Transport layer issues in ad hoc wireless networks

Dmitrij Lagutin, dlagutin@cc.hut.fi

1. Introduction

Ad hoc wireless networks pose a big challenge for transport layer protocol and transport layer protocols designed for wired networks like TCP are not suitable for ad hoc wireless networks. There are many issues [1] listed below that a transport layer protocol needs to take into account.

Induced traffic: Ad hoc wireless networks use multi-hop radio relaying, and a link-level transmission affects neighbor nodes of both sender and receiver of the link. This induced traffic affects throughput of the transport layer protocol.
Induced throughput unfairness: Some MAC protocols, like IEEE 802.11 DCF, may add throughput unfairness to the transport layer. A transport layer protocol needs to take this into account to provide a fair throughput for contesting flows.
Separation of congestion control, reliability, and flow control: The throughput may be improved if the transport control protocol handles congestion control, reliability and flow control separately. Congestion is usually a local activity that affects only neighboring nodes while reliability and flow control are end-to-end issues. Separation of these should not produce significant control overhead.
Misinterpretation of congestion: Commonly used methods of detecting the congestion by measuring packet loss and retransmission timeout are not suitable for ad hoc wireless networks. Packet loss occurs in wireless networks relatively frequently for several reasons. Bit error rates are much higher than in wired networks and path breaks occur frequently because nodes are constantly moving and they may fail e.g. after draining a battery. Thus, a better method for detecting congestion must be used.
Completely decoupled transport layer: In wired networks, transport layer is usually almost completely decoupled from lower network layers. In wireless networks, cross-layer interaction would help transport layer protocol to adapt to the changes in the network
Power and bandwidth constraints: Ad hoc wireless networks are constrained by available power and bandwidth. These constraints affect the performance of transport layer protocol.
Dynamic topology: Topology of ad hoc wireless network may change rabidly and this leads to path breaks and partitioning of network. A transport layer protocol should be able to adapt to these changes.

2. Problems with the traditional TCP in ad hoc wireless networks

TCP [2] is reliable, end-to-end, connection oriented transport layer protocol . TCP is responsible for congestion control, flow control, in-order delivery of packets, and reliable transportation of packets

TCP handles the congestion control in the following way. TCP regulates the number of packets sent to the network by changing the size of the congestion window. Initially, at the beginning of the TCP session, the size of congestion window is set to one maximum segment size (MSS). If the acknowledgment (ACK) is received during the retransmission timeout period (RTO), size of the congestion window is doubled until the size reaches slow-start threshold. After the slow-start threshold is reached, congestion window increases linearly, by one MSS for every received ACK. If the ACK is not received in time, TCP assumes that the packet is lost and invokes congestion control mechanism: slow-start threshold is halved and the size of congestion window is decreased to one MSS.

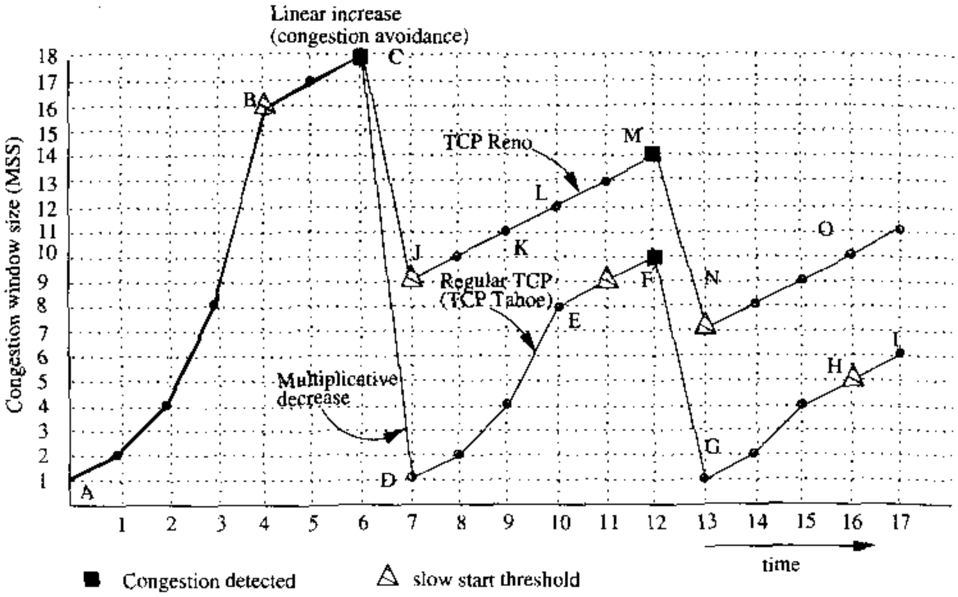

Figure 1: An example of TCP congestion window handling (C. Murthy and B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols)

Figure 1 above shows an example of congestion handling in a traditional TCP and TCP Reno [3]. In the example, the size of the congestion window is increased exponentially until the slow-start threshold (16 in this case) is reached, after the threshold the size of congestion window increases linearly. After packet loss is detected at C, size of the congestion window is decreased to one and the slow-start threshold is halved to 8. Second packet loss situation (F) halves the slow-start threshold further to 4. This means that the size of congestion window will increase at much slower pace in the future. TCP Reno works in a bit different way, upon detecting the packet loss, it decreases the congestion window size to half of the current value and increases congestion window linearly. Thus, the TCP reno usually has bigger congestion window in packet loss situations compared to the traditional TCP.

The congestion handling is the biggest single issue that makes the traditional TCP a poor choice for ad hoc wireless networks. TCP has been designed for wired networks that have very low bit error rates, thus when the packet loss is encountered, TCP assumes that there is significant problem in the network and initiates aggressive congestion control. In ad hoc wireless networks packet loss can frequently occur for many reasons. First, the bit error rates are generally much higher in wireless network. Second, the nodes are constantly moving and path breaks can occur frequently. If the packet loss occurs frequently, the size of congestion window in TCP will stay at very low level most of the time and this naturally decreases the throughput of the network significantly. In addition, after a route reconfiguration, new route may accept higher throughput. However, TCP does not take this into account. The problem worsens when the path length increases, since the increased path length increases the probability that a path break occurs somewhere along the path. With a path length of just 4 hops and link break probability of 10% for each link, throughput of TCP decreases to about 20% of the original.
There are also other issues in using TCP with ad hoc wireless networks listed below.

Uni-directional path: TCP relies on ACK packets for end-to-end reliability. ACKs consume very little of bandwidth in wired networks because they are very short in comparison with data packets. However, in IEEE 802.11 wireless network, every ACK requires RTS-CTS-Data-ACK exchange which can lead to 70 bytes overhead. The forward path and the reverse path between two nodes may be different in ad hoc wireless networks, and these ACK packets may affect the throughput of reverse path. A path break in a different reverse path will significantly affect the performance of the TCP because in that case it can not receive any ACK packets.
Multipath routing: Multipath routing has several advantages in ad hoc wireless networks like reduction of route computation time, better resilience to path breaks, high call acceptance ratio, and

better security. However, multipath routing can lead to a high amount of out-of-order packets which degrade performance with TCP because of invocation of congestion control.
The use of sliding-window-based transmission: TCP uses a sliding-window-based transmission, after ACKs arrive from the destination, TCP transmits further packets. This method avoids the use of individual fine-grained timers for each TCP flow. This can lead to degradation of performance if the MAC layer protocol can not provide fairness. For example, with MAC protocols such as CSMA/CA protocol , a node that has captured the channel has a higher probability of capturing the channel again. This can lead to a high number of ACK packets being delivered to the TCP sender at once which leads to a burstiness in the traffic.

3. TCP based solutions for ad hoc wireless networks

TCP based solutions for ad hoc wireless networks can be classified into two groups: protocols based on the split approach and protocols based on the end-to-end approach. Protocols based on end-to-end approach include Feedback-based TCP [4], TCP with explicit link failure notification [5], TCP-BuS [6] and Ad hoc TCP [7], while Split-TCP [8] is the protocol that is based on the split approach. In this paper Split-TCP and Ad hoc TCP protocols will be covered in more detail.

3.1 Ad hoc TCP

Ad hoc TCP (ATCP) relies on a network layer feedback to make the TCP sender aware of the status of the network path. ATCP takes advantage of explicit congestion notification (ECN) flags and ICMP destination unreachable (DUR) messages to detect network congestion and path breaks. ATCP is not a full replacement to the TCP, instead it operates between the TCP and the network layer. Thus, ATCP is fully compatible with the traditional TCP and the ATCP support is only required for the sender.

When packet loss is detected or packets arrive out-of-order to the destination, ATCP simply retransmits missing packets without invoking congestion control mechanism. This provides a performance advantage against traditional TCP that invokes congestion control every time the packet loss or out-of-order packets are detected.
When the ATCP sender receives ECN message, it moves to the congested state where it lets TCP invoke congestion control normally.
When DUR packets are received, ATCP moves in to disconnect state where it ceases to send packets. After the connection is re-established, ATCP sets the size of the congestion window to one in order to make TCP to determine optimal congestion window size for a new connection.

The major advantage of ATCP is that it is fully compatible with existing TCP while at the same time it offers improved throughput for ad hoc wireless networks by avoiding unnecessarily using congestion control mechanism. Compatibility with TCP means that ATCP can work seamlessly with the Internet.
The disadvantages of ATCP include dependency on the network layer protocol to detect route changes and partitions. This functionality may not be implemented by all routing protocol, thus ATCP can not effectively work with all routing protocols. In addition, adding thin ATCP layer to the TCP/IP protocol stack requires changes to the interface functions.

3.2 Split-TCP

As was mentioned previously, one significant problem with the traditional TCP is rabid degradation of throughput with increased path length. This can also lead to unfairness between TCP sessions, since sessions with a short path length will achieve much higher throughput compared to sessions with a long path length.

Split-TCP provides a solution to this problem by separating transport layer objectives into congestion control and end-to-end reliability. The idea is that the congestion control is mostly a local phenomenon and thus it requires local solution. The reliability is a end-to-end requirement and thus it requires end-to-end acknowledgments.

Split-TCP also splits TCP connection into a set of shorter concatenated TCP connections, intermediate nodes, also called as proxy nodes, are end point of these short connections. The number of proxies is determined by the path length, longer paths will have a higher amount of proxies to keep the length of individual connection short.

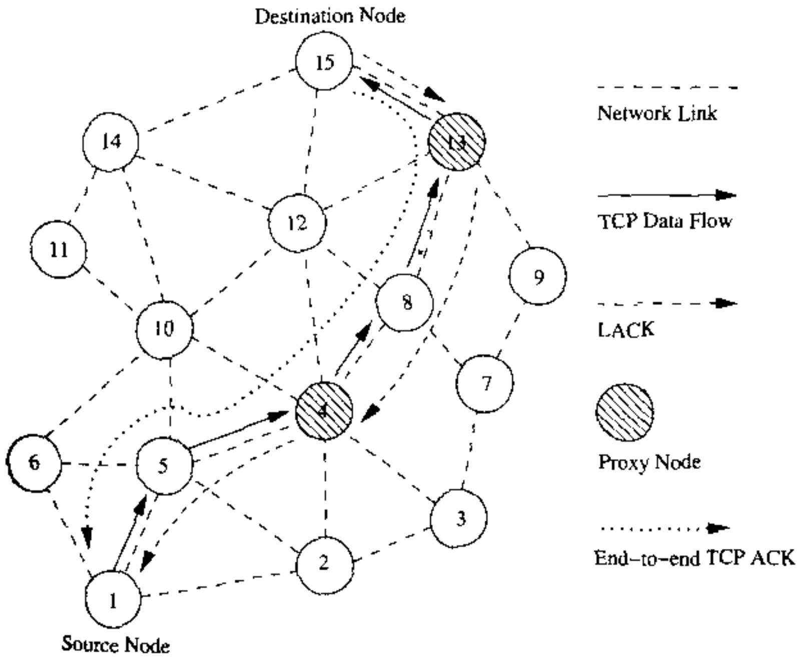

Figure 2: An overview of split-TCP (C. Murthy and B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols)

The Figure 2 above provides an overview of the split-TCP. In this example, nodes 1 and 15 are communicating with each other and the route goes through nodes 5, 4, 8 and 13. Nodes 4 and 13 are acting as proxy nodes for split-TCP. When the proxy receives packets, it stores them into a buffer and send a local acknowledgment (LACK) to the previous proxy or the source. When the proxy receives LACK from the next proxy or the destination, it clears buffered packets. Since LACKs do not guarantee end-to-end delivery, the destination node will also send end-to-end ACK to the source upon receiving packets. The source node clears its buffered packets only after receiving end-to-end ACK. The transmission control window is split in two at the source node: the congestion window which size is determined by LACKs from the nearest proxy and the end-to-end window which size is determined by end-to-end ACKs. The size of the congestion window naturally stays within the size of end-to-end window.

In split-TCP, intermediate nodes along the path determine themselves wherever to act as a proxy or be just a simple forwarding node. A simple algorithm could make this decision based on the number of hops from the last proxy or the source node, if the number of hops exceeds some predefined value, the node will act as a proxy.

The advantages of split-TCP include improved throughput, improved fairness and lessened impact of mobility. All these advantages originate from the shorter path lengths. For example, if the destination node moves in the network, only the last proxy needs to adapt to the change, the source node can still send packets normally to the first proxy.

Split-TCP has also disadvantages, first, it requires modification to be made to the TCP protocol. It also violates traditional end-to-end connection handling which may cause problems with security schemes that use IP payload encryption. Finally, failure of proxy nodes could significantly decrease throughput of the network.

4. Other transport layer solutions for ad hoc wireless networks

There exist also transport layer protocols for ad hoc wireless networks that are not based on TCP, like Application controlled transport protocol [9] and Ad hoc transport protocol [10]. In this paper, Ad hoc transport protocol will be covered in more detail.

4.1 Ad hoc transport protocol

Ad hoc transport protocol (ATP) is a protocol designed for ad hoc wireless networks, it is not based on TCP. ATP differs from TCP in many ways: ATP uses coordination between different layers, ATP uses rate based transmissions and assisted congestion control and finally, congestion control and reliability are decoupled in ATP. Like many TCP variants, ATP also uses information from lower layers for many purposes like estimating of initial transmission rate, congestion detection, avoidance and control, and detection of path breaks. ATP obtains network congestion information from intermediate nodes, while the flow control and reliability information are obtained from the ATP receiver.

The ATP uses a timer-based transmission where the rate is dependent on the congestion in the network. As packets travel through the network, intermediate nodes attach the congestion information to each ATP packet. This congestion information is expressed in terms of weighted average of queuing delay and contention delay experienced by the packets at intermediate node. The ATP receiver collates this information and sends it back to the sender in the next ACK packets, and the ATP sender can adjust its transmission rate based on this information.

During the establishment of the connection, the ATP sender determines the initial transmission rate by sending probe packet to the receiver. Each intermediate node attaches network congestion information to the probe packet and the ATP receiver replies to the sender with an ACK packet containing relevant congestion information. In order to minimize control overhead, ATP uses connection request and ACK packets as probe packets.

ATP increases the transmission rate only if the new transmission rate (R) received from the network is beyond a threshold (x) greater than a current rate (S), *e.g.* if $R>S(1+x)$ then the rate is increased. The transmission rate is increased only by a fraction (k) of the difference between two rates, *i.e.*: $S=S+(R-S)/k$, this kind of method avoids rapid fluctuations in the transmission rate.

If the ATP sender has not receive ACK packets for two consecutive feedback periods, it significantly decreases the transmission rate. After a third such period, connection is assumed to be lost and the ATP sender moves to the connection initiation phase where it periodically generates probe packets. When a path break occurs, the network layer sends an explicit link failure notification (ELFN) packet to the ATP sender and the sender moves to the connection initiation phase.

The major advantage of ATP is the avoidance of congestion window fluctuations and the separation of the congestion control and reliability. This leads to a higher performance in ad hoc wireless networks.

The biggest disadvantage of ATP is incompatibility with a traditional TCP, nodes using ATP cannot communicate directly with the Internet. In addition, fine-grained per-flow timer used at the ATP sender may become a bottleneck in large ad hoc wireless networks.

5. Conclusions

Designing a good transport layer protocol for ad hoc wireless networks is a very challenging task. The problem basically lies in performance and compatibility trade off. A transport layer protocol must modify traditional TCP in order to solve problems that make traditional TCP unsuitable for ad hoc wireless networks. In addition, the transport layer protocol must utilize information from lower layers in order to accurately detect network congestion and path breaks. However, if the modifications to the TCP are too drastic, a new transport protocol will not be compatible with TCP. In addition, not all routing protocols can provide a lot of information about network conditions to upper layers, thus transport layer protocols using a lot of lower layer information will not be compatible with all existing routing protocols. Thus, a transport layer protocol can either achieve a very good performance or a very good compatibility but it can not achieve both at the same time.

The amount of potential transport layer protocols is quite high, and they all have own advantages and disadvantages. Different transport layer protocols are suitable for different applications. For example, if the ad hoc network is a fairly closed and consists of nodes from a single organization that are communicating only with themselves, compatibility with TCP may not be required, and thus a protocol like ATP may be a good choice. On the other hand, if the ad hoc network is open with many different kind of nodes using the network, or if the nodes in the network need to communicate with the Internet, TCP compatibility is essential and thus TCP based transport layer protocols should be considered.

References

[1] C. Murthy and B. S. Manoj. Ad hoc Wireless Networks: Architectures and Protocols. 2004.

[2] J. Postel. Transmission Control Protocol. IETF RFC 793, September 1981.

[3] W. R. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmission and Fast Recovery Algorithms. IETF RFC 2001, January 1997.

[4] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash. A Feedback Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks. IEEE Personal Communications Magazine, vol. 8, no. 1, pp. 34-39, February 2001.

[5] G. Holland and N. Vaidya. Analysis of TCP Performance over Mobile Ad Hoc Networks. In Proceedings of ACM MOBICOM 1999, pp. 219-230, August 1999.

[6] D. Kim, C. K. Toh, and Y. Choi. TCP-BuS: Improving TCP Performance in Wireless Ad Hoc Networks. Journal of Communications and Networks, vol 3., no. 2, pp. 1-12, June 2001.

[7] J. Liu and S. Singh. ATCP: TCP for Mobile Ad Hoc Networks. IEEE Journal on Selected Areas in Communications, vol. 19, no. 7, pp. 1300-1315, July 2001.

[8] S. Kopparty, S. V. Krishnamurthy, M. Faloutsos, and S. K. Tripathi. Split TCP for Mobile Ad Hoc Networks. In Proceedings of IEEE GLOBECOM, vol. 1, pp. 138-142, November 2002.

[9] J. Liu and S. Singh. ATP: Application Controlled Transport Protocol for Mobile Ad Hoc Networks. In Proceedings of IEEE WCMC 1999, vol. 3, pp. 1318-1322, September 1999.

[10] K. Sundaresan, V. Anantharaman, H. Y. Hsieh, and R. Sivakumar. ATP: A Reliable Transport Protocol for Ad Hoc Networks. In Proceedings of ACM MOBIHOC 2003, pp. 64-75, June 2003.