ENSC 835 project
TCP performance over satellite links

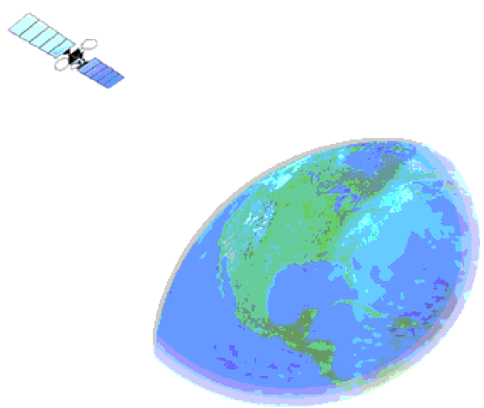
Kenny, Qing Shao
Qshao@cs.sfu.ca
Grace, Hui Zhang
Hzhange@cs.sfu.ca

Road map

- Introduction to satellite communications
- Simulation implementation
  - Window size
  - Initial windows size
  - Maximum segment size
  - Comparison of different TCP flavors
  - TCP burst
- Conclusion
- References

Introduction to satellite communications

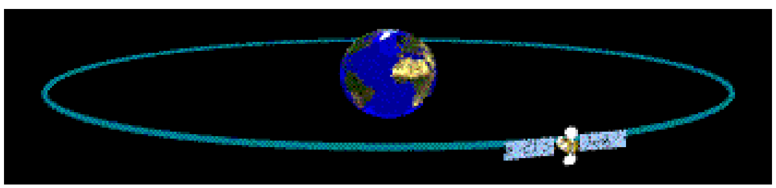
- Satellite communication is a type of the wireless communications technologies. It utilizes satellites to retransmit the wireless signal, and to connect with the multiple earth station.

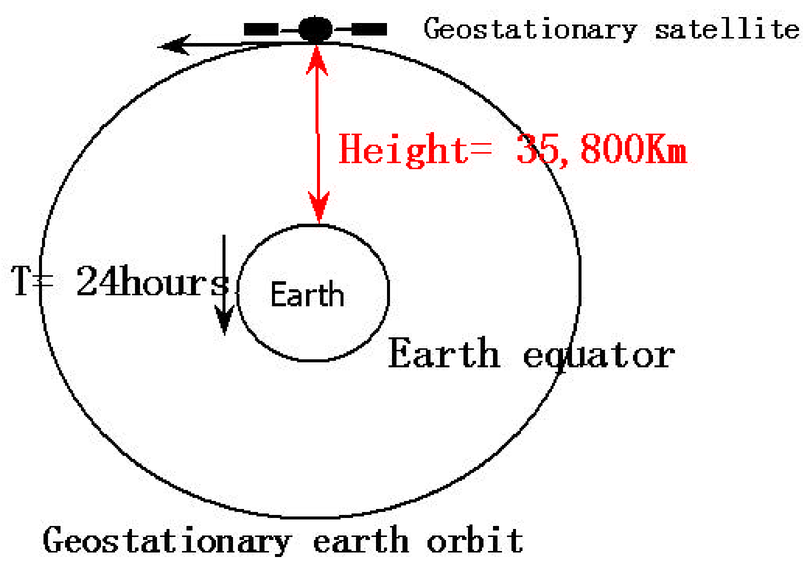


Introduction to satellite communications

Geosynchronous Equatorial Orbit

(From ***geo*** = earth + ***synchronous*** = moving at the same rate.)

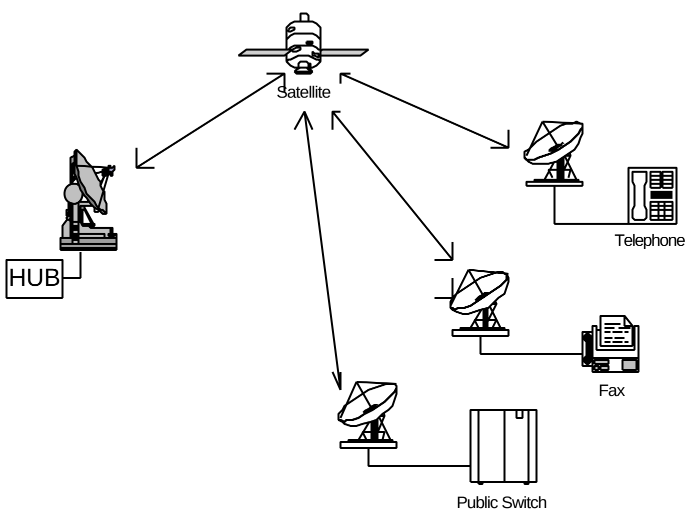


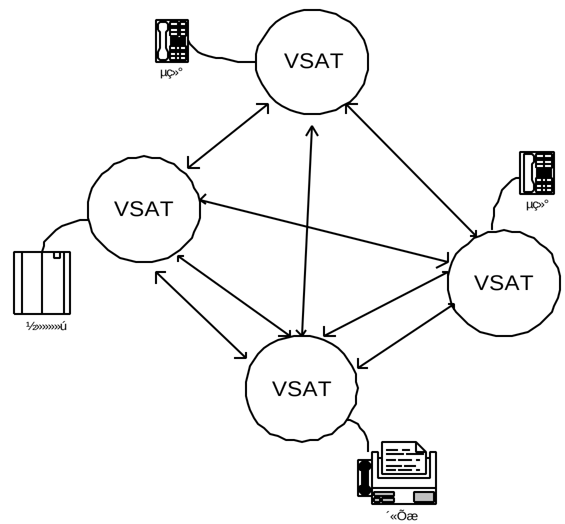


Introduction to satellite communications

Bent pipe GEO satellite

On board switch GEO satellite



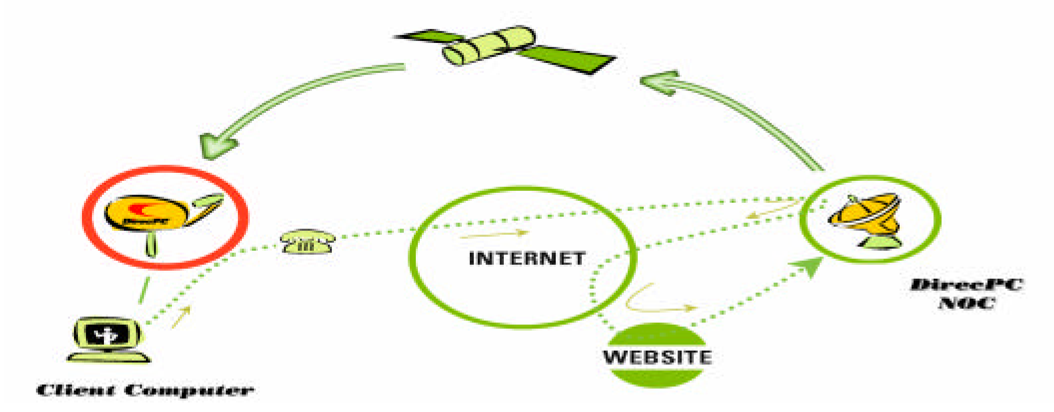


Star network (switch at hub)

Mesh network (switch at satellite)

Why do we choose this project?

- Commercial satellite companies (e.g., Loral, Hughes, Lockheed Martin) have announced plans to build large satellite systems to provide broadband data service.
- Our simulation may help improve TCP performance over long delay and error prone channels.



Project objective

- Implement and study four ways to enhance the performance over satellite links coupling with the characteristics.
- Extend the authors' knowledge of TCP burst problem related to on board switch in GEO satellites.

  - Effect of window size
  - Initial windows size
  - Maximum segment size
  - Comparison of different TCP flavors
  - TCP burst

Window size

- Large delay*bandwidth product

$$W = B * RTT$$

This product defines the amount of data a protocol should have “ in flight”.

Window size

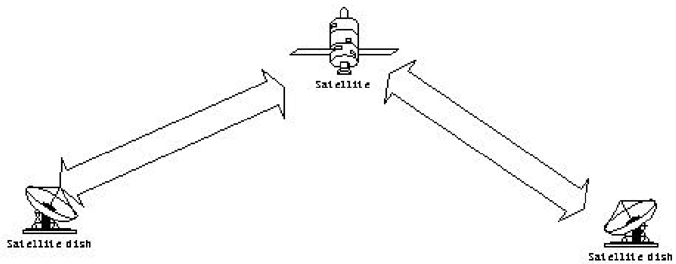
- The original TCP standard limits the advised window size by only assigning 16bits of header space for its value. Hence the advised window size can be no more than 64Kbytes.

$$\textbf{throughput} = \frac{\boldsymbol{CWND}}{\boldsymbol{round\ trip\ time}}$$

$$= \frac{64\ kbyte}{560\ ms} \approx 117,027\ bytes\ /\sec \approx 940\ kbps$$

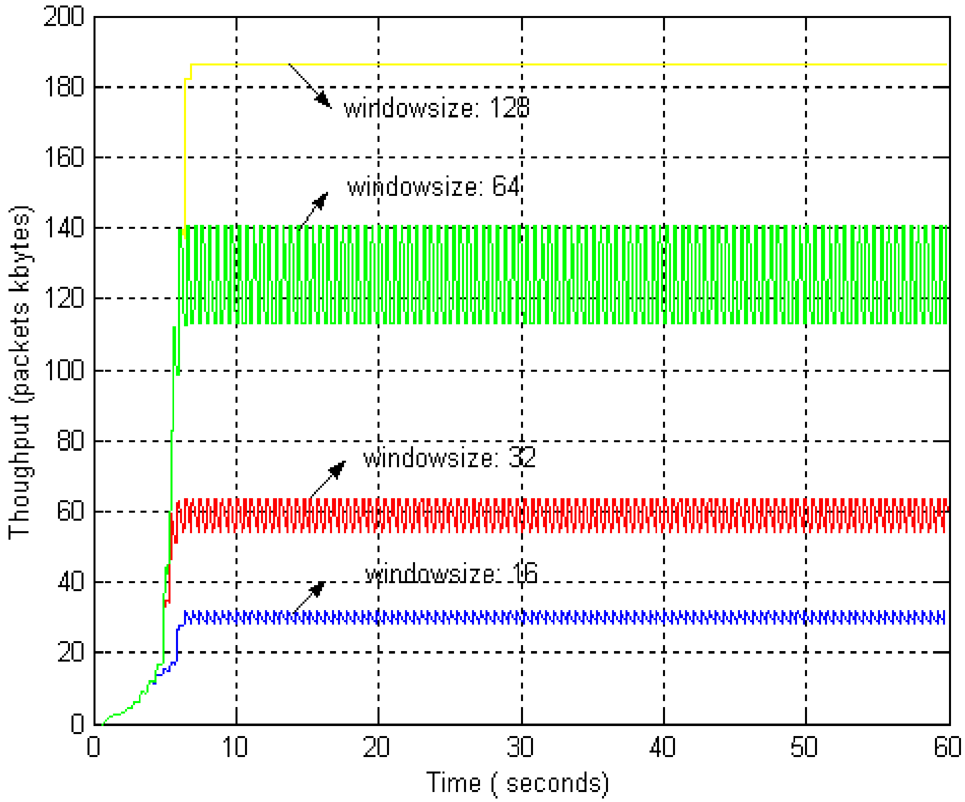
Window size

- Simulation scenario



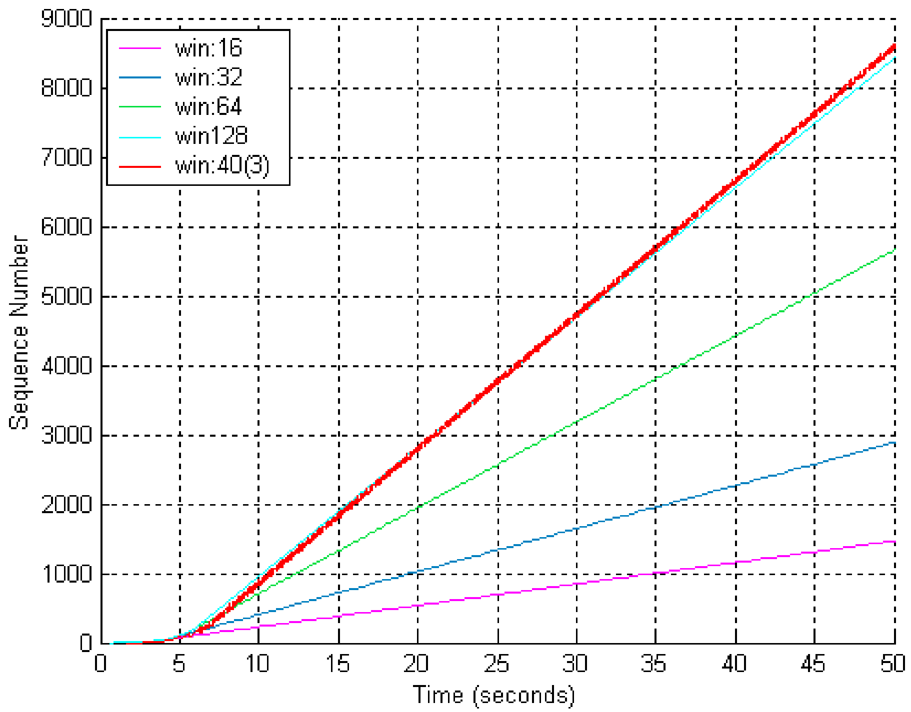
- Parameters:
  - choose various window size: 16, 32, 64, and 128.
  - T1 link: 1.544Mbps.
  - Set the receiver and sender buffer size greater than the delay bandwidth product, so that we can examine how window size affect on TCP throughput: 120.
  - Other parameters are default.

Window size (Result 1)



- Brief Analysis:
  Larger window size can help improve the throughput.

Window size (Result 2)



- Brief Analysis
  - ➢ Multiple long lived connections with small window size can still fully utilize the channel.
  - ➢ Three 40kbytes window connections can almost fully utilize the T1 channel.

Window size: Further discussion

- Problem 1: large window size can lead to more rapid use of the TCP sequence space.
  - Solution: adopt Protect Against Wrapped Sequence Number algorithm (PAWS).

- Problem 2: large window size will also increase the multiple packets loss possibility.
  - Solution: selective acknowledgement.

Initial window size

- Using larger initial window size
  - Advantages:
    - For the connections that transmit small files
    - For the connections that use large congestion window size
  - Disadvantages:
    - Make traffic burst
    - Increase unnecessary drops

Initial window size

The maximum initial window size:

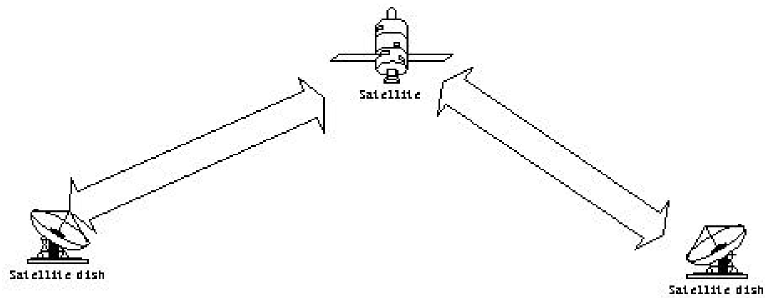
Min (4 * MSS, max (2 * MSS, 4380))

(recommended in RFC 2581)

MSS: Maximum Segment Size

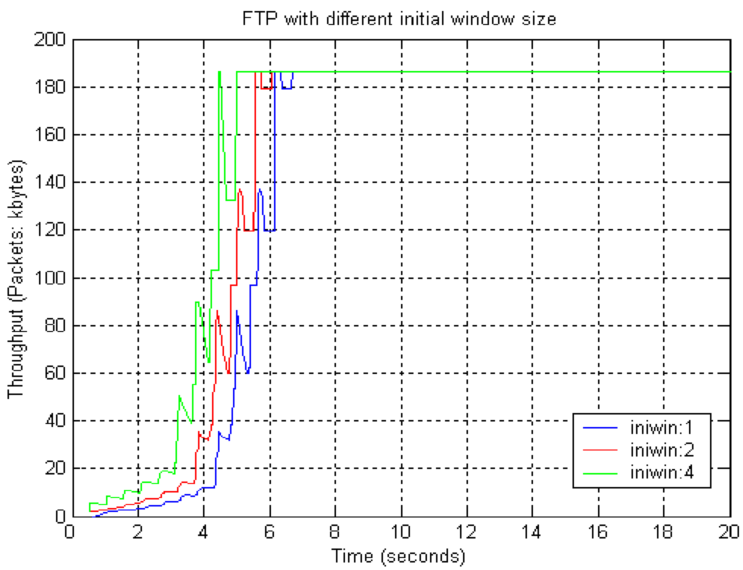
Initial window size

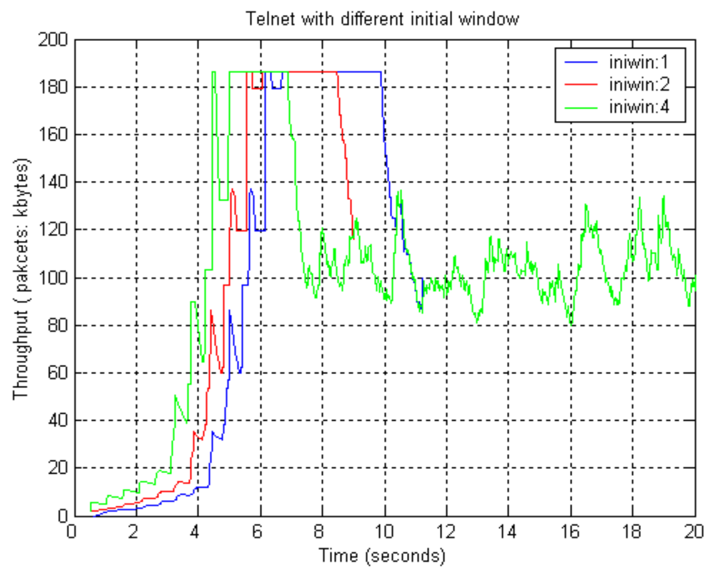
- Simulation scenario



- Parameters:
  - Set the initial window size: 1, 2, and 4.
  - Using different application : ftp and telnet.
  - Set the advised window size: 128.
  - Other parameters are default.

Initial window size (Result 1)

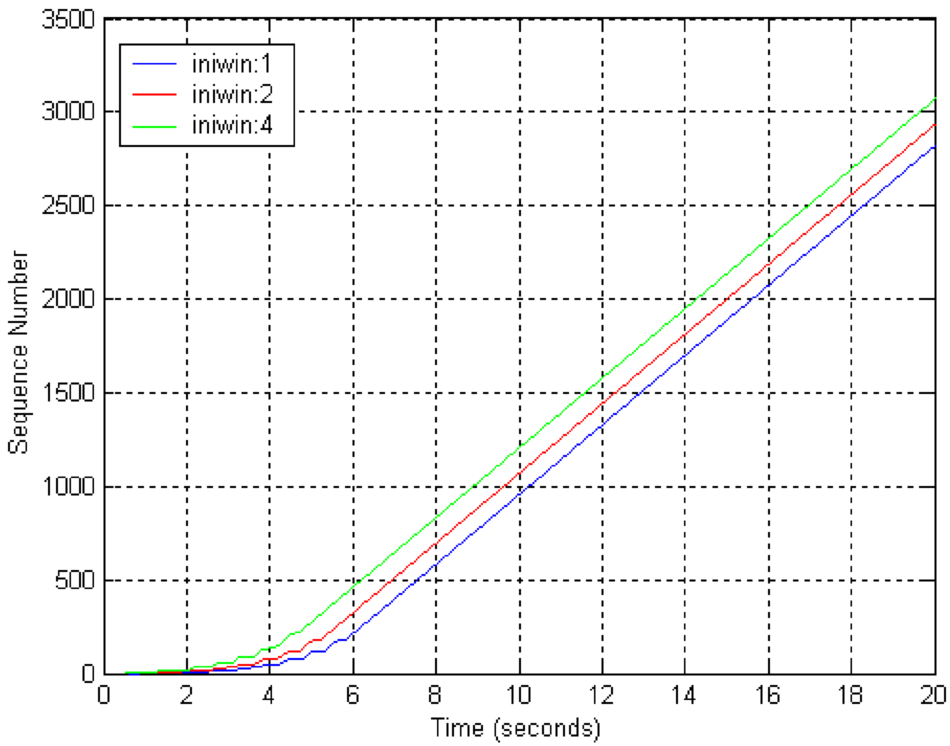




- Brief Analysis:

  Using larger initial window size can help reduce the slow start period, so as to improve the throughput.

Initial window size (Result 2)



- Brief Analysis:
  Starting with larger initial window size allows sender send more packets at the same period of time.

Maximum Segment Size

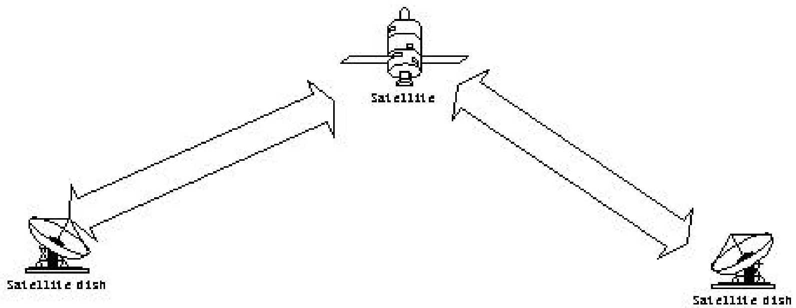
Maximum segment size (MSS):

MSS = MTU – TCP header – IP header

MTU (or the maximum packet size): Maximum Transmission Unit

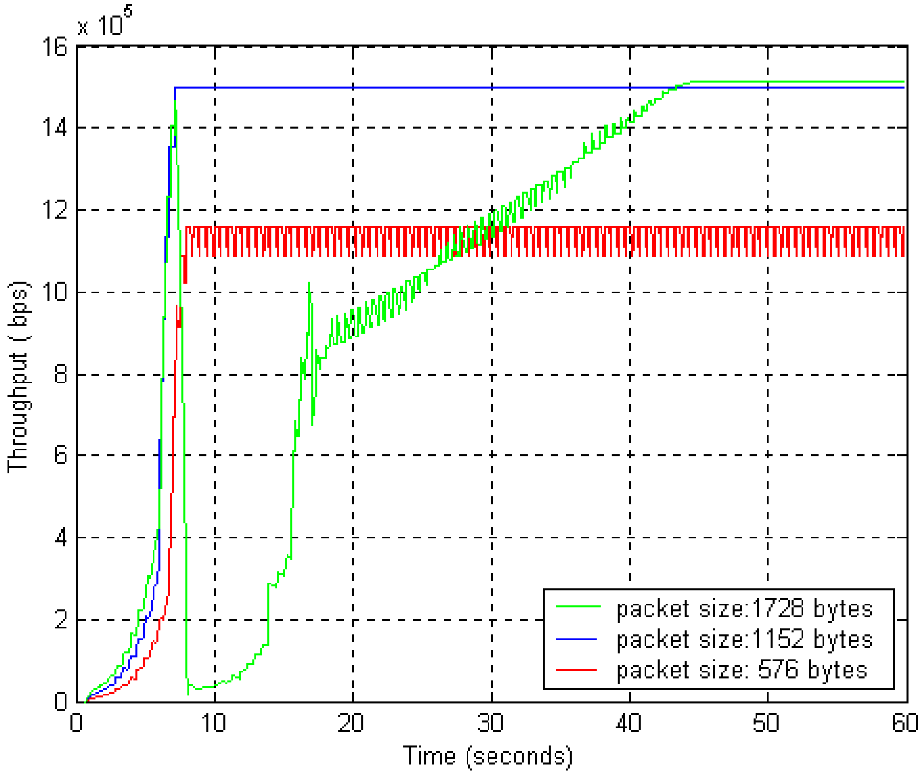
Maximum Segment Size

- Simulation scenario



- We assume that both sender and receiver host are prepared to accept larger segment size.
- Parameters:
  - Set packet size: 576, 1152, and 1728.
  - Set window size:128.
  - Other parameters are default.

Maximum Segment Size (Result)



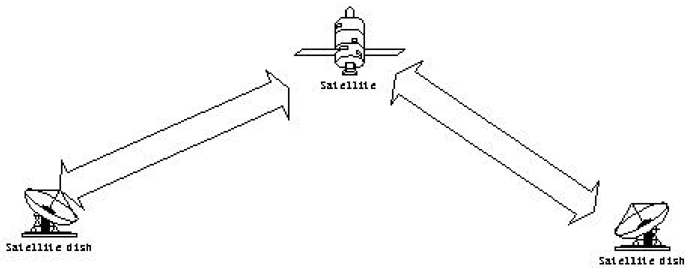
- Brief Analysis:
  Choosing suitable maximum segment size can improve TCP throughput.

Comparison of different TCP flavors

- Using cumulative acknowledgment, the TCP sender can only learn about a single lost segment per round time.
- SACK (RFC 2018) allows TCP to inform the TCP sender which packets have received and which have not.
  - Avoiding reducing the congest window.
  - Avoiding needless retransmissions
  - Avoiding a costly slow start period

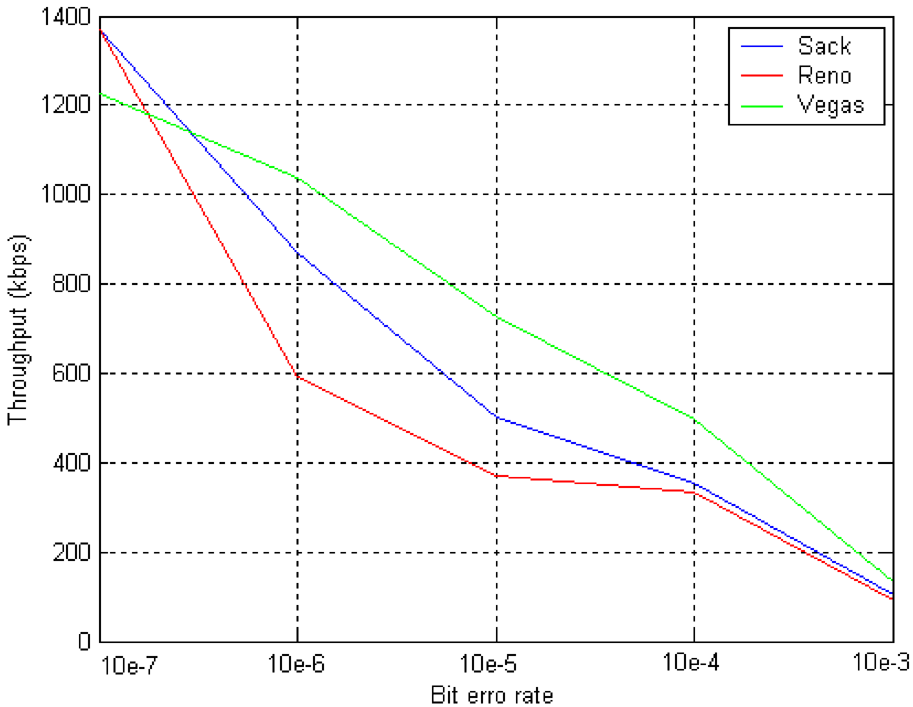
Comparison of different TCP flavors

- Simulation scenario



- Parameters:
  - Add lose model, use different bit error rate : 10e-7, 10e-6, 10e-5, 10e-4, and 10e-3.
  - Compare TCP Sack with other TCP algorithms such as Reno and Vegas.

Result



- Brief Analysis:
  TCP Sack performs better than Reno. However, when bit error rate is from 5*10e-7 ~ 10e-3, Vegas is better than both Sack and Reno.

TCP Burst

- New Generation GEO satellites will use an on board switch to facilitate telecommunication network
- New Challenge for TCP performance: Limited buffer size

TCP Burst

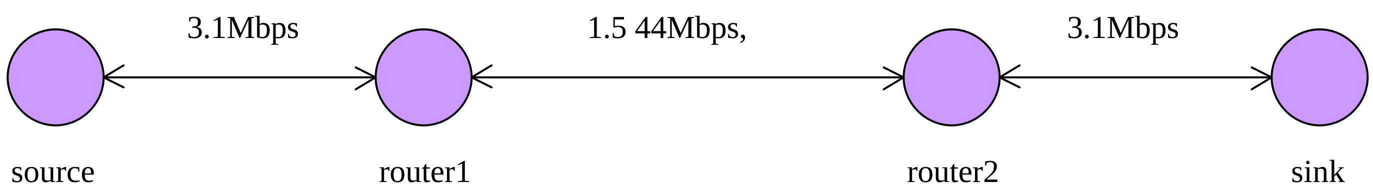
- Why burst?
  - TCP is burst in nature

    Large bursts of data added to the network in a short interval so that creates long queues in the router
  - Large window size Vs limited buffer size

    Usually, in a high delay-bandwidth product link, TCP's maximum window size is much larger than the buffer size of the router.

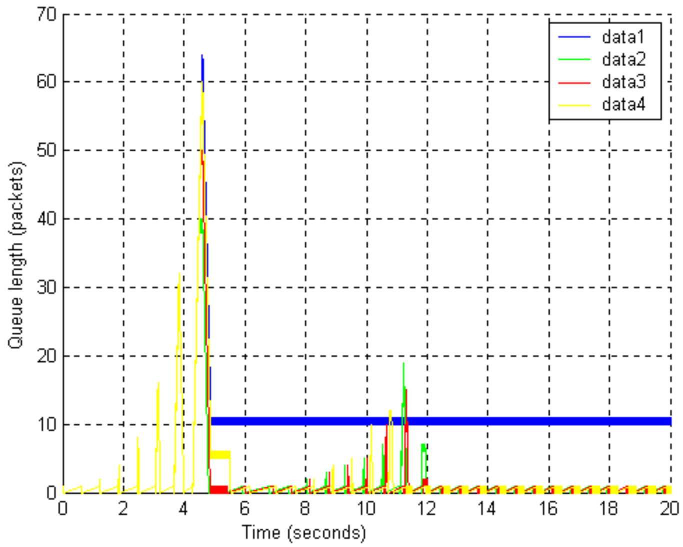
TCP Burst

- Simulation scenario:

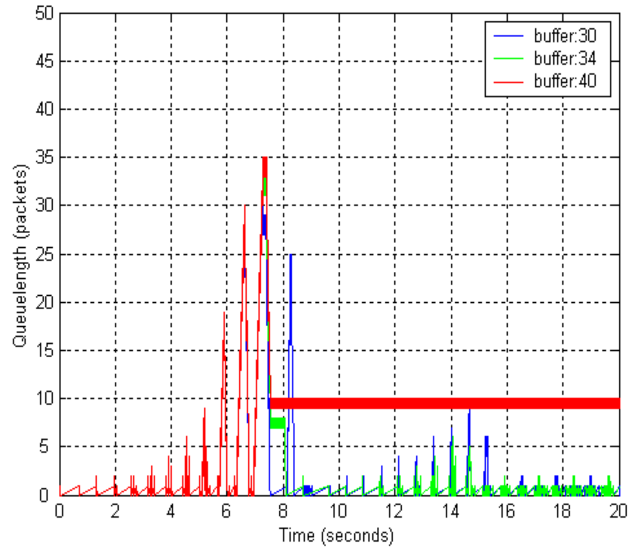


- Parameters:
  - Set various buffer size: 30, 34, 40, and 50.
  - Set the window size: 128.
  - T1 link : 1.544Mbps.
  - Other parameters are default.

TCP Burst



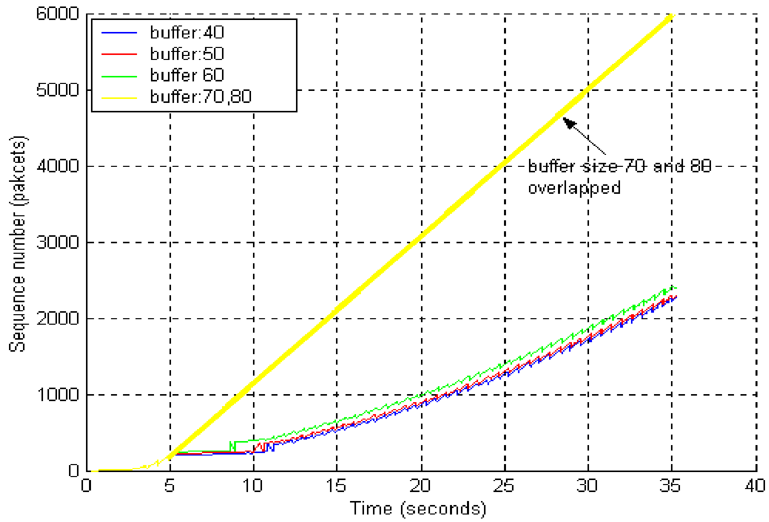
with basic acknowledgment



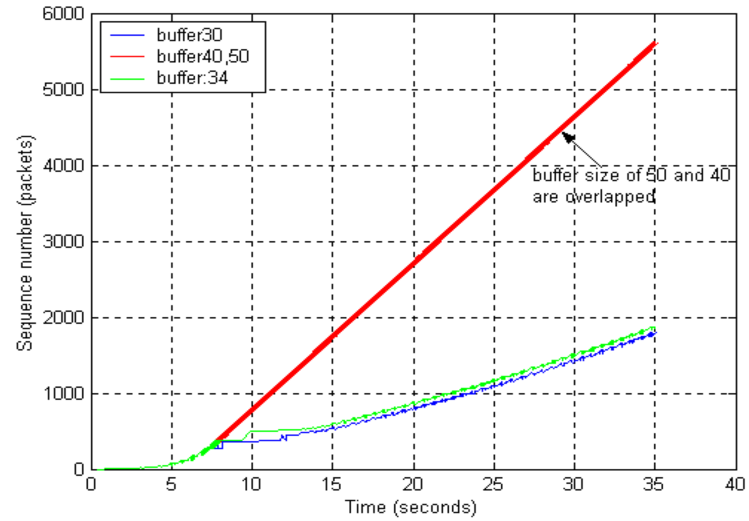
with delayed acknowledgment

- Brief Analysis:
  - The exponential growth of the router queue during initial period represents the macro burst of TCP during slow start , and the lower level of queuing occupancy in later period is due to smaller burst of TCP.

TCP Burst



with basic acknowledgment



with delayed acknowledgment

- Brief Analysis:

  The buffer size impacts on TCP performance severely.

  If the buffer size is approximately smaller than 1/2 of the window size when using basic ack (accordingly, 1/3 when using delayed ack), the throughput will reduce rapidly.

Conclusion

Experiment	Outcome
Window size	**Larger window size can improve the performance**
Initial window	**Using a larger initial window can improve the throughput, especially for short transfers**
Maximum segment size	**Larger maximum segment size will improve the throughput, however, it may cause link congestion or router overload**
Comparison of different TCP flavors	**TCP Sack is better than Reno under the error prone channel. However, Vegas performs even better than Sack if the bit error rate is large**
TCP Burst	**Burst has a severe influence on TCP performance**

Reference

- [1] Van Jacobson. Congestion Avoidance and Control. In ACM SIGCOMM, 1988.
- [2] J. Mo, R. J. La, V. Anantharam, and J. Warland, ``Analysis and comparison of TCP Reno and Vegas," Proceedings of the Conference on Computer Communications (IEEE Infocom), New York, Mar. 1999.
- [3] L. Brakmo, S. O'Malley, and L. Peterson. TCP Vegas: New techniques for congestion detection and avoidance. In Proceedings of the SIGCOMM '94 Symposium (Aug. 1994) pages 24-35
- [4] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and SACK TCP," SIGCOMM Computer Communication Review, 26(3), July 1996

Reference

- [5] A Simulation Study of Paced TCP, Joanna Kulik, Robert Coulter, Dennis Rockwell, and Craig Partridge, September 1999
- [6] [RFC 2488] Enhancing TCP over Satellite Channels, Mailman, and D.Glover January 1999.
- [7] Larry L.Peterson and Bruce S.Davie. Computer networks: A system approach. Morgan Kaufman, 1996.
- [8] [RFC 2581] TCP Congestion Control, M.Allm
- [9] [RFC 1122] Transport Layer TCP
- [10] [RFC 1323] TCP Extensions for High Performance, V.Jacobson, R.Braden, D.Borman, 1992