TCP Optimization for Mobile Ad Hoc Networks

Despina Triantafyllidou and Khaldoun Al Agha
HIPERCOM@LRI – University of Paris XI
Orsay Cedex, 91405, France
{dtriant, alagha}@lri.fr

***Abstract*—TCP is the standard transport protocol for reliable communications, over the wireless channel. This makes necessary to overcome its performance defficiencies, in the ad hoc environment. In this context, the attempt is twofold. First, to extensively study its behavior in mobile ad hoc networks (MANETs). Second, to optimize its performance, preserving its end-to-end operation and semantics.**

We present a novel simulation model for TCP, based on the key idea of representing each network node by a subqueue. Compared to the existing models, ours is fully scalable and optimized. Then, we apply an efficient adaptation for the TCP retransmission (RTO) timer, to avoid the spurious timeouts in MANETs. The main contribution is the utilization of a link state protocol to convey current delay information for the whole end-to-end path, to TCP, which in turn refreshes its timer accordingly. This approach outperforms regular TCP, in high density networks. Last, we present an alternative route discovery algorithm in MANETs, based on the lowest delay path. Under certain scenarios, the proposed routing proves beneficiary for the throughput, while in other cases the latter decreases.

I. A Scalable Optimized Simulation Model for TCP

We argue that it is not easy to study the performance of TCP within short time limits. The TCP mechanisms are event driven, rather than time scheduled. Thus, any network event entails the initialization of the appropriate recovery mechanisms. Leaving TCP reach its steady state is time demanding, but essential, in order to study its performance.

To carry out this evaluation, we adopt the idea of multiple subqueues for representing up to 100000 network nodes, which are placed in a region by randomly selecting their coordinates. The model supports an arbitrarily high number of TCP connections. Such an approach is highly extensible, considering that, in wired networks, we have to explicitly interconnect the stations. With this subqueue representation, the number of physical interconnections is minimized.

We can reach simulation times of over 1 week. Our experiments reveal that TCP reaches its steady state much farther in time, than the usual simulations present. So, our model allows to study the real performance of the system, in its stationary state.

Another interesting aspect of this approximation is the significantly lower portion of resources needed during the simulation. We no longer have packets being transferred between nodes. The latter is abstracted to two queuing operations: one dequeue from the sending "node", and one enqueue from the receiving "node". This abstraction leads to much lower execution times for the simulated scenarios.

II. Link-State Based TCP RTO Timer Adaptation (LSTRA)

The RTO timer of TCP is an upper limit prediction of the round-trip time (RTT). We propose a link-state based adaptation for this timer, where, accurate hop-by-hop delay information for all internediate node pairs is gathered, and communicated to the end systems. We exploit the OLSR control packets to estimate and convey this information to all network nodes.

The OLSR HELLO message exchange is used for the estimation of the local RTT between any two neighbors. During the TC flooding, the TC messages that are forwarded through the MPR nodes, aggregate these local RTT estimations, and form the delay estimation for the whole end-to-end path. For this purpose, the HELLO and TC control packets carry additional timestamps.

The delay value is then communicated to TCP. This can be achieved by means of a cross-layer scheme. Since the TC messages are handled by the application layer, this has to set the appropriate value to the TCP parameter used for the adaptation of the timer. How often TCP gets this feedback, depends on the TC interval.

This approach does not require any synchronization among the stations. Moreover, it accounts for the network loss ratio. Due to the way we account for the local RTT, we also consider the local buffer size, i.e., the network congestion. The simulation results confirm this intuition.

III. Average Path-Delay (APD) routing

The key idea now is to apply the link-state routing to affect the TCP throughput. The standard heuristic of OLSR determines the MPRs, so as to mitigate the overhead produced by the flooding operation of the link-state algorithm. Such an approach, provides the shortest path in terms of number of hops, but does not advertise the links with good quality characteristics, such as high bandwidth and low delay. These links might remain idle, throughout the TCP flow lifetime.

We again exploit the OLSR HELLO message periodic exchange, to estimate the hop delay between two nodes; we also account for the queuing delay. To provide some stability, an average hop-delay is determined by giving a certain weight to the most recent delay measurement, taking also into account the previous average values. This value is conveyed to the rest of the network with the TC forwarding procedure. The goal is to choose a certain route that minimizes the sum of the average delay values. The route discovery algorithm will select

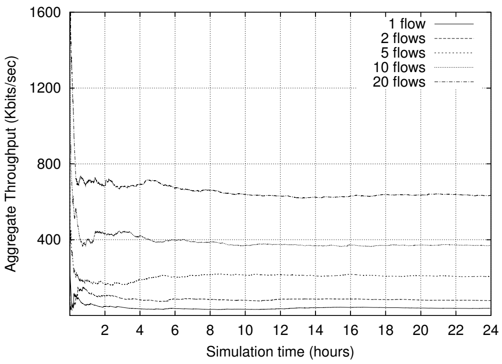

Fig. 1. Random scenario, max node speed 2 m/sec, throughput stabilizes after hours.

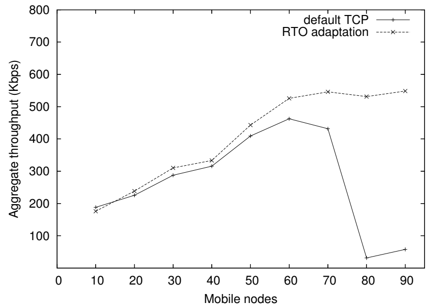

Fig. 2. Throughput comparison of LSTRA with default TCP, max node speed 2m/sec.

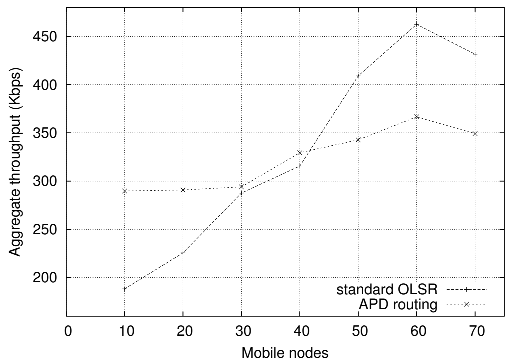

Fig. 3. Throughput comparison of standard OLSR with APD routing, max node speed 2 m/sec.

the links that minimize the aggregate delay, for the whole end-to-end path. The shortest hop constraint is not of our interest; the calculated path can be longer, in terms of number of hops.

IV. Performance Evaluation

The scenarios involve randomly moving stations, in a 1000m*1000m area, which run 10 FTP applications, unless otherwise stated.

A. Defining the Point of Stability

Fig. 1 depicts the aggregate throughput of five random topologies, with 50 nodes, and a different number of concurrent FTP connections, within 24 hours of simulation time. A closer look shows that the system reaches stability at least after several hours of simulation time.

B. Performance of LSTRA scheme

In Fig. 2, LSTRA does not offer any improvement in sparse networks. For higher density networks, the improvement becomes significant. Actually, LSTRA keeps the throughput from radically falling. This can be explained as follows. In dense MANETs, the overhead offered by the OLSR control packets is larger. These segments occupy the local buffers, and contend for channel capture, as if they were data packets. The TCP connections cannot defend against the control packets, because these are transmitted periodically, rather than based on a flow control scheme, like the TCP's congestion avoidance. So, they are defeated, in terms that when they experience timeouts they have to proceed with go-back-N retransmission and reduce their rate. The proposed adaptation makes the TCP connections more robust to network load.

C. Performance of APD routing

Fig. 3 depicts the achieved throughput of 10 FTP connections with the APD routing, compared to standard OLSR. The APD scheme performs better in a sparse network, but standard OLSR prevails on the APD routing, for dense networks. Interestingly, our approach reveals the following contradiction. In dense networks, a decrease on the path's average delay might be achieved at the expense of a longer path, in terms of number of hops, which increases the collision probability, and results in reduced throughput. So, the low delay route is not always the best choice. This can be verified by the increased collision percentage measured at the MAC layer.

V. Conclusion and Future Work

By implementing the scalable simulation model for TCP, we have a powerfull and reliable tool to investigate the long term performance of realistic MANET scenarios. Therefore, we locate the interactions among TCP and the involved protocols, and try to make it work seamlessly. The work descriebed above can be found in detail in [1]–[3].

Due to the contradiction presented by the APD routing, we intend to also consider the channel loss probability, when deciding the optimal path. Another challenge is to study the effect of the local buffers availability to the maximum allowed window size of TCP. In that case, we might minimize the negative effect of retransmitting a large window during the go-back-N procedure.

References

[1] D. Triantafyllidou and K. AlAgha, "Evaluation of TCP performance in MANETs using an optimized scalable simulation model," *IEEE MASCOTS*, Istanbul, Turkey, October 2007.

[2] D. Triantafyllidou and K. A. Agha, "Efficient adaptation of TCP's RTO timer to avoid spurious timeouts in MANETs," in *10th International Symposium on Wireless Personal and Multimedia Communications (WPMC)*, Jaipur, India, December 2007.

[3] D. Triantafyllidou and K. A. Agha, "A link-state approach for improving TCP in MANETs using average path-delay routing," in *Submitted to IEEE Global Communications Conference (IEEE GLOBECOM)*, 2008.