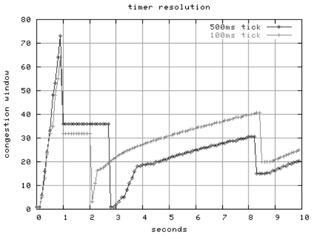
Internet Programming & Protocols Lecture 23

Satellite networks

Proxies

NASA networks

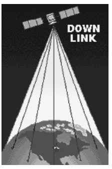
Timer granularity -- assignment 8 (Fig 11.5 text)

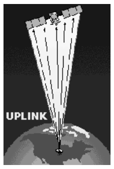


- •Two competing TCP Reno flows with timeouts (ns simulation **tcpTick_**).
- higher resolution timer allows timeout to be detected “sooner”
  - •Check every 100 ms rather than every 500 ms
- •Throughput: 816 kbs with 100 ms timer, 486 kbs with 500 ms timer
- Vegas/FAST/Westwood need hi res timer 0.01

Satellite networks for data communications

- Ubiquitous coverage
  - Good in areas of low subscriber density
- Bandwidth flexibility – dependent on dish size 256 kbs to 100 mbs
  - Tunable per customer
- Cost is independent of distance
- Fast deployment (after initial launch)
- Reliable and secure (backhoe resistant ☺)
- Good for emergency backup to landlines
- Support for broadcast/multicast



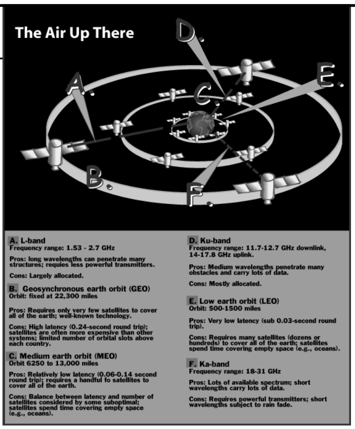


Satellite systems (LEO, MEO, GEO)

- Geostationary satellites (GEO)
  - Broadcast TV, weather observation
- Non-stationary (LEO and MEO)
  - Ground stations need to do buffering for handoffs to the next visible bird
  - Variable RTT
  - LEO’s for telephone service
    - Iridium (66 birds)
    - GlobalStar (48 birds)
    - Spy satellites
  - MEO’s
    - GPS (24 birds, 20000 km altitude)
    - ICO (telephone)

	LEO	MEO	GEO
Altitude(km)	2000	15000	36000
Bw/bird (Gbs)	10	10	50
RTT (ms)	40	260	560

Satellite bands



A. L-band
Frequency range: 1.53 - 2.7 GHz

Pros: long wavelengths can penetrate many structures; requies less powerful transmitters.

Cons: Largely allocated.

B. Geosynchronous earth orbit (GEO)
Orbit: fixed at 22,300 miles

Pros: Requires only very few satellites to cover all of the earth; well-known technology.

Cons: High latency (0.24-second round trip); satellites are often more expensive than other systems; limited number of orbital slots above each country.

C. Medium earth orbit (MEO)
Orbit 6250 to 13,000 miles

Pros: Relatively low latency (0.06-0.14 second round trip); requires a handful fo satellites to cover all of the earth.

Cons: Balance between latency and number of satellites considered by some suboptimal; satellites spend time covering empty space (e.g., oceans).

D. Ku-band
Frequency range: 11.7-12.7 GHz downlink, 14-17.8 GHz uplink.

Pros: Medium wavelengths penetrate many obstacles and carry lots of data.

Cons: Mostly allocated.

E. Low earth orbit (LEO)
Orbit: 500-1500 miles

Pros: Very low latency (sub 0.03-second round trip).

Cons: Requires many satellites (dozens or hundreds) to cover all of the earth; satellites spend time covering empty space (e.g., oceans).

F. Ka-band
Frequency range: 18-31 GHz

Pros: Lots of available spectrum; short wavelengths carry lots of data.

Cons: Requires powerful transmitters; short wavelengths subject to rain fade.

LEO Satellite System Architecture

- **LEO satellite network**
- **70 ... 100 satellites**
- **one satellite connected to 4 neighbors via optical Inter-Sattelite-Links (ISL) (inter- and intra-orbit)**
- **ISLs existing all the time, used for routing/handoff**
- **dynamic bandwidth allocation**
- **2400 ... 300000 users per satellite depending on the allocated bandwidth**
- **Low RTT but may need handoff**

Uplink bitrate	fixed and portable terminals: up to 2048 kbit/s mobile terminals: up to 384 kbit/s in steps of 16 kbit/s
Downlink bitrate	up to 32786 kbit/s in steps of 16 kbit/s
Modulation scheme	QPSK
Access scheme	uplink: MF-TDMA downlink: TDM
Spotbeam diameter	50 km – 500 km
Satellite switch capacity	5 Gbit/s - 10 Gbit/s
ISL capacity	7 Gbit/s - 10 Gbit/s
Downlink data rate per carrier	32 Mbit/s
Maximum number of downlink channels per carrier	32Mbit/s / 16kbit/s = 2000

TCP over satellite links

- There are several factors that limit the efficiency of TCP over satellite links.
  - Long RTT
    - Longer time in slow start decreases throughput.
  - Large Bandwidth-delay product
    - Small window sizes cause under utilization.
  - High Bit Error Rates (10^{-4} to 10^{-7})
    - TCP assumes congestion and decreases window.
    - Inverse square root p law, data rate = $\frac{MSS\sqrt{3/2}}{RTT\sqrt{p}}$
    - p =10^{-7} and RTT 540 ms, max TCP bandwidth = 86 mbs
    - p= 10^{-4} then 2 mbs

TCP's sensitivity to RTT

- Slow-start: double data rate each RTT
- Congestion avoidance: increase cwnd by one each RTT
  - In one second we will add (1/RTT) segments
  - So at end of that second we will have sped up by **MSS/RTT²** bits/sec
  - If you double the RTT, it will take 4 times as long to reach data rate
- Periodic loss (p)

$$\text{data rate} = \frac{MSS\sqrt{3/2}}{RTT\sqrt{p}}$$

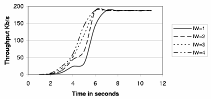
- Droptail unfairness to RTT – longer paths penalized
- Delayed ACKs aggravate problem (maybe do byte counting)
- Bandwidth delay product: buffer size = capacity * RTT
- LEO's and MEO's can have varying RTT
- Multi drops – newreno only does one retransmit per RTT (need SACK)

Solutions for TCP over satellites

- End-to-end solutions
  - TCP tuning
  - Best flavor of TCP
  - TCP hacks
- Non end-to-end
  - Proxies (per hop tuning/protocol)
- combo

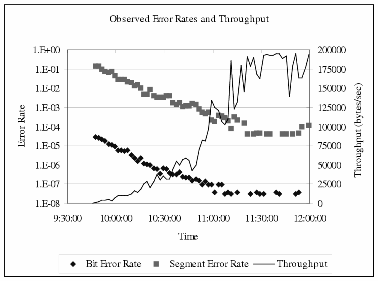
TCP options for satellites

- Bigger MTU will help, path MTU discovery (or preset if pMTUd too slow)
- Need big windows (RFC1323, PAWS/timestamps, window scale)
- Large initial window (4) on slow-start
- Byte-counting to counter delayed ACK effect
- No delayed ACKs for slow-start (linux)
- RED/ECN (RTT fairness)
- Compression
- Aggressive recovery
  - SACK/FACK
  - HS TCP
  - STCP
  - Parallel TCP
  - BI-TCP
  - TCPW (non congestive loss)
    - Some satellite channels may be "dedicated", no contention



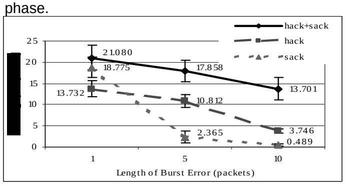
TCP SACK and signal fade

- Bit error rate change as dish aligns properly with satellite (T1)



TCP Header Acknowledgment (HACK)

- TCP HACK is an extension proposed for the TCP protocol to improve its performance over lossy links.
- In HACK, a connection is able to recover uncorrupted header of TCP packets with corrupted data and determine that packet corruption and not congestion has taken place along the link.
- TCP can then respond accordingly and avoid going into congestion control phase.



Throughput for 2% burst error for various burst lengths
(window size of 64KB)

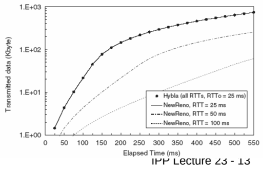
TCP Hybla

- Make TCP on long RTT path behave like it had a short RTT_0
  - make slow-start (SS) and congestion avoidance (CA) independent of RTT
  - normalized RTT $\rho = RTT/RTT_0$ (e.g. RTT_0 is 25 ms)
  - cwnd is calculated $$W^{H}(t) = \begin{cases} \rho 2^{\rho t/\mathrm{RTT}}, & 0 \leqslant t < t_{\gamma,0}, \quad \mathrm{SS} \\ \rho\left[\rho \dfrac{t - t_{\gamma,0}}{\mathrm{RTT}} + \gamma\right], & t \geqslant t_{\gamma,0}, \quad \mathrm{CA} \end{cases}$$

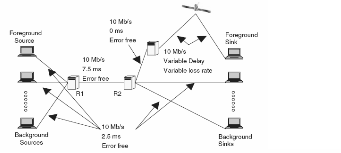
    γ is slow-start threshold (ssthresh)
- Use bandwidth-delay estimation to set ssthresh
- Also does packet spacing to reduce burstiness
- Insists on SACK and timestamps
- Option in Linux 2.6.13

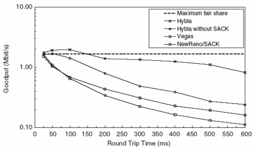
  for SS **cwnd = cwnd + 2^{ρ} – 1**

  for CA **cwnd = cwnd + 2^{ρ}/cwnd**
- Choosing RTT_0 ?

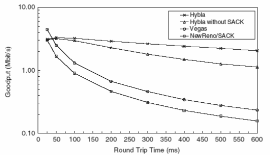


TCP Hybla performance (ns)





congestive loss

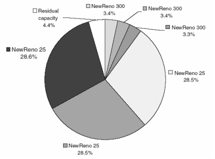


random loss (1%)

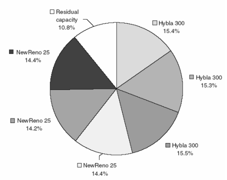
TCP hybla fairness

- Standard TCP RTT unfair

  RTTs 25 ms and 300 ms

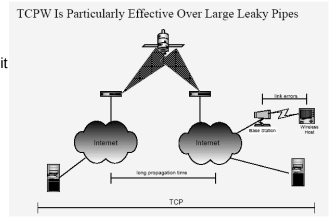


- Hybla with packet spacing

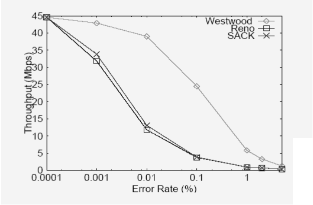


TCPW and satellite paths

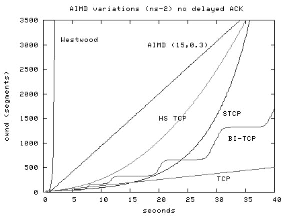
- TCP Westwood sets ssthresh to recent Fair Share Estimate
- Efficiency
  - Better link utilization when loss are due to non-congestive events (random loss, lossy medium (wireless)) as well as congestion
  - Significant gain for large pipe with big RTT
- Better fairness over varying RTTs
- Friendliness good
- Stability good
- If dedicated circuit, use it



TCPW and random loss



If there is no congestion, then fair share estimate results in near instantaneous recovery

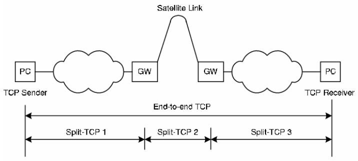


Transport protocols for satellite

- Our old favorite, UDP, use rate-based or other custom mods at application level
- T/TCP (TCP for transactions – request/reply) RFC 1644
  - 3 packets (instead of TCP's 7)
    - 1)→ SYN, data, FIN 2) ← SYN, data, FIN, ACK 3)→ ACK
  - Lower latency – request/response in one RTT
- Satellite Transport Protocol (STP)
  - Reliable stream
  - Automatic Repeat Request (ARQ) -- Selective NACK (SNACK)
    - No reverse path ACK traffic
    - Sender periodically asks for info on good packets received
  - Packet pacing and numbering
- Space Communications protocol (SCPS-TP)
  - SNACK, header compression, timestamps, noncongestive loss response
  - More later
- The latter two are usually implemented in a proxy

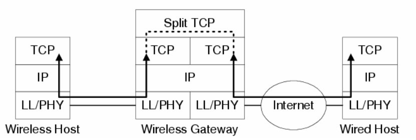
Performance Enhancing Proxy (PEP)

- Insert boxes to isolate long-delay, lossy segment
  - Hack, but hopefully transparent
  - Violates end-to-end principles
  - Think of your NAT box on your homework
    - Transparent translation of your internal net IP addresses
  - Problems with IPsec which “prevents” modifying packets in flight
- Split or spoof or snoop
  - Also see indirect TCP (I-TCP) for wireless



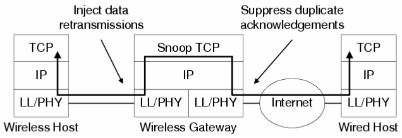
TCP splitting

- Intercept the SYN and opens a separate TCP connection over the lossy link (sort of like what NAT does)
- Gateway maps packets from one flow to the other
  - Could use different flavor of TCP for lossy link (TCPW)
  - Use different buffer sizes, MSS, etc.
- Faster recovery on lossy segment
  - Wired side appears to have shorter RTT
- Violates end-to-end semantics



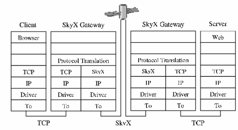
TCP snooping/spoofing

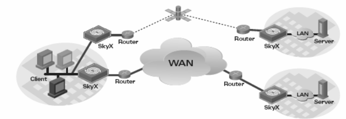
- Objective: confine retransmissions to the lossy link
- Snoop the connection and transparently retransmit corrupted packets
  - Cache packets from wired side to use in case retransmit needed
  - Release when ACK comes back from lossy path
- Doesn’t break end-to-end semantics
- Usually outperforms split TCP
- TCP spoofing sends ACKs from the gateway to sender immediately
  - Illusion of short delay path (violates end-to-end)



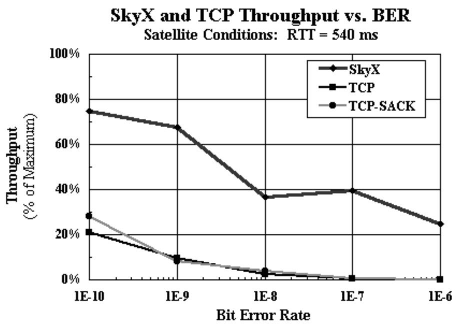
SkyX

- Intercepts TCP connections and replaces it over satellite with “SkyX protocol”
  - uses NACKS to request lost data packets
  - unlimited window size
  - no slow start over satellite link
  - streamlined TCP handshake on initial connection
  - TCP rate control over satellite link
    - Uncongested circuit
    - Loss is due to error
  - Compression
  - Web prefetch
  - multicast





NASA SkyX testing



Israeli Mentat SkyX results (Jan 2000)

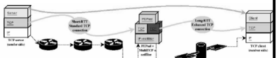
- 30Mb/sec pipe, iperf to U of Oregon
- No SkyX (560ms RTT)
  - 8Kbyte TCP window- 118kbit/sec
  - 64Kbyte TCP window - 646kbit/sec
  - 500Kbyte TCP window - 2.9Mbit/sec
- With SkyX (560ms RTT)
  - 8Kbyte TCP window - 19.5Mbit/sec
  - 64Kbyte TCP window - 18.0Mbit/sec
  - 500Kbyte TCP window - 18.5Mbit/sec

Cidera (SkyCache)

- Data broadcasting at 45Mb/sec
  - Usenet news, web caching, streaming audio/video
- Uses GE-4 satellite over North America (ku-band)
  - requires 1.2 meter dish
  - slightly larger dish needed in Alaska (1.9 meter)
- Maintains 3 uplinks via 7.6 meter dishes in Laurel, MD
- Predictive caching
  - pre-populating the cache with well known popular sites like Yahoo, CNN, ESPN
  - can't handle things like Mars Pathfinder event
- Reactive caching
  - analyze "miss streams"
  - 3 misses worldwide and Cidera prefetches the page
- Relatively inexpensive for these broadcast services

PEP summary

- PEPs isolate long-delay and/or lossy segments
  - Provide "shorter" (split) RTT
  - Generate early ACK's
- PEPs for asymmetric paths
  - ACK filtering
  - ACK reconstruction
  - compression
- PEPs often provide custom congestion protocol
  - Manage non-congestive loss
  - Rate-based (no slow-start)
- PEPs hide link outages by generating repeated ACKs
  - Avoid TCP timeouts and connection shutdown
  - TCP retransmit timeout typically 100 to 500 seconds before RESET


Server
TCP
IP
TCP server (sender side)
Short-RTT Standard TCP connection
PEPsal
TCP
TCP
IP+netfilter
PEPsal + MultiTCP + netfilter
Long-RTT Enhanced TCP connection
Client
TCP
IP
TCP client (receiver side)

PEPsal: a linux proxy box with TCP Hybla

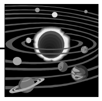
Proxy problems

- Fate sharing
  - If PEP box dies, state information is lost
  - In internet if a router dies, alternate path can keep connection flowing
  - BUT satellite path may be your ONLY path
- End-to-end reliability
  - Early ACKs may lie – packet may not actually make it to end point
  - Application may need to insure end-to-end reliability
- End-to-end diagnostics
  - ping's and traceroute may take different path than proxied TCP
- IPsec requires end-to-end semantics
  - If packets encrypted, PEP may not work
  - Altering authenticated packets would result in failure at the receiver
  - Need workarounds like for NAT

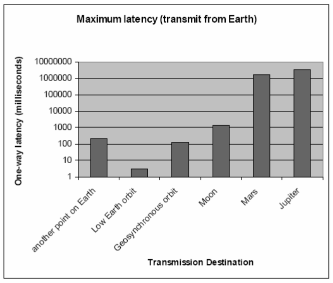
Satellite summary

- Satellite TCP
  - long RTT, higher bit error rate
  - Asymmetric paths
  - BIG windows, SACK, TCPW
  - Proxies to "shorten" RTT and custom transport protocol
- Satellite networking is being outpaced by fiber
  - OC-12 not available by satellite, let alone OC-48, OC-192
  - pricing not able to compete with fiber over last few years
- Satellite networking is excellent for data broadcasting
  - very cheap deals available
- May be your only choice if you're in the jungle, desert, or on a spaceship

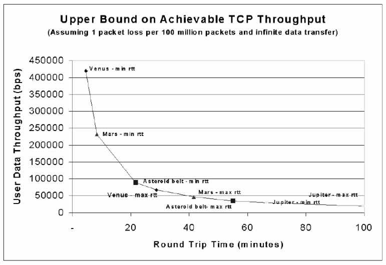
Galactic TCP



- Delay-tolerant networking
  - Lunar RTT 2.6 seconds
  - Martian RTT 14 minutes
  - Alpha Centauri RTT 8 years



TCP performance (inverse sqrt p)



Deep space networking

- Spacecraft have limited power and antenna size
  - Asymmetric nets, low bandwidth
- Bit error rate high (solar wind interference, …)
- Long RTT and intermittent connectivity
- Connection establishment could take days
- Lots of out of order datagrams
- Non-volatile storage for buffers
- End-to-end transmission through relay elements
  - Intermittent connectivity implies store-and-forward
  - Relays should provide point-to-point reliability
- Centrally managed channels ➔ no congestion
- CCSDS protocols
- CFDP – reliable file transfer between spacecraft and ground stations

Space protocols

- Blend of standard TCP/IP at edge nets with custom transport over big delay path
- Reliability at the link layer
  - Forward Error Correction (FEC) can reduce loss
    - Consumes bandwidth whether losses or not
    - No help for sustained loss
  - Need **Automated Repeat Request (ARQ)**
    - TCP does end-to-end ARQ
    - Need hop by hop ARQ
      - Reduce buffer requirements
- Need secure transport, too

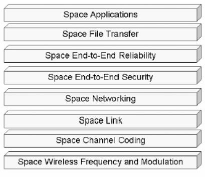
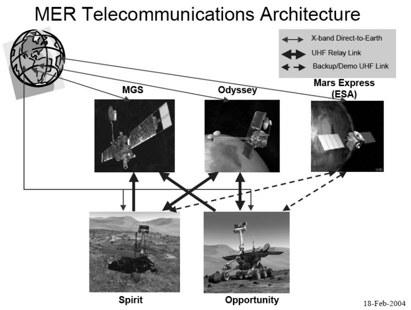

Space Applications
Space File Transfer
Space End-to-End Reliability
Space End-to-End Security
Space Networking
Space Link
Space Channel Coding
Space Wireless Frequency and Modulation

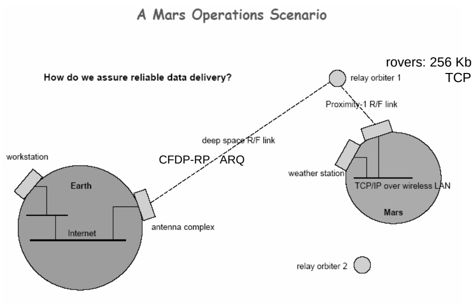
Figure 2: Space Protocol Stack

Mars rovers



- UHF channel is just TV technology, not directional

Mars net

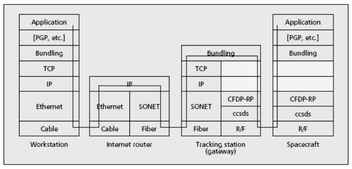


rovers: 256 Kbs uplink
TCP

CFDP-RP ARQ

Delay Tolerant network (DTN) '02

- Internet independent middleware inserted between the applications and the locally optimized stacks
- Violates Internet end-to-end philosophy, but optimizes the transports for specific path characteristics and takes advantage of ARQ
- Unit of exchange in DTN are "bundles" (like email messages)
  - Store-and-forward operations
  - For a file transfer request, bundle together file name, user, password
- Tiered routing, addressing, ARQ, security, congestion control
- Limited resource, scheduled (have to aim antennae)



Next time …

- Wireless networks