Performance Evaluation and Optimization of SCTP in Wireless Ad-Hoc Networks

Antonios Argyriou and Vijay Madisetti
School of Electrical & Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
{*anargyr, vkm*}@*ece.gatech.edu*

Abstract

In this paper we evaluate the performance of the Stream Control Transmission Protocol (SCTP) in a wireless ad-hoc network environment under two routing protocols: DSR and AODV. Furthermore, we propose a cross-layer optimization to the SCTP and DSR protocols, that handles pro-actively route failures. The novelty in the proposed mechanism consists in the identification of route failures which is done at the transport layer, allowing thus faster recovery. Our simulation results prove the efficiency of our approach for a variety of mobility scenarios.

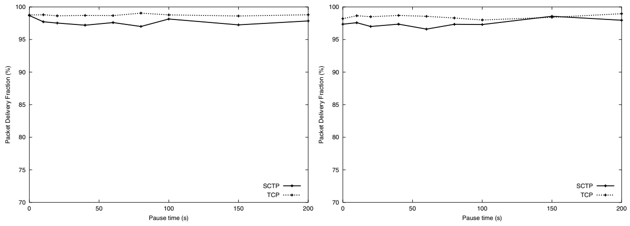
1. Introduction

The evaluation of SCTP [1] and at the same time its comparison with TCP over DSR [2] and AODV in mobile ad-hoc networks is presented in this paper. In addition we propose a number of modifications to the SCTP/DSR protocols and we show that by maintaining multiple active paths at the transport layer, a mobile node can recover faster in the case of route failures. The use of a lightweight probing mechanism for monitoring path status ensures the minimum necessary signalling for this job.

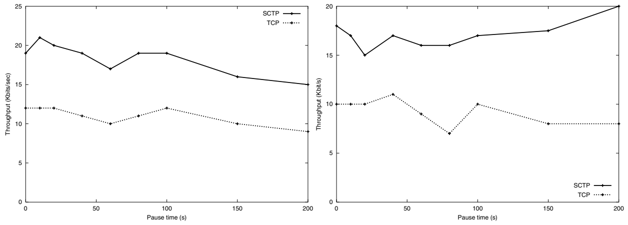
2. SCTP performance evaluation

The ns-2 network simulator was used for all our experiments. We used 802.11 as the MAC level protocol. We defined a $600m \times 600m$ area for node movements, node speed of 20 m/s, and a simulation time of 200 seconds. The random way-point model was used for creating all the node movement scenarios. Backlogged FTP sources were used, creating randomly 20 source-destination pairs out of a total of 50 nodes in a scenario.

Fig. 1(a) shows ratio of the application data packets each protocol was able to deliver when DSR was used, as a function of node mobility rate (pause time). TCP delivers the largest part of its packets while SCTP suffers from 1-2% reduced packet delivery fraction (PDF). Even though SCTP delivers a slightly smaller percentage of its packets, it is able to deliver a higher number of them (Fig. 1(c)). This happens because SCTP uses the SACK algorithm resulting into less RTO timeouts. In the case of AODV, the average throughput and PDF, follow the results of DSR, but the absolute values are lower (Fig. 1(d)). For the same scenario, the number of link breakages that resulted in timeouts for SCTP was nearly 50% less when compared to TCP. Only long duration route breakages, that resulted in loosing more than half window of packets, had to be resolved through an RTO expiration and re-transmission. Finally, table 2 shows results for throughput and PDF for various mobile host speeds.



(a) PDF with DSR (b) PDF with AODV



(c) Throughput with DSR (d) Throughput with AODV

Figure 1. Results for SCTP/TCP over DSR/AODV.

Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)

	DSR		AODV	
Speed	TCP	SCTP	TCP	SCTP
10m/s	12.1	20	10	16
20m/s	12	19	10	18
30m/s	13	19	11.3	14.1

(a) Throughput in Kbps

	DSR		AODV	
Speed	TCP	SCTP	TCP	SCTP
10m/s	98.73	98.7	98.84	97.3
20m/s	98.77	98.68	98.2	97.35
30m/s	98.56	98.67	99.06	98.11

(b) PDF (%)

Figure 2. Throughput and PDF for various host speeds.

3. Protocol modifications

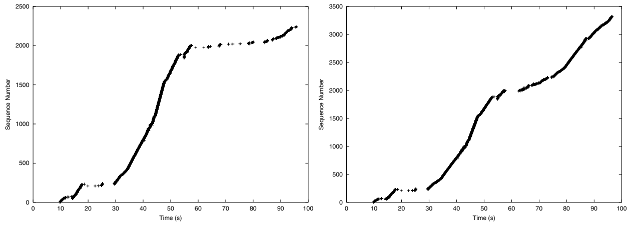
We modified the SCTP heart-beating and retransmission algorithms.

Heartbeating. Each endpoint of an SCTP association sends Heartbeat messages (every RTO) that monitor the status of all the remote addresses/paths of a multi-homed host [1]. In the modified protocol, we ensure that paths given to SCTP by DSR are disjoint. This allows Heartbeat messages to be sent explicitly to different paths. The modified SCTP assigns the best path to the main data flow, while, only Heartbeat messages are sent to the secondary paths. Now, after two Heartbeat retransmissions fail, we mark a specific path inactive. We followed this approach, because we observed that when mobility is increased, two failed Heartbeat re-transmissions indicate route breakage and not packet losses. In addition, in this way, we avoid route discovery delay which can be a very lengthy process in the case of DSR. Current approaches that freeze the state of the sender are problematic, since the saved state does not correspond to the new path but to the old one. So, after a route failure has been resolved, the state parameters will be invalid. However, with SCTP we maintain congestion control parameters for every path, allowing thus the maintenance of a distinct state.

Overcoming packet reordering. Out-of-order delivery of packets happens when a packet flow switches between paths that are asymmetric. Side-effects of this phenomenon include spurious retransmissions and reduction of $cwnd$. The modified protocols handle this as follows: DSR notifies SCTP when the primary route fails, so that SCTP switches to a new path. In addition SCTP will also do that, as we earlier said, after the $2RTO$ Heartbeat timer expires on the primary path. By doing so, we are sure that the time of sender idleness is minimum. Furthermore, the SCTP sender maintains two variables, for each local address, that keep the lowest and highest TSN [1] sent during the last $cwnd$ round to the receiver. When the receiver replies with a SACK containing a Gap report for TSNs that do not belong to this range, the sender does not increase the Gap Ack reports and process the SACK chunk as a normal SACK that only acknowledges any outstanding data. In this way packets still in transit in a slow link will not cause spurious retransmissions.

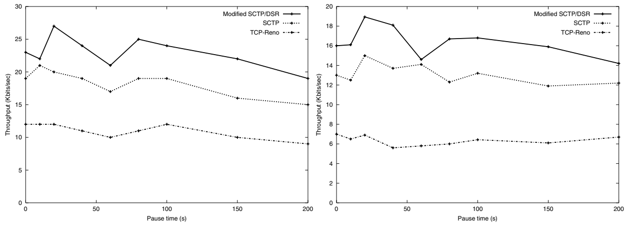
4. Results for the modified SCTP

In Fig. 3(b) we can see that the modified SCTP protocol does not suffer from the timeout of Fig. 3(a). The reason is that at time 60 sec, after time $2RTO$, the modified SCTP sender switches to an existing alternative path. This also happened for other flows in the same scenario. The average throughput for the modified protocols is shown in Fig. 3(c). The proactive nature of the modified protocols, is capable of taking advantage the alternative paths, and switch to them as soon as possible. Fig. 3(d) shows results for a scenario with the same number of hosts, but for a larger area. As expected, performance of both vanilla and modified protocols, is significantly reduced since the number of alternative paths is decreased.



(a) Sequence no. for SCTP

(b) Sequence no. for modified SCTP



(c) Tput. ($600m \times 600m$)

(d) Tput. ($800m \times 800m$)

Figure 3. A-SCTP/DSR, SCTP and TCP throughput with DSR for varying pause time.

5. Conclusions

In this paper we proposed a set of modifications to the SCTP protocol for handling pro-actively route failures in mobile ad-hoc networks. We showed that the transport layer allows for faster path selection, in case a number of paths exist, leading thus to improved overall throughput.

References

[1] R. Stewart et al. Stream Control Transmission Protocol. RFC 2960, October 2000.

[2] David B. Johnson et al. The DSR protocol for Mobile Ad Hoc Networks. draft-ietf-manet-dsr-09.txt, April 2003.