Networking over Next-Generation Satellite Systems

Tom Henderson and Randy Katz

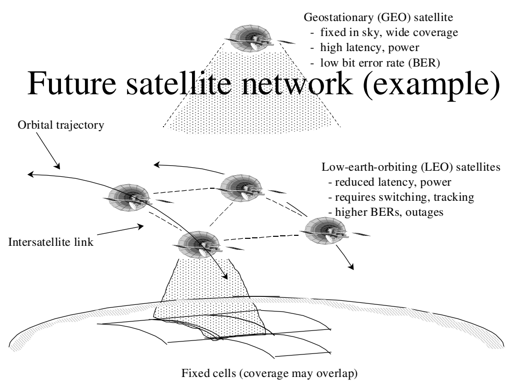
BARWAN retreat: June 11, 1998

Dept. of Electrical Engineering and Computer Science
University of California at Berkeley

What are next-generation satellite systems?

- Advances in satellite technology will enable “Internet in the sky”
  – Geostationary (GEO) satellites for multicast traffic and “lifeline” unicast traffic
  – Low-earth-orbiting (LEO) satellites for low latency unicast traffic
- Through the use of on-board processing, these systems will have global connectivity
- Some examples: Iridium, Teledesic, Celestri

Future satellite network (example)



Service focus

Last-hop IP access is the primary service.

- Interested in fully packet-switched, space routing architecture
- Potentially highly asymmetric access
- Interested in scaling issues

Outline

My research focuses on *routing* and *transport* protocols for next-generation satellite systems

- Summary of transport protocols for satellite networks
  – TCP performance over satellite links
  – Satellite Transport Protocol
- Routing for LEO satellite networks

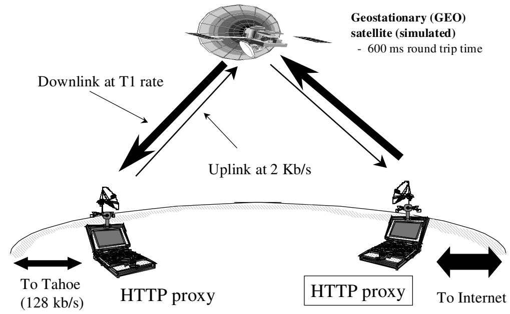
Satellite TCP findings

- Satellite TCP’s two biggest problems are:
  1. TCP is not robust to a mix of connections with different RTTs
     - “congestion avoidance” is inherently unfair
  2. “Slow start” is too slow
     - Venkat’s “fast start” would be an improvement
- “Splitting” of end-to-end connections offers best performing solution to the problem
  – e.g., web caches dramatically improve performance

Satellite Transport Protocol

- An "SSCOP-based" transport protocol optimized for highly asymmetric, high latency links
- BSD/OS kernel implementation and ns implementation/scripts available
- Porting to FreeBSD for NASA ACTS OC-12 experiments this summer

Browse with STP



Proxy settings: 128.32.130.30, port 3128

Transport conclusions

- End-to-end TCP performance over GEO satellite links is problematic
- Split connections perform much better
  - Berkeley TCP modifications (esp. asymmetry and SACK) and T/TCP are most helpful
- Satellite Transport Protocol (STP) optimized for satellite links
  - native satellite transport connections
  - split connections

Routing over LEO networks: service requirements

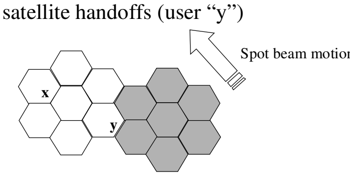
- Best-effort packet delivery
  - User mobility
  - Multicast
- Coarse-grained QoS capabilities
  - Bandwidth reservations
  - Priority packet forwarding

LEO network characteristics

- Network is a time-varying mesh topology with periodicity
- On-board packet switching is low-risk, but architecture must be simple
- Ground-to-satellite bandwidth is at a premium
  - particularly uplinks from user terminals

Handoffs

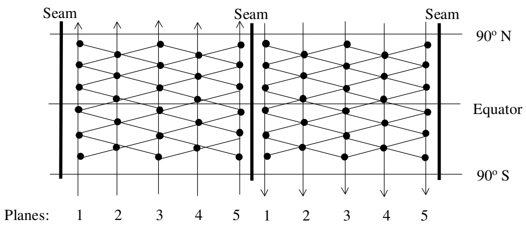
- Two types of handoffs:
  - beam handoffs (user "x")
  - satellite handoffs (user "y")



- Cross-seam handoffs introduce large RTT variations

Handoffs (cont.)

- Cross-seam handoffs:
  - Route to new gateway, or
  - Route over poles

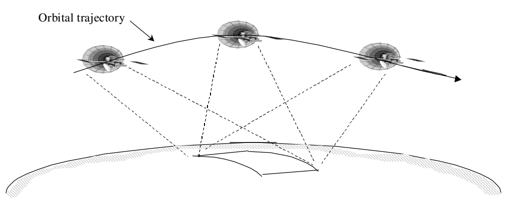


Strategies for simplicity

- Take advantage of network periodicity
  - Satellites embody different logical nodes in a virtual network topology
- Use geographic addresses to simplify routing
- Offload as much routing functionality to ground as possible

Network periodicity

- Earth-fixed cells vs. moving cells
  - Earth-fixed cells could eliminate beam handoffs and reduce the number of network states



Geographic routing

- Add geographic prefix to IP destination address
  - Terminals filter on IP address
- Packet forwarding from distant locations performed solely on prefix
- Packet forwarding in local vicinity based on table lookup
- Satellites exchange routing information for terminals in their local area

Some open issues

- Mobility
  - Nameserver vs. HA/FA handling of mobility
- Multicast
  - Limited set of multicast labels?
- Load balancing
- Policy routing
  - What if both LEO and GEO routes available?
- Reservations
  - Enforce at edge of network?

Design validation

- Use modified *ns* simulator (LEO network simulator) to explore tradeoffs
  - Reduce current implementation in order to conserve memory (scaling)
  - Add capability for time-varying, periodic network topology
  - Design simple traffic generation agents that generate Internet-like workloads