XSTP

The eXtended Satellite Transport Protocol: its Design and Evaluation

By

M. Elaasar, Z. Li, M. Barbeau and E. Kranakis

http://www.scs.carleton.ca/~barbeau/Picosat.html
LEO Satellite Access Network

What is the target network? LEO-SAN

What are the error characteristics of LEO-SAN?

- Bit Corruption
- Handoff
- Limited Connectivity
- Congestion
Data Transport over LEO-SAN

What is the goal?
• A reliable end-to-end data transport protocol for LEO-SAN that efficiently deals with its unique error characteristics.

What are the challenges?
1. **Congestion-centric** error control mechanisms
   • Slowing down transmission, underutilizing the link, affecting throughput/energy
2. Limited detection of the changing nature (duration, frequency) of error
   • Poor adaptation of error control, affecting throughput/energy
**LEO-SAN Data Transport Propositions**

- **Different approaches to have a discriminating error control strategy:**
  - Pure link layer: hide link error from transport protocol (ARQ)
  - Cross layer signaling: inform transport protocol of error nature (ECN, ELN)
  - Split path: wired and wireless connections each handling its error kinds (Snoop)
  - End-to-end: sender / receiver detection heuristics (Probing, Wave & Wait)

- **Different approaches to have a LEO-SAN transport protocol**
  - Add satellite extensions to existing protocols like TCP (complex design)
  - Design new satellite-centric transport protocols (highly integrated design)
Selected Propositions

Selected Transport Protocol: Satellite Transport Protocol (STP)
- Highly integrated design incorporating a lot of known satellite extensions
  - Flow control: transmission pacing, byte counting
  - Error control: SNACK, ACK-polling, no ACK timeout
- Good performance in LEO-SAN (although it inherits the same congestion control bias)

Selected Error Control Mechanism: TCP Probing
- Investing some time and transmission effort to discover source of error

![Diagram]

Error Detected → Suspend transmission
Record current RTT
Initiate probing cycle

Terminate only after 2-consecutive successful PROBE/ACK exchanges

Compare probe RTTs with pre-loss RTT
- At least one is greater
- Both are smaller

Congestion Control
Immediate Recovery
eXtended Satellite Transport Protocol (XSTP)

1. Implemented using the *Protocol Implementation Framework for Linux (PIX)*
   - Object-oriented architecture
   - Formal protocol interface
   - Ease of deployment and configuration
   - Access to light-weight libraries

2. Based on the *Satellite Transport Protocol (STP)*

3. Incorporates an extended adaptation of the *TCP Probing* algorithm.

<table>
<thead>
<tr>
<th>Feature</th>
<th>XSTP Probing</th>
<th>TCP Probing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment</td>
<td>Sender only</td>
<td>Sender / receiver</td>
</tr>
<tr>
<td>Semantics</td>
<td>Reuses Polling cycle</td>
<td>Introduces new cycle / segments</td>
</tr>
<tr>
<td>Triggers</td>
<td>SNACK (explicit)</td>
<td>DUPACK (heuristic)</td>
</tr>
<tr>
<td></td>
<td>Early timeout (1 / polling-rate RTT)</td>
<td>ACK timeout (min 1 RTT)</td>
</tr>
<tr>
<td>Integrity</td>
<td>Detects premature activation</td>
<td>Prone to premature activation</td>
</tr>
<tr>
<td></td>
<td>Slow-Start (idle transmission long enough)</td>
<td>Slow-Start (triggered by timeout)</td>
</tr>
<tr>
<td>Speed</td>
<td>Accepts pre-loss Probe ACK</td>
<td>First Probe ACK after at least 1 RTT</td>
</tr>
<tr>
<td></td>
<td>Tolerates a delayed Probe ACK</td>
<td>Ignores a delayed Probe ACK</td>
</tr>
<tr>
<td>Configurability</td>
<td># required probe cycles</td>
<td>no configuration</td>
</tr>
<tr>
<td></td>
<td># outstanding probe cycles</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTT tolerance</td>
<td></td>
</tr>
</tbody>
</table>
XSTP Probing Mechanism

Normal Transmission

- Suspend transmission
- Send and record PROBE

Retransmission required
- Explicit feedback

Normal Transmission

Perform retransmissions

Apply congestion control

Congestion Detected
- Yes (RTT delays)

Probing Done

Record Pre-loss Probe Cycle

Record Probe Cycle

Early timeout

Send and record another PROBE

Wait for a PROBEAK

No (Premature Activation)

Yes

Re-transmission still required

Send and record another PROBE

Every RTT

PROBEAK

No (False Alarm)

Yes

Pre-loss PROBEAK

No (Not enough consecutive cycles yet)
Simulation Framework

- Framework: Protocol Implementation Framework for Linux (PIX)
- Application Protocol (APP): sessions transmit end-to-end bulk data
- Queue Link Protocol (QLP): sessions forward packets through POSIX message queues
- eXtended Delay and Drop Algorithm (XDELDROP) after VDELDROP
  - Sessions are modeled as a continuous time Markov chain with 2 states
  - States are parameterized with Duration, Drop Rate, Delay Range [min, max]

Simulation Configuration
Three Simulation Experiments

- **Three experiment categories**: Bit corruption, Handoff, Limited Connectivity
- Each category simulates random phases of congestion and one kind of link error.
- Individual experiments within a category vary in the frequency and duration of error.
- In each experiment, the performance of XSTP with probing turned ON and OFF is compared.

- The performance metrics:
  - Effective Throughput (bit/sec)
  - Transmission Overhead (%)
  - Throughput / Overhead ratio (bit/sec)

Simulated error states:
- No Error
- Link Error
- Moderate Congestion
- Heavy Congestion
Up to 150% Gain in Throughput

- Bit-Corruption
- Handoff
- Limited Connectivity

1-sec random phases of congestion with one link error
Up to 50% Reduction in Overhead

1-sec random phases of congestion with one link error
**Conclusion & Future Work**

- XSTP probing is an error control strategy that helps sessions get more conservative when error is persistent and more aggressive when error is found to be transient.
- XSTP probing was also integrated with other layers like FTP and IPv4 with DSR routing, and tested over a packet radio network running the AX.25 protocol as a link layer. For more information, please refer to our paper in the proceedings.
- Future work:
  - Enhanced decision making heuristic
  - Reduce probing overhead
  - Different traffic patterns (interactive, HTTP, full-duplex)
  - Different ways of measuring energy
  - Evaluating XSTP probing in TCP
  - Comparing TCP and XSTP probing