanishvili@tsu.ge*

Scientific Research: General theoretical information research, information systems and computer sensor networks

Tariel Khvedelidze*-Iv.Javakhishvili Tbilisi State University, Faculty of Exact and Natural Sciences, professor of Computer Sciences e-mail: tariel.khvedelidze@tsu.ge*

Scientific Research: General theoretical information research, information systems and computer Sciences