TCP/IP Over Satellite

Marc Emmelmann,
emmelmann@fokus.gmd.de

Introductory paper to current enhancements and aspects of TCP/IP over satellite channels.[1]

Introduction

TCP/IP, the protocol suite the internet is based on, is now very widely deployed. Originally designed for - in today's terms - rather slow connections providing a good link quality, packet loss is interpreted as an indication of network congestion and appropriate back-off mechanisms are being taken in order to prevent any further overload.

The fact that packet loss due to transmission errors, disclosing to be possibly bursty, might end up in congestion control back-off strategies, reveals the need for TCP/IP enhancements when used over satellite links. Additionally, connections utilizing high speed "wired" links might be limited in terms TCP/IP's efficiency due to the 2^{16}-byte- large sliding window which might be too small compared to a high bandwidth-delay product.

TCP/IP extensions, e.g., the window scaling option proposed by [JBB92], as well as the different "flavors" of TCP handling possibly lost packets (e.g., vanilla TCP, TCP Reno, and SACK TCP) reflect the continuous improvement of TCP's efficiency under the constrains of a high bandwidth-delay product and erroneous links.

Satellite Channel Characteristics

The following main characteristics of an end-to-end path including a satellite channel can be hold responsible for TCP/IP performance degradation: the large end-to-end delay and delay variation[2], high bandwidth links resulting in a large bandwidth-delay product, a high bit error rate (BER) as compared to wired links as well as the

[1] This paper is an extended version. The original document dealing only with LEO satellite systems appeared in [BBW+00a].

[2] Applies only to LEO systems.

burstiness of bit errors, and – in the worst case – asymmetric characteristics between the forward and reverse channel. [HK99]

END-TO-END DELAY AND DELAY VARIATION

According to [GKJ+99], the end-to-end delay (D), as characterized by Equation 1, can be further granulated into transmission delay (t_t), up- and downlink delay (t_{up}, t_{down}), inter-satellite link delay (t_{isl}), switching delay (t_s), and buffering delay (t_q).

$$D = t_t + t_{up} + t_{down} + t_{isl} + t_s + t_q \qquad \textbf{(EQ 1)}$$

Having a satellite environment, the switching and buffering delay will be assumed to be neglectable. From the end-system's point of view, i.e. source and destination of the TCP/IP connection, the transmission delay as defined by Equation 2, can be influenced by the data source itself due to the adaption of packet sizes. Therefore, t_t can be hold neglectable small for delay-sensitive traffic; even when the packet reaches a typical MTU size as considered for UMTS, the transmission delay is about one magnitude less than the total end-to-end delay D and is further disregarded.

$$t_t = \frac{\text{packet\_size}}{\text{data\_rate}} \qquad \textbf{(EQ 2)}$$

Considering the end-to-end delay, it is worthwhile to take a closer look at the up- and down-link as well as at the inter-satellite link delays.

The up- and down-link delays in Equation 3 are assumed to be equal as in the following sections, only the minimum and maximum values of the end-to-end delay D will be further considered.

$$\begin{aligned} t_{up} &= \frac{\text{source\_satellite\_distance}}{\text{speed\_of\_siglnal}} \\ t_{\mathrm{down}} &= \frac{\text{satellite\_destination\_distance}}{\text{speed\_of\_siglnal}} \end{aligned} \qquad \textbf{(EQ 3)}$$

For geostationary orbital satellites having an orbital altitude of approx. 36,000 km the delay for stations near the equator is around 120 ms.

For low-earth-orbital satellites with a minimum elevation of 20^{o} between sender and satellite and an orbital altitude of 1,300 km, the minimum delay is around 4.3 ms and the maximum delay around 12.7 ms. This already reveals the high delay variation of the end-to-end delay D in a LEO satellite environment.

The inter-satellite link distance of GEO systems is stable and depends on the number of satellites in the constellation. For three satellites, the ISL length is about 73,000 km and $t_{isl} \approx 243ms$. Having 12 satellites, an ISL distance of about 21,000 km results in $t_{isl} \approx 73ms$.

Having a LEO satellite environment, the ISL lengths reveal a high variation. As a result, typical values of a 6x12 constellation, i.e. 12 orbital planes each holding 6 satellites, are in between 5 ms and 14 ms for the in inter-satellite delay.

$$t_{isl} = \frac{\sum \mathrm{ISL\_lengths}}{\mathrm{speed\_of\_siglnal}} \quad \text{(EQ 4)}$$

The remaining sections will be based upon the following approximation of the round trip time RTT (i.e.: 2xD).

Round Trip Time	**LEO constellation**	**GEO constellation**
minimum[a]	18 ms (25 ms[b])	240 ms
maximum	120 ms[c]	800 ms

[a] only t_{up} and t_{down}, no ISLs

[b] considering the allowed min. elevation angle of 20^o

[c] This is a worst case approximation based upon [AGS99], [GKJ+99], and [HK99] assuming 5 ISLs and $t_{isl} \approx 110ms$.

BANDWIDTH-DELAY-PRODUCT

The bandwidth-delay-product is the major factor when it comes to the dimensioning of buffers (and window sizes) for the end system [GJF+99]. The following table provides an overview of characteristic bandwidth-delay products for a LEO and GEO satellite constellation.

Bitrate	**Bandwidth-Delay-Product — BDP (min. and max. values)**		**Comments**[a]
	LEO Constellation	GEO Constellation	
16 kbps	36 Byte	492 Byte	Min. bitrate uplink and downlink
	246 Byte	1.6 kB	
64 kbps	148 Byte	1.92 kB	
	948 Byte	6.4 kB	
384 kbps	886 Byte	11.52 kB	Max. bitrate uplink (mobile terminal)
	5.76 MB	38.4 kB	
512 kbps	1.108 kB	15.36 kB	
	7.68 kB	51.2 kB	
1.024 Mbps	2.304 kB	30.72 kB	
	15.368 kB	102.4 kB	
2.048 Mbps	4.608 kB	61.44 kB	Max. bitrate uplink (fixed terminal)
	30.72 kB	204.8 kB	
32.768 Mbps	73.728 kB	983.04 kB	Max. bitrate downlink
	491.52 kB	3.2 MB	

[a] Referring to design consideration within the ATM-SAT project.

As the buffers reside in the end-systems, buffer sizes of several MB are feasible besides that the BDP represents the link capacity shared by all TCP connections and at most 50% of the BDP is a feasible buffer size.

ERRONEOUSNESS OF LINK

The radio signal strength falls in proportion to the square of the distance traveled resulting in a rather low signal-to-noise ratio. Additionally, some frequencies are particular prone to atmospheric effects such as rain attenuation. In particular for mobile applications, multi-path distortion and shadowing (e.g., blockage by buildings) is a relevant factor when considering the link's bit error rate. For today's satellite link, a BER of 10^{-7} or less is typical. Advanced error control coding allows to achieve a error performance comparable to today's fiber [AGS99].

Therefore, for GEO satellite systems a BER between 10^{-9} and 10^{-10} will be assumed. Even though multipath distortion and shadowing cause a variable BER of LEO satellite systems, LEO links will be assumed to be "upgradeable" to either of the two following states: error free or unavailable [HK99]. The resulting burstiness of losses affect the choice of an appropriate TCP/IP implementation.

Improving TCP/IP over Satellite Channels

When taking a closer look at TCP/IP and how to improve its performance over satellite channels, two main approaches can be identified [GJR+99a]: (1) end-system policies and (2) network policies.

End-system policies mainly deal with congestion avoidance and data recovery mechanisms represented by the different TCP/IP protocol implementations ("flavors"). They reflect the long, fat pipe problem of high bandwidth connections with a long round trip time (RTT) and apply as well to satellite channels as to high speed wired links.

When talking about network policies, parameters (i.e. buffer schemes, drop policies, and minimum rate guarantees) are tuned by the network operator in order to optimize resource utilization. From the network operator's point of view, these optimizations are not necessarily forced by TCP/IP constrains.

END-SYSTEM POLICIES

Standard TCP serves as a reliable end-to-end, streaming data service to applications. It receives arbitrarily sized chunks of data and packages them in variable-length segments, each indexed by a sequence number. The receiver's ACKs contain the number of the next expected byte of data in order to achieve a continuos data stream in its buffers; therefore, duplicate ACKs indicate a lost or corrupted data segment. Basic TCP interprets any segment loss as an indication of network congestion and reduced its pace at which it floods the network with data [HK99].

TCP maintains a variable called congestion window (CWND) reflecting the number of unacknowledged outstanding packets; the variable called slow start threshold (SSTRESH) marks the point at which the function increasing CWND switches from exponential to linear behavior. The exponential increase, known as slow start algorithm, doubles CWND for each received ACK; congestion avoidance, the linear algorithm, increases CWND by one for each received ACK.

Currently, four major ways of TCP congestion control and avoidance, i.e., its way to handle packet loss, are well known: Vanilla TCP, TCP Reno, TCP New Reno, and SACK TCP [GJK+99].

Vanilla TCP. The implementation of *slow start and congestion avoidance* is mostly referred to as Vanilla TCP. The detection of congestion is only based upon the expiration of the retransmission timer. After SSTHRESH is set to vale of CWND, the slow start algorithm is applied to further increase CWND with a new initial value of one. Congestion avoidance takes over as soon as CWND reaches SSTHRESH.

TCP Reno. The TCP Reno implementation is based upon the fast *retransmit and recovery* algorithm. As an acknowledgment is sent for each (second) packet received[1], a duplicate ACK containing the same number of byte expected to be received, indicates a lost (or late) package. After three duplicate ACKs, the sending TCP retransmits the missing packet immediately. After this fast retransmit and the acknowledgement of the lost package, the fast recovery algorithm halves CWND and artificially increases it by one for each duplicate ACK. Afterwards, the congestion avoidance phase is entered. This behavior allows to recover from one lose segment within one RTT.

It should be noted that a lost package causing several duplicate ACKs may actually result in several reductions of CWND within one RTT.

TCP New Reno. TCP New Reno is a set of bug fixes of the original [HK99]. It avoids multiple window reductions in one window of data and constrains the burstiness of the sender upon leaving fast recovery.

SACK TCP. Selective acknowledgments (SACKs) are used by the receiver to provide exact information about the packets correctly arrived. During the fast re-transmission phase, the sender first retransmits all suspectly lost packets before sending new ones. This allows to recover from several lost segments within one RTT.

Obviously, *Vanilla TCP*'s efficiency is suboptimal in the context of a long RTT as lost packets depend on the expiration of re-transmission timers. The (unmodified) fast re-transmit and recovery algorithm, as implemented in *TCP Reno*, is detrimental to TCP performance due to the burstiness of packet loss in a satellite environment. *TCP New Reno* fixes this handicap but is still outperformed by *SACK TCP* allowing to recover from several packet losses within one RTT.[2] [GJF+99]

Besides taking a closer look at the different “flavours” of TCP and how they behave in a satellite environment, tuning these implementations is strongly advised. Actions should be taken in order to adapt (1) the standard window size, (2) avoid fragmentation of TCP/IP datagrams, (3) determine an appropriate re-transmission time-out, and (4) improve performance for short transactions [HK99].

Window Size. Standard TCP/IP allows a maximum window size of 65,535 bytes. As previous considerations revealed the large bandwidth-delay-product in a satellite environment, the former window size is not adequate to fill the “channel pipe” with data. The window scaling option as defined in [JBB92] increases the window size to a maximum of 2^{30} bytes (1GB) which is sufficient for LEO constellations and most GEO systems. Besides, the probability is high that several TCP connections will be simultaneously present on a satellite channel, reducing the need for window sizes equivalent to the bandwidth-delay-product.

[1] If the receiver employs delayed ACKs, only every second segment is confirmed.

[2] It should be noted that SACK TCP might worsens the throughput when it comes to package loss due to severe congestion. This situation is neglectable if network policies are applied to improve TCP/IP efficiency.

Path MTU measurement. In order to reduce the cost of fragmentation and reassembly, Path MTU discovery as defined in [MD90] should be performed. Ideally, the PMTU size minus 40 bytes (20-byte IP header and 20-byte TCP header) corresponds to the maximum segment size (MSS), i.e., the largest "chunk" of data that TCP will send to the other end. Even though the Path MTU option allows to determine the most efficient packet size, the iteration of reducing an initial (rather large) transfer unit due to forwarding-router's neglects causes a delay before TCP is able to start sending data [AGS99], [Ste98]. TCP implementations might be "hands-on" tuned if they reside at clients permanently connected via a satellite connection; these instances may always start TCP connections with a MSS proposal adequate to the satellite channels characteristics.

Round Trip Time Measurement. Well chosen values of the re-transmission timeout (RTO) become essential when dealing with large congestion windows as an a prior expiration of the RTO results in heavy, unnecessary retransmits. Round trip time measurements utilizing the time-stamp option are a recommended. Timestamps proportional to a real-time clock are inserted into data packets and returned with the receiver's acknowledgements; having these information, the sender is able to calculate appropriate RTO estimations frequently. Special consideration should be paid to the receiver sending delayed ACKs, a hole in the sequence space due to lost segments, and a "filled" hole due to retransmits as outlined in [JBB92]. In contrast to GEO satellite systems, LEO constellations encounter delay-variations whose impact on TCP performance is currently an open issue [AGS99].

Larger Initial Congestion Window. Critical in terms of wasted capacity is the time spent in the initial slow start phase. Starting with an initial congestion window of one segment will result in a time-out at the receiver's side before sending an acknowledgement if the latter applies delayed ACKs. The standards-track document RFC 2581 [APS99] allows a TCP to use an initial cwnd of up to two segments. Taking into account that one third of traffic flows have between 100 and 1,000 bytes [FRC98], the transmission might be handled within one RTT of data exchange.

Further initial increase of the congestion window can still increase TCP's performance over high bandwidth-delay-product links without severe competition of background traffic flows. Nevertheless, the initial window as proposed in RFC 2414 [AFP98] is only "experimental" and not mature enough to be recommended for wide-spread use by the IETF.

TCP Extensions for Transactions. In a satellite environment, the three way handshake of standard TCP adds an extra RTT to the latency of a transaction. Especially for small and medium transaction this extra costs is detrimental to TCP's overall efficiency. Even though categorized as "experimental", RFC 1644 exploits a backwards-compatible option to TCP which would reduce this overhead and is considered for implementation in satellite TCP stacks.

NETWORK POLICIES

Even though for large propagation delays, end system policies (i.e., end-to-end congestion control) are the most important factors, network policies (i.e., buffer dimensioning and drop policies, and rate guarantees) should be considered as well in order to increase TCP's efficiency over high delay-bandwidth product connections [GGJ+98].

Buffer Size. Simulations show a asymptotic behavior of TCP's efficiency as a function of buffer size. TCP performs better with increasing buffers; buffer sizes greater then 0.5 times the bandwidth-delay-product are sufficient to reach 98% of the maximum throughput. This behavior proves to be independent of the number of sources when TCP/IP is run over UBR (unspecified bit rate) ATM connections applying per-VC buffer allocation with selective drop. [GJF+99].

Buffering Drop Policies. For long-delay satellite networks, drop policies have no significant effect in terms of fairness and efficiency of TCP connections. [GGJ+98].

When RTTs reach values comparable to WAN latencies (might occur for certain elevation angles of LEO satellites), a per-VC selective drop (SD) slightly surpassed early packet discard (EPD) and improves TCP performance. Its affect on TCP efficiency in a LEO satellite environment with high delay variation and possibly low RTT of 5 ms has still to be considered.

Rate Guarantees. High priority traffic may cause the starvation of TCP/IP connections over UBR. Equally granting a minimum rate to all TCP connections over a link raises TCP's efficiency by approx. 30% (efficiency values increase from 0.6 to 0.8 and 0.7 to 0.9 for LEO and GEO systems correspondingly). Rate guarantees for LEO satellite constellations show little, for GEO satellite constellations neglectable performance improvement [GJF+99]. Even though rate guarantees do not increase TCP performance when compared to end-system-policies, they assure a minimum flow of status information (RTT measurements etc.) between corresponding TCP entities and might be considered for implementation.

Summary / Design Parameters

End-system Policies mostly affect the efficiency of TCP/IP connections over large bandwidth-delay-product satellite connections. As a first step towards efficiency improvements of TCP, clients are assumed to be "satellite aware", i.e., stations are either directly connected to a satellite link or the LAN they are attached to is connected to the internet via a satellite link. This assumption allows the TCP stacks to be "manually tuned" to meet satellite needs. In a second phase, the tuning might become a matter of automation.

As a first approach, the following design parameters will be chosen:

TCP flavour	SACK TCP
initial congestion window	two segments
max. congestion window	1.5 MB
RTO (no RTT measurements)	500 ms (LEO)
	1000 ms (GEO)[a]
Buffer Sizes (Switches)	0.5 * BDP (accumulative)
Buffering Drop Policies	___ [b]
Rate Guarantees	

[a] Rather large RTOs should not hurt TCP's performance due to the aggressive behavior of TCP SACK.

[b] Due to their minor influence on TCP efficiency compared to TCP flavor, drop policies and rate guarantees are not mandatory for the demonstrator.

References

[AGS99] M. Allman, D. Glover, L. Sanchez. Enhancing TCP Over Satellite Channels using Standard Mechanisms. Internet Engineering Task Force, Network Working Group. IETF, RFC 2488, 1999.

[APS99] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. Internet Engineering Task Force, Network Working Group. IETF, RFC 2581, 1999.

[AFP98] M. Allman, S. Floyd, and C. Partridge. Increasing TCP's Initial Window. Internet Engineering Task Force, Network Working Group. IETF, RFC 2414, 1998.

[BBW+00a] H. Bischel, J. Bostic, M. Werner, K. Sood, F. Klefenz, A. Dreher, P. Todorova, M. Emmelmann, F. Krepel, T. Luckenbach, J. Tchouto, C. Tittel, H. Brandt, G. Eckhardt, and M. Trefz. ATM-Sat: ATM-Based Multmedia Communication via LEO-Satellites – System Architecture Report. Photocopied, 2000.

[BGI+97] K.B. Bhasin, D.R. Glover, W.D. Ivancic, and T.C. vonDeak. Enhancing End-toEnd Performance of Information Services Over Ka-Band Global Satellite Networks. NASA Scientific and Technical Information Program. NASA, TM_97-20629, 1997.

[FRC98] A. Feldmann, J. Rexford, and R. Cáceres. Efficient Policies for Carrying Web Traffic Over Flow-Switched Networks. In IEEE/ACM Transactions on Networking 6 (December 1998): 673-85, 1998.

[GGJ+98] M. Goyal, R. Goyal, R. Jain, B. Vandalore, and S. Fahmy. Performance Analysis of TCP Enhancments ofr WWW Traffic using UBR+ with Limited Buffers over Satellite Links. ATM Forum, Technical Committee – Traffic Management Working Group. ATM Forum, ATM_Forum/98-0876R1, 1998.

[GJK+99] R. Goyal, R. Jain, S. Kalyanaramon, S. Fahmy, B. Vandalore, and X. Cai, X. ATM Forum, Technical Working Group Members (AF-TM). ATM Forum, ATM_Forum/97-0424, 1999.

[GJR+99a] R. Goyal, R. Jain, S. Rahmy, B. Vandalore, S. Kalyanaraman. UBR Buffer Requirements for TCP/IP over Satellite Networks. ATM Forum, Technical Working Group Members (AF-TM). ATM Forum, ATM_Forum/97-0616, 1999.

[GJF+99] R. Goyal, R. Jain, S. Fahmy, B. Vandalore, and M. Goyal. Improving the Performance of TCP/IP over Satellite-ATM Networks. In Design Issues for Traffic Management for the ATM UBR+ Service for TCP Over Satellite Networks, ed. Rai Jain. NASA Scientific and Technical Information Program. Hanover: NASA Center for Aerospace Information, 1999.

[GKJ+99] R. Goyal, S. Kota, R. Jain, S. Fahmy, B. Vandalore, and J. Kallaus. Analysis and Simulation of Delay and Buffer Requirements of Satellite-ATM Networks for TCP/IP Traffic. In Design Issues for Traffic Management for the ATM UBR+ Service for TCP Over Satellite Networks, ed. Rai Jain. NASA Scientific and Technical Information Program. Hanover: NASA Center for Aerospace Information, 1999.

[HK99] T.R. Henderson and R.H. Katz. Transport Protocols for Internet-Compatible Satellite Networks. IEEE Journal on Selected Aereas in Communications 17 (February): 326-44, 1999.

[JBB92] V. Jacobson, R. Braden, and D. Borman. TCP Extensions for High Performance. Internet Engineering Task Force, Network Working Group. IETF, RFC 1323, 1992.

[MD90] J. Mogul and S. Deering. Path MTU Discovery. Internet Engineering Task Force, Network Working Group. IETF, RFC 1191, 1990.

[Ste98] W. Stevens. TCP/IP Illustrated. Vol. 1, The Protocols. Massachusetts: Addison-Wesley, 1998.