Chapter 7

Conclusions and Future Work

In this chapter we conclude this dissertation by summarizing our contributions and discussing directions for future work.

7.1 Summary

In this dissertation, we have focused on two problems relevant to Internet data networking over next-generation broadband satellite systems that provide "last-mile" network access: i) improving the performance of reliable transport protocols over high-latency paths, and ii) routing strategies for constellations of low-earth-orbiting satellites. In this section, we briefly review our main conclusions.

7.1.1 Transport Protocols for Broadband GEO Systems

Recall that we focused on the performance problems encountered when using TCP over GEO satellite connections. Our data supports the following conclusions:

- **TCP fairness** As described earlier, the problem of TCP fairness in a hetergeneous environment is a long-standing research problem. While we were able to reproduce the simulation results reported by Floyd [39] that showed how TCP connections, if they were to all use a "Constant-Rate" linear increase rate during congestion avoidance, could achieve better global network fairness, our simulations suggest that it would be difficult to deploy this algorithm incrementally in the network, because "Constant-Rate" connections are less aggressive in obtaining a share of congested network bandwidth. We presented some simulation results indicating that long-delay TCP connections may be able to unilaterally improve the fairness of the network by becoming slightly more aggressive during congestion avoidance. The interesting aspect of this result is not that more aggressive connections could do better but that, in our simulation topologies, such connections *always* improved the network fairness metric without compromising utilization of the bottleneck link. Based solely on our simulation results, we cannot recommend this strategy or a particular policy for deployment until it has been examined further (particularly through network experiments), but the potential seems promising. We note that the best solution to this problem appears to be the inclusion of mechanisms within routers that guarantee fair sharing of bandwidth on a per-flow basis, but it is unclear at this time whether there are sufficient incentives to deploy such mechanisms in the Internet.

- **Satellite optimized TCP implementations and split connections** We conducted a study of satellite TCP performance, focusing on end-to-end performance in which part of the satellite TCP connection traverses the Internet, and investigating both file transfer performance and short Web-like connection performance. We demonstrated how SACK loss recovery and NewReno congestion avoidance principles should be used in conjunction to achieve good TCP file transfer performance over GEO satellite links, and we explained why other types of TCP implementations often perform much worse. Nevertheless, we showed how even moderate levels of congestion in the wide-area Internet can derail the file transfer performance of even satellite-optimized TCP connections. We also quantified the performance gains that TCP enhancements such as TCP for Transactions and an increased initial window can have on Web-like traffic, showing that user-perceived latency can be reduced by a factor of two to three through such protocols. Because congestion can have such a negative impact on end-to-end satellite TCP file transfers, and because enhancements like TCP for Transactions may not be deployed in the Internet due to security concerns, we concluded that the safest bet for ensuring high performance TCP connections over GEO satellite links appears to be a split-connection approach. We demonstrated how the end-to-end performance degradations could be nearly eliminated through a split connection approach, provided that IP security protocols do not constrain deployment of such gateways.

- **Satellite-optimized transport protocol** We described the overall design and performance of a satellite-optimized transport protocol (STP) that is specifically designed for a broadband satellite network characterized by high degrees of bandwidth asymmetry. We then experimented with simulation models and a BSD/OS kernel implementation of the protocol. We compared this protocol to TCP and found that it provides very good performance in a high loss environment (even with bit error ratios as low as 10^{-5}), less sensitivity to large variations in the round trip delay experienced by packets, delay performance approaching that of TCP for Transactions for short transfers, and a reduction of up to an order of magnitude in the amount of bandwidth used on the reverse channel to return acknowledgments.

7.1.2 Unicast Packet Routing for LEO Constellations

We also studied the unicast packet routing problem for LEO constellations and concluded the following:

- **LEO extensions for the *ns* simulator** We described the design and construction of a packet-level simulator for LEO networks. This network simulator is valuable because it is integrated with the *ns* simulator already in wide use by the research community. This simulator revealed some interesting fundamental delay performance properties of LEO networks that have not yet been seen in the literature. In particular, we illustrated typical delay performance that might be seen by users of the Iridium and (proposed) Teledesic systems, and showed the impact of using hop counts, rather than delay metrics, as the cost metric used in shortest path routing algorithms.

- **Distributed LEO routing using geographic-based addresses** We explored the hypothesis, advanced by several researchers, that by making locally optimal packet forwarding decisions that minimize the geographic distance to the destination, one can obtain routes that are close to optimal in terms of delay performance. We constructed a distributed routing protocol based on this hypothesis, and found that while the LEO network mesh was sufficiently dense and regular to admit good routes based on this approach (routes that were, on average, no more than 5 to 10 ms– less than 10%– worse than globally optimal routes), there are a number of problems with commercially-proposed LEO network topologies that make construction of a robust protocol difficult. In particular, the distortions in the topology in the polar regions and at the counter-rotating orbital planes require significant additions to a distributed routing protocol based on geographic addresses.

- **Centralized LEO routing using geographic-based addresses** We examined the use of geographic addressing and cell geometries for use with a centralized routing system. We developed an optimal cell numbering scheme for rectilinear grids on the Earth's surface and proved its optimality. We then compared two approaches for reducing the amount of traffic needed to support forwarding table changes that must be periodically uploaded to satellites. For the first approach, exploiting temporal consistency in routing tables from state to state, our numerical results for a simulated Teledesic-like system indicated that, generally, 70 to 95 percent of the routing table entries persist between states. We then examined the benefit of aggregating contiguous forwarding table entries into a smaller number of entries. We developed an greedy algorithm for cell aggregation in one dimension that is optimal in aggregating contiguous cells, and demonstrated that it could be used as an effective approximation algorithm for cell aggregation in two dimensions, since the problem of address aggregation in two dimensions is NP-complete. Our numerical results suggest that, with this algorithm, the size of satellite-borne forwarding tables devoted to non-local destinations can be reduced by an order of magnitude, from tens of thousands of entries to a few thousand entries.

7.2 Software Availability

We have made available online, in source code form, most of the protocol and software implementations described herein:

- Our satellite-optimized TCP implementation (described in Appendix A) for BSD/OS 3.0 is available at `ftp://daedalus.cs.berkeley.edu/pub/tcpsack/`

- Simulator extensions for generating background HTTP traffic in the *ns* simulator are available at `ftp://daedalus.cs.berkeley.edu/pub/ns/httptrafficgen.tar`

- Source code for BSD/OS 3.0 and FreeBSD kernel implementations of the Satellite Transport Protocol (STP) are available at `ftp://daedalus.cs.berkeley.edu/pub/stp/`

- Our LEO satellite extensions to the *ns* simulator have been incorporated into the main distribution and documented. *ns* is available at `http://www-mash.cs.berkeley.edu/ns/`

7.3 Future Directions

The results of this dissertation point to several interesting directions for future work:

- **TCP fairness** Our results above indicated that it may be worthwhile to selectively increase the aggressiveness of certain TCP connections to improve network fairness. However, our results are preliminary and need to be experimentally validated before deployment can be recommended. For example, a wider range of topologies should be considered, and experiments as well as more simulations are needed to decide on the best policy. The impact of this algorithm on connections used for short data transactions (such as many small Web transfers) should be studied more. Finally, mechanisms for more accurately determining a connection's RTT, as well as policies that might be invoked as a function of the RTT observed, require more study. It should be emphasized that mechanisms for per-flow fair sharing of congested links would obviate the need for improvements to TCP's end-to-end algorithm, so further work on the design and deployment of these mechanisms would also be very useful. Also, Mo has demonstrated the potential existence of a fair, distributed flow control algorithm but has not been able to construct such an algorithm [89]; if such an algorithm were discovered, it may be a substantial improvement over the current one.

- **Satellite-optimized TCP** We have identified the performance of the slow start and congestion avoidance algorithms, as well as implementation details such as correct sizing of socket buffers and use of the correct TCP options, as the biggest hurdles to overcome to improve TCP performance over satellite links. Further work on automating the correct configuration of TCP, such as described in [122], could help the deployment of more satellite-friendly implementations.

- **Split connections** The main obstacle to the deployment of split-connection protocol gateways is their interaction with a security infrastructure. In particular, any IP security protocols that encrypt the payload of an IP packet render a split connection gateway useless. Work on how to integrate performance enhancing proxies into the trust infrastructure of a secure network would be valuable. Another issue that could be pursued further is how data flows between split connections should interact. Specifically, consider the case of using a split connection to aid in Web browsing. In this case, packets may trickle in at a slow rate from the terrestrial (server) side of a connection. If the split connection is implemented as a pure byte pipe, then these packets would be sent over the satellite as they arrive at the gateway. However, these types of short packet exchanges are the least efficient because they require frequent use of the backchannel for acknowledgments. It would be much more efficient to bundle and send the whole Web page at once rather than to stream it gradually across a satellite connection. In other words, knowing application-level data boundaries may aid in efficient communication (this is the concept of Application Level Framing). Design of split connnection gateways around this approach may outperform simple connection-splicing approaches as described herein.

- **LEO networking** The field of LEO networking is ripe for further work, as there are many more interesting issues involving unicast LEO routing than we were able to cover. For example, integration of a LEO network with terrestrial wireline or wireless networks is an interesting problem. Given a LEO network with multiple gateways, how can the network decide upon which gateway to exit the network? What kind of load balancing, such as routing around hot spots, is needed? Would alternate constellation designs admit simpler routing protocols? Can loop-free distributed routing protocols with support for load balancing be developed? The problems are not limited to unicast routing. The performance of TCP connections in a LEO environment that induces RTT variations could be studied. We did not even touch on multicast routing; what kind of multicast routing protocols are best for a LEO constellation? What are optimal queue sizes for on-board switches? We hope that our *ns* simulator extensions enable research in these areas and in other areas not touched by our research, such as multiple access protocols, handoff strategies, and the performance of other classes of applications such as reliable multicast. The publishing of additional information or research on issues such as the link availability and error performance of LEO links and the terminal and satellite hardware capabilities related to link handoff would aid such future studies greatly.