TCP Performance Improvement

March 26, 2013

Contents

1 Issues for using TCP over Satellite

- TCP faces the following challenges in satellite systems
  - **Long round trip time (RTT).**
    - This delay affects interactive applications such as telnet, as well as some of the TCP congestion control algorithms.
  - **Large delay∗bandwidth product.** The delay is the RTT (end-to-end) and the bandwidth is the capacity of the bottleneck link in the network path.
    - The delay∗bandwidth (DB) product defines the amount of data a protocol should have 'in flight' (data that has been transmitted but not yet acknowledged) at any one time to fully utilize the available channel capacity.
    - Hence, TCP slow-start may under-utilize the system resources.
  - Satellite channels exhibit a **higher bit-error rate (BER)** than typical terrestrial networks.
    - TCP assumes that all packet drops are cased by network congestion and reduces its window size in an attempt to alleviate the congestion.
    - In the absence of knowledge about why a packet was dropped (congestion at the network or corruption at the receiver), TCP must assume the drop was due to network congestion to avoid congestion collapse.
    - Therefore, packets dropped due to corruption cause TCP to reduce the size of its sliding window, even though these packet drops do not signal congestion in the network.
  - **Asymmetric satellite networks.**
    - a larger data rate in the forward direction (from satellite gateway station to user earth stations) than the return direction (user earth stations to satellite gateway station), because of limits on the transmission power and the antenna size at different satellite earth stations.
    - This asymmetry may have an impact on TCP performance.
  - **Variable round trip times**
    - in LEO constellations, the propagation delay to and from the satellite varies over time.
    - This may affect retransmission time out (RTO) granularity.
  - **Intermittent connectivity due to mobility**

* in non-GEO satellite orbit configurations, TCP connections may be handed over from one satellite to another or from one ground station to another from time to time. This may cause packet loss if not properly performed

– **Multiple satellite hops**

* in some situations, network traffic may traverse multiple satellite hops between the source and the destination. Such an environment aggravates the satellite characteristics. This is a generic problem with special circumstances or space communications where there are many more constraints due to long delay, error and bandwidth.

– One of the major problems is that TCP does not have any knowledge about the total data size and the available bandwidth. If the bandwidth B is shared among many TCP connections, the available bandwidth B can also be variable.

2 TCP Improvements

- In order to accommodate satellite environment (more generally wireless links), several solutions are developed. Such solutions may vary from just introducing minor modifications to the protocol to developing a new wireless transport protocol.
- In the following, we will be investigating different well-known approaches pointing out their performance gains and main limitations. These solutions include
  - Split Connection solutions such as Indirect TCP
  - Direct and indirect link layer solutions for error management
  - Selective Acknowledgment option for error management
  - TCP-Peach for long round trip delays
  - TCP-Freeze for handling mobility
  - Proactive TCP-Freeze solutions such as TCP Westwood and TCP Jersey for variable bandwidth.
- To this end, it is worth mentioning that the presented solutions are selected from a large pool of techniques developed for wireless systems. Additionally, it is important to note that many of these solutions may be combined together for incremental performance gains.

2.1 Improving Slow-start for Satellite Systems

2.1.1 Larger initial window

- By increasing the initial value of cwnd, more packets are sent during the first RTT of data transmission, which will trigger more ACKs, allowing the congestion window to open more rapidly.
- The use of a larger initial cwnd value of two segments requires changes to the sender's TCP stack
- Using an initial congestion window of three or four segments is not expected to present any danger of congestion collapse, however, it may degrade performance in some networks if the network or terminal cannot cope with such burst traffic.
- Using a fixed larger initial congestion window decreases the impact of a long RTT on transfer time (especially for short transfers) at the cost of bursting data into a network with unknown conditions.

2.1.2 TCP Peach

- TCP peach is based on the concept of dummy packets. Dummy packets are low-priority segments generated by the sender as copies of the last data segment transmitted as propping packets.
- If there is no congestion, these packets are acknowledged by the receiver and TCP congestion window is increased.
  - If the system is congested, the system drops these dummy packets because they have a lower priority.

  - Hence, TCP peach is identical to TCP Reno during congestion, while it saturates the system bandwidth faster in lightly loaded systems.

- TCP-Peach Algorithms
- Using dummy packets, TCP Peach defines two new phases known as **Sudden Start** and **Rapid Recovery**.
- Sudden Start
  - TCP Peach enters the Sudden Start phase at the beginning of a connection and whenever a timeout expires.
  - The Sudden Start algorithm lasts for one RTT after which the sender enters the Congestion Avoid" phase.
  - During the Sudden Start algorithm, the sender transmits one data packet and (rwnd - 1) dummy packets, where rwnd is the maximum value allowed for the congestion window size given by the receiver.
  - During the sudden start, the sender will increase its congestion window, cwnd, by one packet each time it receives an ACK for a dummy packet. The figure shows the behavior of the congestion window, cwnd, for a TCP connection using the Sudden Start Algorithm.

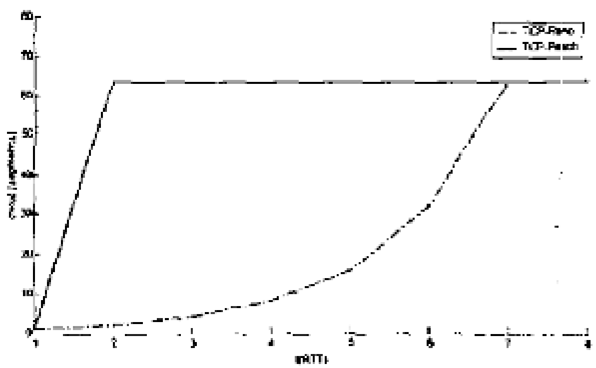

Fig. 1. TCP-Peach and TCP-Reno behavior at the beginning of a new connection.

- Rapid Recovery
  - Rapid Recovery replaces Fast Recovery in TCP.
  - Upon receiving three duplicate ACKs, TCP-Peach executes the Fast Retransmit algorithm and then enters the Rapid Recovery phase which begins by halving the current congestion window cwnd.
  - TCP-peach then generates dummy packets that can fill up to the original congestion window value on receiving the duplicate acknowledgments.
  - Again, if there is a room in the bandwidth, the dummy packets will make it to the receiver and will be acknowledged and the bandwidth is not wasted.
  - Otherwise, dummy packets will be dropped and TCP-peach continues as TCP-Reno.
- Limitation: assumes the network capability of dropping extra dummy packets

References

[1] Zhili Sun, "Satellite Networking: Principles and Protocols," John Wiley & Sons Ltd, 2005