TCP Performance Improvement

March 26, 2013

Contents

1 TCP Improvements

1.1 Handling Bandwidth Variations

1.1.1 Adjusting ssthresh

- In most implementations, the initial value for ssthresh is the receiver's advertised window.
- slow start, TCP roughly **doubles the size of cwnd every RTT** and therefore **can overwhelm the network** with at most twice as many segments as the network can handle.
- Key idea is to estimate the available bandwidth and set ssthresh accordingly.
- **Packet pairs algorithm and the measured RTT** can be used to estimate the available network bandwidth.
  - The algorithm observes the spacing between the first few returning ACKs to determine the bandwidth of the bottleneck link.
  - Together with the measured RTT, the delay $*$ bandwidth product is determined and ssthresh is set to this value.
- Limitations
  - obtaining an accurate estimate of available bandwidth in a dynamic network is very challenging
  - this mechanism will work equally well in all symmetric satellite network configurations.
  - Estimating ssthresh requires changes to the data sender's TCP stack and/or receiver stack.
- Receiver-based bandwidth estimators are more accurate.
- Proactive transport protocols represent another alternative to TCP. These protocols tries to estimate the available bandwidth and adjust their transmission accordingly. Note that TCP is considered a reactive protocol as all the congestion control actions are taken in response to congestion events (duplicate ACK or RTO firing)

- Examples of proactive protocols include TCP Westwood, TCP Vegas

1.1.2 r

- Instead of sending prope packets, TCP Westwood relies on mining the ACK stream for information to help it better set the congestion control parameters: Slow Start Threshold (ssthresh), and Congestion Window (cwin) [in case of three duplicate ACKs (DUPACKs) or a retransmission timeout (RTO).]
- TCP Westwood+ significantly increases throughput over wireless links and fairness compared to TCP Reno/New Reno in wired networks.
- A main limitation in TCP Westwood is lacking the ability to discriminate the cause of packet loss. Therefore, it will adjust the sending rates constantly even upon experiencing packet losses by transmission errors, resulting in a lower throughput in high bit-error-rate (BER) wireless networks. In addition, TCP Westwood's available bandwidth estimation is based on a complex algorithm.
- The first implementation of TCP Westwood in ns2 was done at University of California at Los Angeles (UCLA) in 1999.
- The new version Westwood+ was implemented for the first time in the kernel of Linux 2.4 by Roberto Ferorelli and in the kernel of Linux 2.6 by Angelo dell'Aera, both working at Politecnico di Bari as undergraduate students in 2003 and 2004.

1.1.3 TCP Vegas

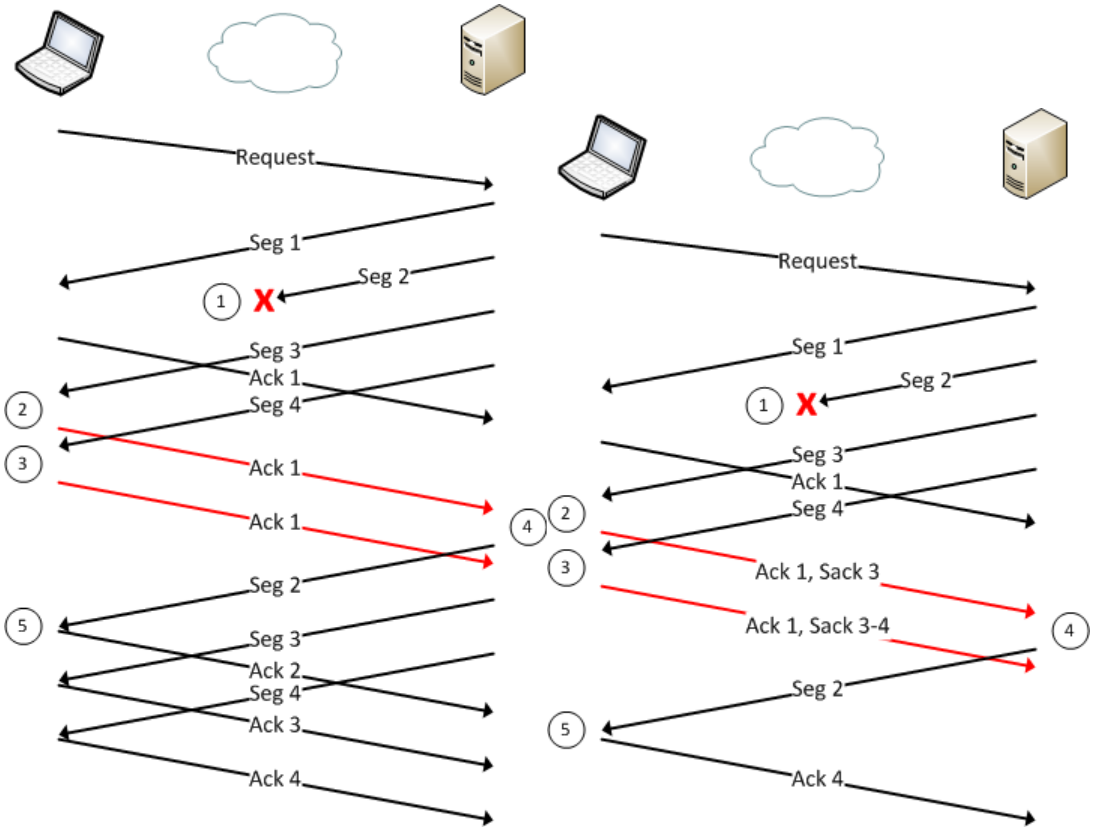
- TCP monitors the RTT and record
  - minimal RTT to derive the optimal network throughput
  - Average RTT to estimate the actual network throughput
- Based on these throughput values, TCP Vegas estimates the backlogged packets in the network.
- Vegas sets two thresholds corresponding (Insert an animation with two lines representing the thresholds for backlogged data)
  - In case of too little backlogged data, TCP increase cgwin linearly
  - In case of too much backlogged data, Vegas decreases cgwin linearly
- A key benefit advantage of Vegas is it proactively adjusts the congestion window without a dramatic change in the congestion window.

1.2 Handling High bit error rate

- TCP's defined behaviour in terrestrial wired networks is to assume that all loss is due to congestion and to trigger the congestion control algorithms and hence starts the slow start.
- There are several approaches to handle wireless channel characteristics. In the following we discuss three well-known approaches identifying their pros and cons. These approaches include
  - TCP selective acknowledgment option
  - Indirect TCP (I-TCP)
  - Link layer solutions

1.2.1 TCP selective acknowledgment

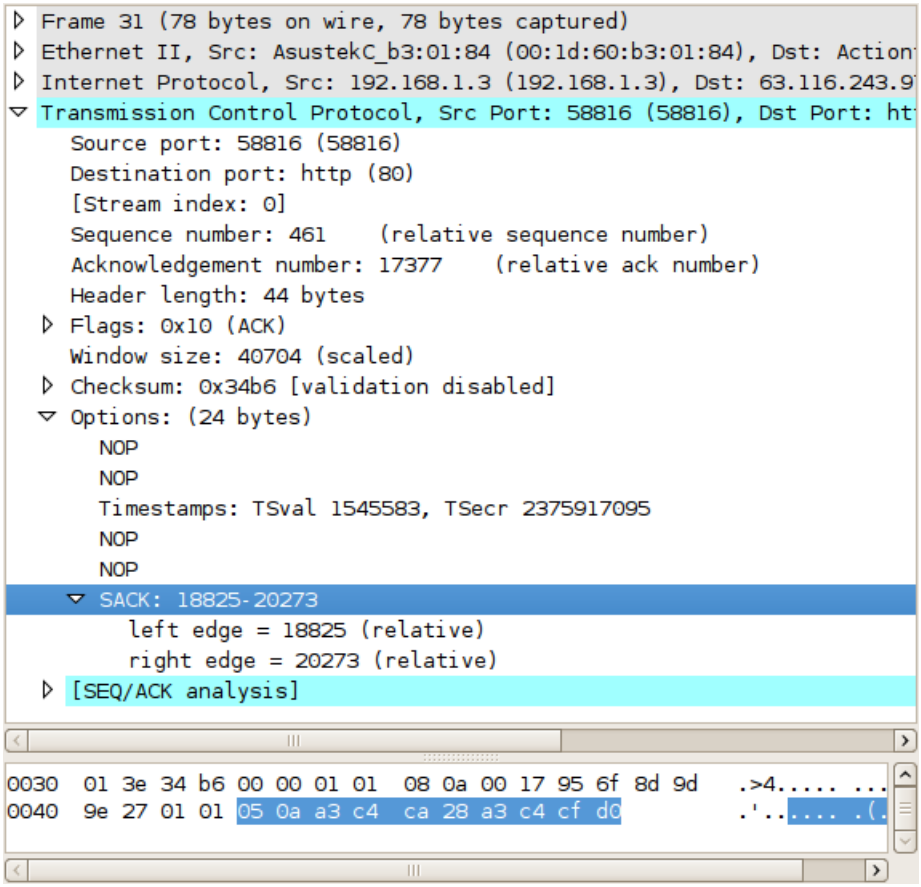
- TCP, even with fast retransmission and fast recovery, still performs poorly when multiple segments are lost within a single transmission window.
- TCP can only learn of a missing segment per RTT, due to the lack of cumulative acknowledgements.
- The selective acknowledgment (SACK) option format allows any missing segments to be identified and typically retransmits them within a single RTT. By adding extra information about all the received segments sequence numbers, the sender is notified about which segments have not been received and therefore need to be retransmitted.



- SACK is proposed (RFC2018).

Trial number	TCP variant	Wireless FER	Mean, median RTT (s)	Mean, median CWND (MSU = 1500B)	Mean, median Normalized goodput	Number of timeouts
1	New Reno	2.6×10^{-3}	1.56, 1.23	59.6, 50.2	0.32, 0.38	16
	New Reno SACK	3.3×10^{-3}	2.03, 1.37	119.0, 63.0	0.39, 0.42	11
2	Westwood+	4.6×10^{-3}	1.83, 1.24	101.0, 54.0	0.31, 0.33	21
	Westwood+SACK	3.9×10^{-3}	1.72, 1.09	78.0, 45.0	0.32, 0.32	12
3	New Reno	1.4×10^{-3}	1.90, 1.30	58.9, 55.0	0.27, 0.28	21
	Westwood+	1.1×10^{-3}	1.65, 1.34	58.3, 56.0	0.30, 0.28	11
4	New Reno SACK	3.1×10^{-4}	3.99, 3.99	196, 170	0.46, 0.45	1
	Westwood+SACK	4.3×10^{-4}	4.05, 4.10	213, 224	0.50, 0.52	2

Figure 1: Mean and median of: RTT, congestion window, and normalized goodput; Wi-Fi frame error rate; number of time outs (T.O.).[1]



- Check for SACK operation in linux (youmay need to be root)

  sysctl -a | grep sack

  net.ipv4.tcp_sack = 1

  net.ipv4.tcp_dsack = 1

1.2.2 Indirect TCP (I-TCP)

- Typically wireless base stations are connected to a reliable wired backbone

- Indirect-TCP splits each TCP connection into two connections at the point where the wireless and wired networks meet, i.e., at the base station (BS).
- BS acknowledges TCP segments as soon as it receives them and the BS takes care of sending the segment to the end host.
- Pros
  - Commonly used in Satellites
  - Transmission errors on the wireless link do not propagate into the fixed network
  - A very fast retransmission of packets is possible
- Cons
  - Obviously violates TCP semantics
  - I-TCP does not handle handoffs efficiently
  - I-TCP is also not suitable for cases where the wireless link is not the last part of a connection path

1.2.3 Link Layer-Aware Protocols: Snoop Protocol

- The awareness here indicates the protocol knowledge of the carried headers. Based on this knowledge, aware solutions usually performs packet inspection (checking the fields of other protocols) .
- Protocol Operations
  1. A snoop agent resides in the base station and inspects every TCP segments.
  2. After packet transmission, agent caches unacknowledged TCP segments
  3. On receiving duplicate ACKs, the agent locally (over the wireless link) retransmits segments
  4. Agent uses retransmission timers similar to the one used by TCP and a packets is retransmitted if it times out.
- Pros:
  - No requirements on wired networks.
  - Improves the performance for small RTTs only L
- Cons
  - Do not perform well on frequent disconnections
  - Will fail if packets are encrypted

1.3 Handling mobility

- Freeze TCP is a modified version of TCP aiming to avoid any expected degradation and packet losses during handoff.
- The key idea of Freeze TCP is preventing unnecessary packet transmission during a handoff by modifying the protocol stack at the serves mobile receiver.
- TCP freeze works as follows
  - The mobile receiver monitors the received signal strength from the base stations. Note that such monitoring is considered a cross layer that benefits from information from lower layer to take actions at a higher layer.
  - If the strength of the signal received by the mobile receiver falls below a certain level, the MN predicts the occurrence of a handoff.

  - The mobile receiver sends a zero window advertisement (ZWA) to the sender. ZWA is originally used for flow control and indicates that the receiver buffer is full.
  - On receiving the ZWA, the sender
    * sets its congestion window to zero
    * fixes values of all other variables such as timers
    * Sender ceases transmission during a handoff as if the connection is frozen.
  - Upon the completion of the handoff, the mobile receiver sends a positive acknowledgment (P-ACK) to the sender with a non-zero receiver window size.
  - On receiving the P-ACK, the sender continues its session using the stored parameters prior to the occurrence of the handoff.

- Using Freeze TCP, the session avoids unnecessary packet transmission and thereby copes with handoff situations in a more proper manner than typical TCP Reno.
- The assumptions made during the development process, however, took into consideration a homogeneous network consisting of network cells with an identical bandwidth only, which lead to a difficulty in obtaining the available bandwidth after a handoff to a new network cell in a heterogeneous network.
- Furthermore, this scheme has another downside that a lost ZWA or P-ACK would affect the performance. For example, a lost P-ACK leaves the sender unable to detect the end of a handoff resulting in severe performance degradation.

1.4 Application Sensitivity

- It is impossible to have one perfect solution for all the different applications without knowing the characteristics of these applications.

1.4.1 Bulk transfer protocols

- Example: File Transport Protocol (FTP)
- Comparing bandwidths of 64 kbit/s and 9.6 kbit/s, throughput was proportional to the bandwidth available and delay had little effect on the performance.
- At a bandwidth of 1 Mbit/s however, window exhaustion occurred and the delay had a detrimental effect on the through- put of the system.
  - Link utilization dropped from 98% at 64 kbit/s and 9.6% kbit/s to only 30% for 1 Mbit/s.
  - however, was still higher for the 1 Mbit/s case (due to reduced serialisation delay of the data)
- Impact of return link
  - At 64 kbit/s link capacity the return link could be reduced to 4.8 kbit/s with no effect on the throughput of the system. At 2.4 kbit/s return link bandwidth, transfer showed a 25% decrease in throughput.
  - At a 1 Mbit/s outbound link speed, the performance of FTP was not affected until the return link dropped to 9.6 kbit/s and started to show congestion (15% drop).

1.4.2 Interactive protocols

- A telnet session allows the user to log onto a remote system
- These applications are more sensitive to delay more than the bandwidth.
- sessions used to view and move directories/files were performed satisfactorily down to 9.6 kbit/s.

References

[1] Tomaso de Cola, Luca Simone Ronga, Tommaso Pecorella, Paolo Barsocchi, Stefano Chessa, Erina Ferro, Alberto Gotta, Gabriele Oligeri, Francesco Potorti, Raffaello Secchi, Arjuna Sathiaseelan, V. M. Castro, R. J. Peral, Carlo Caini, Rosario Firrincieli: Communications and networking over satellites: SatNEx experimental activities and testbeds. Int. J. Satellite Communications Networking 27(1): 1-33 (2009)

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