Cross-Layer Enhancement of TCP Split-Connections over Satellites Links

Lijuan Wu, Fei Peng and Victor C. M. Leung

Department of Electrical and Computer Engineering

The University of British Columbia

www.ece.ubc.ca/~vleung

IEEE Mini-Symposium on Satellite Communications

Richmond, BC

February 25, 2005

Outline

- Issues of TCP performance over satellite
- Solutions addressing TCP performance issues
- Proposed dynamic congestion control technique
- Simulation results
- Conclusions

Issues of TCP Performance over Satellites

- Geostationary earth orbit (GEO) satellite channels have long propagation delay of around 270 msec
- BER over GEO satellite channels could be as high as 10^{-6}
- Throughput of TCP connection incorporating GEO satellite link can suffer substantial degradations due to:
  - Standard 16-bit windows only allow 64 KB per RTT
  - Slow-start phase could take up to several seconds
  - Packet losses recovered by one packet per RTT
  - Packet losses due to transmission errors are mistaken as congestion resulting in reduction of congestion window or even restart of slow-start; recovery takes a long time

Possible Solutions

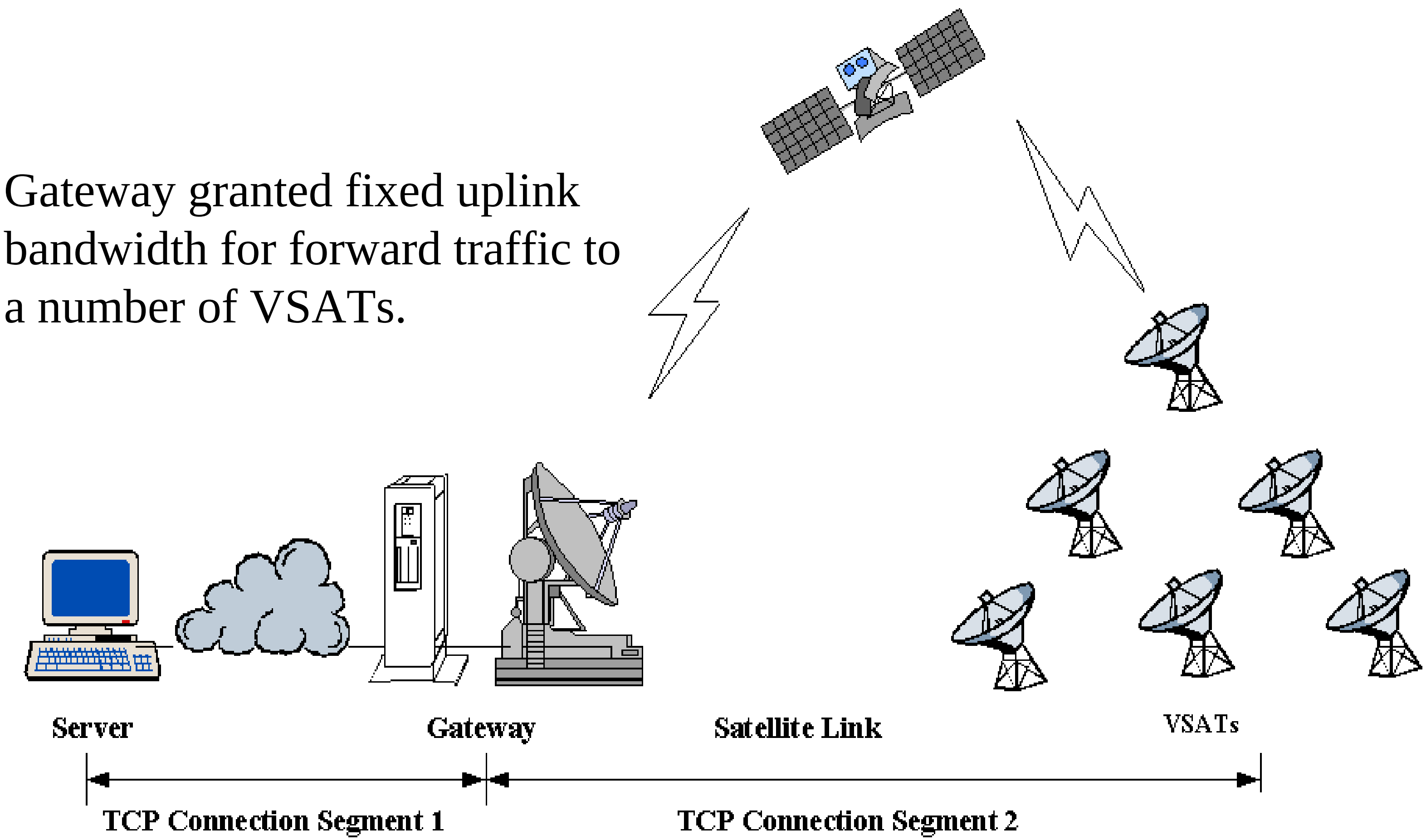
- Link layer solutions to improve packet loss rate
- End-to-end solutions involving TCP options or extensions:
  - Window scaling offers a 30-bit window size
  - Selective acknowledgement (SACK) allows multiple packet losses to be recovered per RTT
  - Large initial window to speed up slow-start
- TCP performance enhancement proxy (PEP)
  - Enhance TCP performance without changing configurations of end systems
  - TCP-spoofing PEP
  - TCP split-connection PEP

Our Approach

- Deploy split-connection PEP to deal with specific characteristics of satellite channels
- Propose proxy service based on TCP Vegas
- Develop novel dynamic congestion control scheme for cross-layer enhancement of TCP
  - In satellite networks with bent-pipe satellite links, system bottleneck occurs at the uplink
  - Possible to provide immediate cross-layer congestion feedback to TCP virtual source from the underlying MAC layer
  - Uncouple congestion control and error recovery

System Configuration

Gateway granted fixed uplink bandwidth for forward traffic to a number of VSATs.



VSATs access a shared uplink by CFDAMA for return traffic.

TCP Vegas

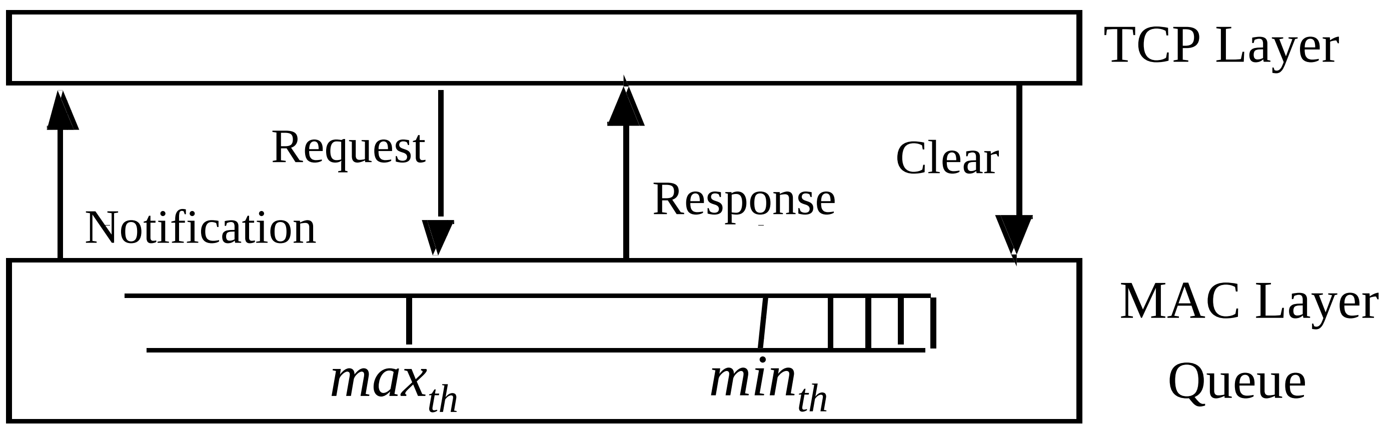
Employs fine-grained timer for more accurate RTT estimate, adjust congestion window cwnd according to changes in RTT:

$$\text{cwnd} = \begin{cases} \text{cwnd} + 1; & \text{diff} < \alpha / \text{baseRTT} \\ \text{cwnd}; & \alpha / \text{baseRTT} \le \text{diff} \le \beta / \text{baseRTT} \\ \text{cwnd} - 1; & \text{diff} > \beta / \text{baseRTT} \end{cases}$$

$$\text{diff} = \text{cwnd} / \text{BaseRTT} - \text{cwnd} / \text{currtt}$$

where currtt is the currently observed RTT, baseRTT is the minimum value of measured RTTs, and α, β (nominally set to 1 and 3, respectively) are thresholds which determine the extra buffers the sender can take.

Cross-Layer Feedback Mechanism



- Satellite uplink – bandwidth bottleneck between PEPs
- Congestion information available locally at PEP from MAC layer
- Gentle-RED AQM employed at MAC queue to generate cross-layer notifications to TCP layer
- TCP requests queue length from MAC and clears indication

Dynamic Vegas (DVegas) Mechanism

- Dynamically adjust α and β using the formula:

$$\alpha = \begin{cases} \alpha + 1/(ratio \bullet 2); & qavg < min_{th} \text{ and } ratio < 1.1 \\ \alpha \bullet 1/(ratio \bullet 2); & qavg > max_{th} \text{ or } ratio > 1.2 \end{cases}$$

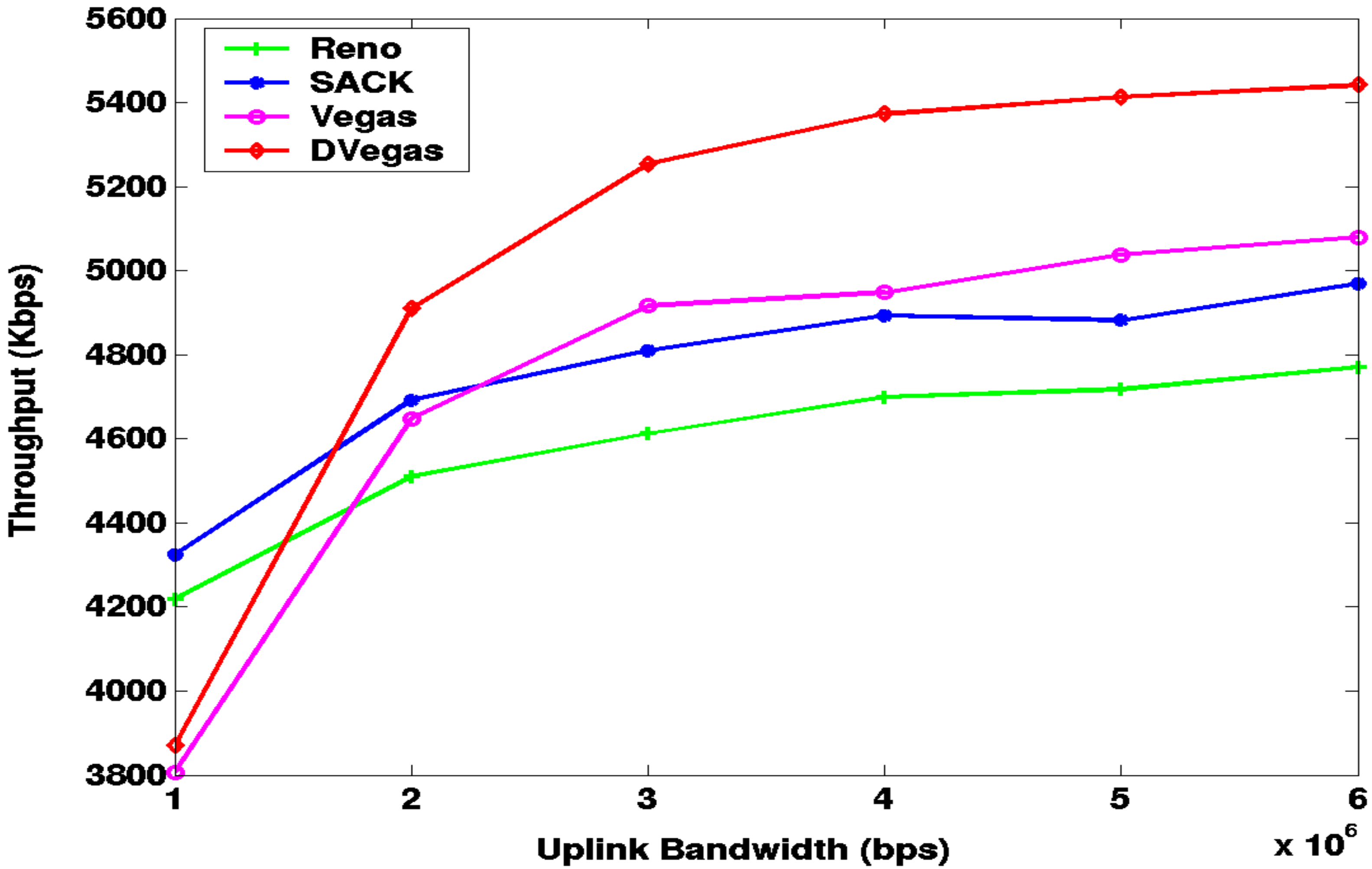
$$ratio = averageRTT / baseRTT \geq 1$$

- In the presence of the congestion notification, TCP queries status of MAC queue and reduce congestion window by
  - ¼ if *qavg* falls between min_{th} and max_{th}
  - ½ if *qavg* exceeds max_{th}
- Slow start mechanism follows regular TCP Reno (double cwnd every RTT rather than every other RTT in TCP Vegas)

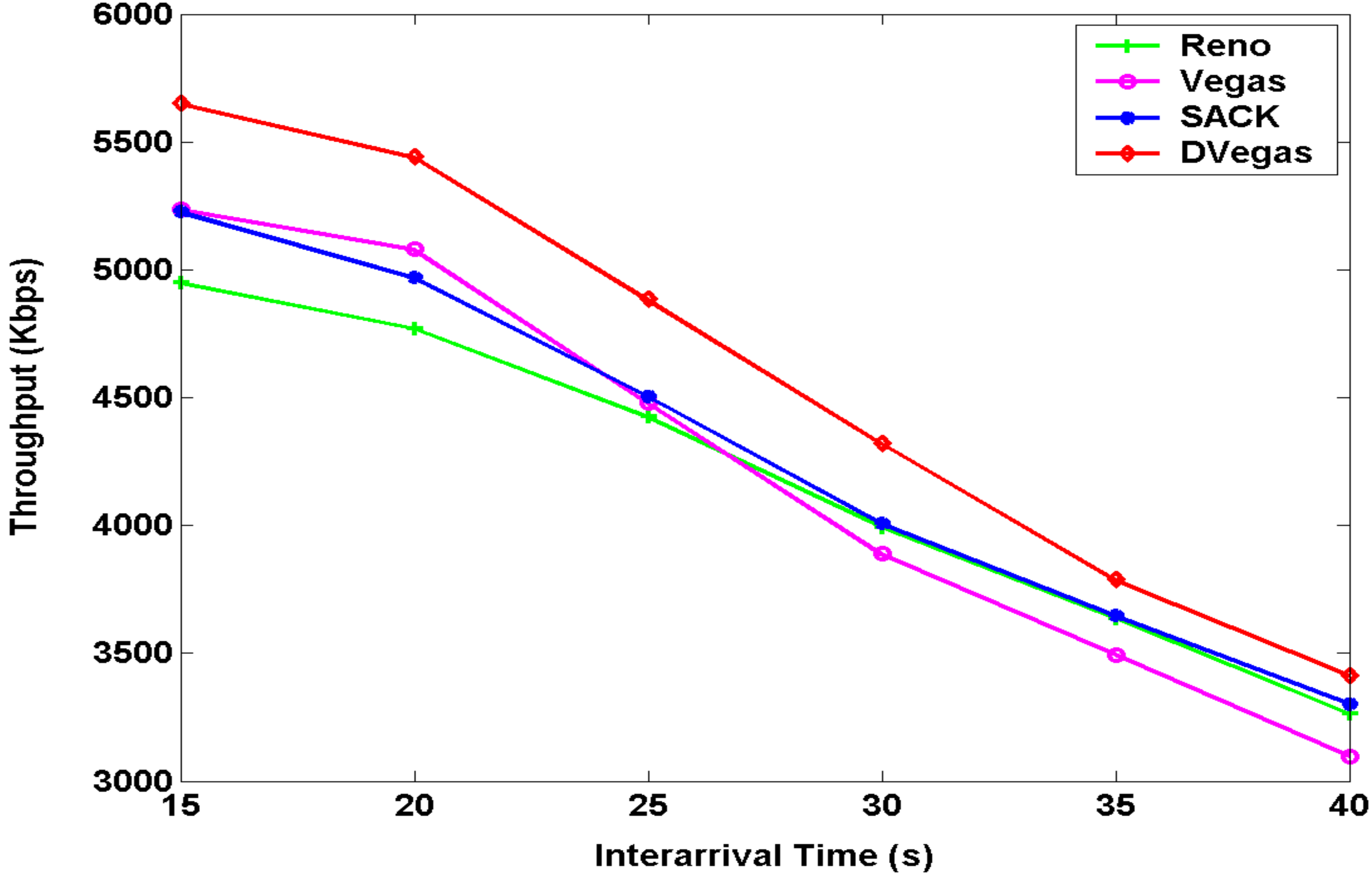
Performance Evaluations

- Simulations using *ns*-2
- Satellite bandwidth = 6 Mbps for forward and return links
- Terrestrial bandwidth = 10 Mbps
- RED parameters: $min_{th} = 5$, $max_{th} = 15$
- 100 TCP connections simulated
- Web traffic simulated with 200 KB file transmissions and exponentially distributed inter-arrival times
- Compare performance of TCP DVegas proxy service against TCP Reno, TCP Vegas and TCP SACK proxy services

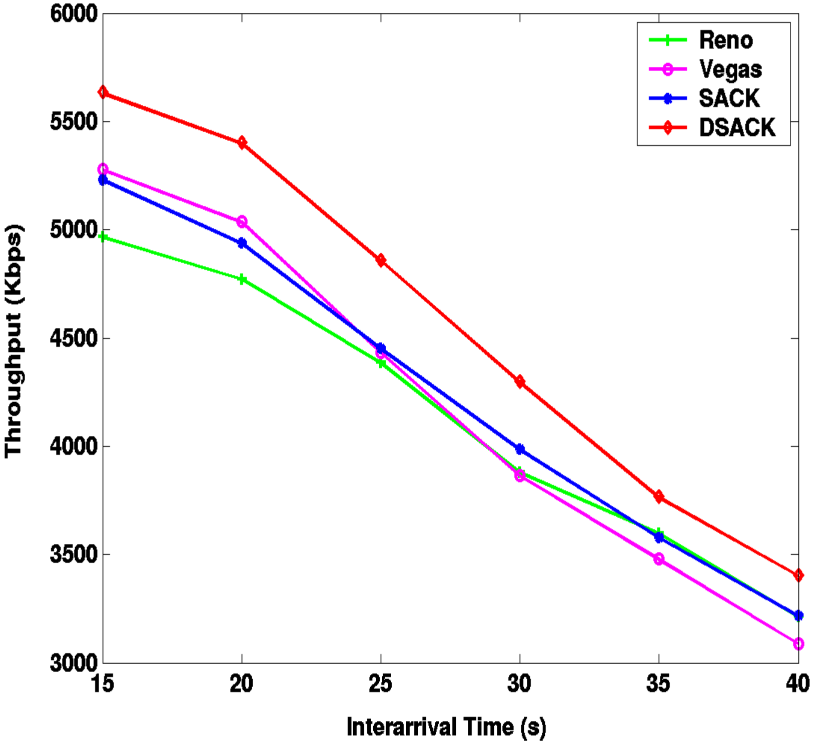
Impact of Uplink Bandwidth

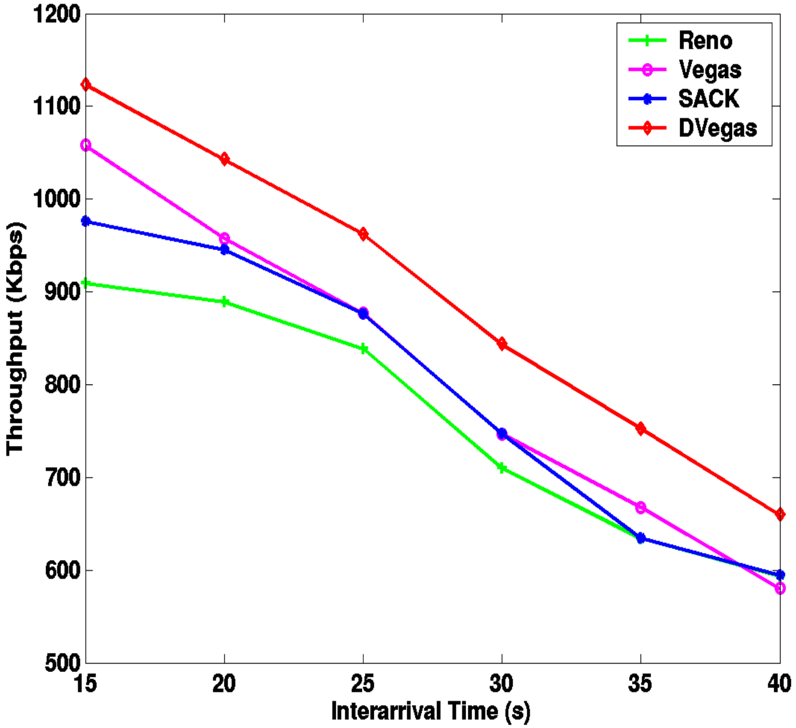


Impact of Traffic Load



Results with 20% Connections Degraded





Overall throughput

Throughput of degraded connections

Conclusions

- Performance degradations of TCP over satellite links can be effectively overcome using split-connection performance enhancement proxies
- Novel proxy service called TCP DVegas presented
- Employs immediate cross-layer feedback from MAC layer to TCP layer to effectively speed up congestion control and separate it from error recovery
- Simulation results presented to demonstrate effectiveness of the proposal

Thank you!