

A Survey of TCP Optimizations for Wireless Channels

Manfred Georg

mgeorg@arl.wustl.edu

Abstract

The use of wireless links causes several problems for TCP connections. Among them are high loss rates, high delay latency, high delay jitter, reverse path contention, dynamic link capacity, and link capacity dependent on transmission rate. These problems can be ameliorated through the use of modifications to TCP, with or without support from a router in the network. Solutions fall into two general categories: modifications which utilize an intermediate router with or without breaking the end-to-end semantics of TCP, and modifications which do not rely on any cooperation from routers. Solutions which utilize an intermediate router generally are more effective because they do not need to guess whether losses were caused by congestion or corruption. However such solutions are more costly and difficult to deploy and may break the end-to-end semantics of TCP. In general, modifications to TCP must be carefully crafted so that they remain fair to legacy TCP implementations.

Keywords

TCP, Wireless, Congestion Control, Loss Classifier, Split-TCP, SRP, WTCP, MTCP, SNOOP, Split TCP, Airmail, FEC, Satellite

Table of Contents

1. [Introduction](#)

1.1. [Basic Background on TCP](#)

1.2. [Wireless Channels](#)

1.3. [Roadmap](#)

2. [TCP Features](#)

3. Router Supported Solutions

3.1. Split Connection Protocols

3.2. Mobility

3.3. Snooping Protocols

3.4. Forward Error Correction

3.5. Link Level Retransmission

3.6. Summary of Protocols

4. Router Unaware Solutions

5. Practical Applications

5.1. Satellite Communication

5.2. Bluetooth

5.3. 802.11

6. Conclusion

7. References

8. Glossary

[table of contents](#)

1. Introduction

The last decade and a half have seen a massive explosion in networking technology. From its humble beginnings the Internet has emerged as a vital infrastructure servicing the communication demands of a new generation. The success of the Internet was built on

heterogeneity. Rather than replacing all the networks it grew out of, the Internet philosophy has been to combine these networks via gateways in such a way that communication can occur between separate networks. What this means in a practical sense is that new methods such as wireless communication, can easily be added onto the Internet; however, since it was not designed with such extensions in mind, communication may not be optimal while using these extensions. This paper will survey a number of optimizations to the Internet TCP protocol which attempt to improve performance over wireless channels.

[table of contents](#)

1.1. Basic Background on TCP

The cost of Internet infrastructure in place today and the lack of a central owner makes changes to the underlying architecture difficult and costly. On the other hand, the evolution of the Internet in terms of speed and versatility of the underlying network has changed dramatically over the years. Among these changes are the adoption of wireless links as a possible physical medium of communication. This causes a number of problems which were not adequately addressed in the original Internet design.

The original purpose of the Internet was to connect a number of existing networks together. These networks had their own communications protocols and special perks. However, when combining these networks together, it was necessary to create a new suite of network protocols which were usable on all networks. By necessity this suite of protocols could only require functionality that was present on all networks. The result was Internet Protocol (IP) and Transmission Control Protocol (TCP). The basic method of sending data over the Internet is through IP packets, which have just enough information in their headers to be routed to a destination. The TCP protocol is built on top of IP and uses signaling between the sender and receiver and retransmission of lost packets to ensure that all data is successfully transmitted. Additionally, TCP implements congestion control algorithms which ensure that the network will not be flooded with communications causing massive packet losses for prolonged periods of time. This latter functionality was only added later when it became apparent that it was needed.

[table of contents](#)

1.2. Wireless Channels

Communication over a wireless channel causes a number of problems which must be addressed in order to make efficient use of network bandwidth. The most important problems in wireless channels are their tendency to have high loss rates, high delay latency, high delay jitter, reverse path contention, dynamic link capacity, and link capacity dependent on transmission rate. Some of these problems are fundamental, while others only cause problems because of the current TCP/IP infrastructure. In general, the solutions presented have taken the form of variations to TCP with the possible inclusion of router support. However, the solution to some of these problems have caused other problems. This paper provides a summary of a number of methods to solve or ameliorate the effects of these problems.

[table of contents](#)

1.3. Roadmap

I first discuss the features of TCP and specific problems that arise because of the design of TCP. Then I present the general behavior of wireless channels and why they are challenging. I will give specific examples of bad interactions between TCP and wireless channels. Then I will provide a summary of the best standard TCP options for wireless communication. Finally, I will present more comprehensive solutions for effectively sending over wireless channels. In general, the solutions to this problem are relatively old and the only recent developments have been slight tweaks and practical applications of known methods. I will present the general approaches, and then proceed to more recent variations and fine tuning. Finally, I will conclude with a summary of the paper.

[table of contents](#)

2. TCP Features

TCP is an Internet standard protocol defined in [\[RFC 793\]](#). It is meant to reliably transfer a stream of data between two network end points. To do this, the sender maintains a buffer, called a window (or sliding window), of data that has been sent. A receiver acknowledges received data by sending feedback packets. When a sender receives an acknowledgment for data in its window, it can remove that data, since it has been successfully transmitted to the receiver. This is what is meant by the end-to-end semantics of TCP. The end hosts negotiate a protocol that reliably delivers data, even when the underlying network makes no such guarantees. In general, there is no need for intermediate routers to know that TCP data is traveling over their links. Recently, there is a growing trend for routers to be more aware of the type of data they are transmitting, thus allowing the possibility of better queue management schemes such as per-flow-queuing. Several of the proposed improvements to TCP over wireless channels leverage the participation of an intermediate router, usually the wireless access point.

The window based transmission procedure in TCP was designed so that a receiver can throttle the sender when it cannot process the data as fast as it is arriving. This is called flow control and is accomplished through the receiver window (rwnd) parameter in each ACK (acknowledgment) packet. This mechanism tells the sender how much buffering space is available at the receiver, so that the sender will never overfill the receiver's buffer window.

Originally there was no mechanism provided in TCP to detect the presence of congestion in the network. It was theoretically recognized that this could cause congestion collapse, or a situation in which the network is over utilized and dropping packets for a sustained period of time. In the mid to late 1980's the threat of widespread congestion collapse, triggered the last major addition to TCP, the inclusion of a congestion control mechanism [\[RFC 2581, RFC 2914\]](#). Since then a number of optimizations to this mechanism such as TCP NewReno [\[RFC 3782\]](#) and TCP SACK [\[RFC 2018, RFC 2883\]](#) have been proposed and standardized.

The main limitation of TCP's congestion control mechanism is its reliance on the loss of packets for detecting network congestion. Explicit Congestion Notification [\[RFC 3168\]](#) specifies the addition of notification bits in the IP header which are set when a router is congested so that the end hosts are able to slow the transmission rate before the router is forced to drop

packets. However, this feature is not generally available and therefore cannot be relied upon.

The timestamp option of TCP sends a timestamp field in every TCP packet. This can be used to calculate the Round Trip Time (RTT) very accurately, even when it is liable to change very quickly. The use of this option is generally recommended in wireless networks since, it allows the better computation of the Retransmission TimeOut (RTO) value.

[table of contents](#)

3. Router Supported Solutions

Although it would be preferable to place as much functionality as possible in the end hosts, leaving the network as simple as possible, this does not always lead to satisfactory solutions. It has become common practice to build some level of router support into modern wireless communications devices. This section discusses solutions that leverage router support. The basic solutions to the problem are the use of an intermediate router which intercepts packets and does something with them on behalf of the receiver.

For the most part the solutions center around the fact that standard TCP will misinterpret losses in the network due to corruption as due to congestion. Therefore, standard TCP will reduce its transmission rate without reason and underutilize the link. Table 1 shows the effects of Bit Error Rate (BER) on the Performance of a TCP Tahoe flow (Reproduced from [\[Elaarag02\]](#)).

Table 1: Effects of BER on TCP Performance

	BER = 10^{-5}	BER = 10^{-6}
Throughput (pkt/sec)	39.439	87.455
Success Probability	0.9892	0.999
Transfer time of 5000 pkts (sec)	123.847	58.032

[table of contents](#)

3.1. Split Connection Protocols

Indirect TCP (I-TCP) [\[Bakre95, Bakre97\]](#) simply terminates the TCP connection at the intermediate router and opens a new connection between the intermediate router and the wireless node. The hope is that the short connection over the wireless network will be able to recover from losses fast enough to keep up with the other TCP connection. While this works to some extent, it does not solve the underlying problems in transmitting TCP connections over wireless channels, it merely isolates the problem. Furthermore, this approach breaks the end-to-end semantics of TCP; meaning that although the sender has received an acknowledgment of the data, it has not been received at the end host but merely at the intermediate router. If something happens at the router, this can cause data to be permanently lost.

Selective Repeat Protocol (SRP) [Yavatkar94] is a similar protocol to I-TCP; however, it does not use a TCP connection on either end. Instead it uses a custom protocol which is better suited to transmission over a wireless channel.

Mobile TCP [Haas97] is a more comprehensive split TCP solution. It allows for asymmetric split connections and keeps the wireless receiver simple. The intermediate router uses both Forward Error Correction (FEC) and detects errors. Furthermore, header compression is used on the wireless portion of the connection.

[table of contents](#)

3.2. Mobility

The problem of mobility is frequently linked with wireless communication. Mobility means that having an established connection through a router does not imply that that router will be used indefinitely. Although, wireless communication can be between fixed end points such as a satellite uplink, it is more frequently associated with cell phones or laptops. The problem of mobility in IP networks is extensive and out of the scope of this paper; however, in the context of router assisted wireless communication it becomes important. This requires some amount of handover capability between routers which are assisting in wireless communication.

A common trick that can be done to TCP connections is to manipulate the receiver window (rwnd) parameter to deceive the sender in some fashion. [Brown97]. Normally this parameter tells the sender how much room the receiver has in its receive buffer. By setting this parameter to zero, the sender will assume there is no room in the buffer and stop transmitting data. At the same time, the sender will maintain the state of the connection exactly. This allows a connection to be frozen with minimal adverse effects when it is restarted. Under most implementations of TCP the connection can even resume sending at the same rate as it left off. This sort of temporary freezing of a connection can be very useful while a wireless node is temporarily out of range or while performing a handover of a connection.

However, being forced to freeze a connection during a handover can cause problems such as timeouts. A more compelling solution is to use path prediction to guess where a wireless node will go and to cache packets for that node at neighboring routers [Hadjieftymiades02]. This sort of a scheme is meant for cellular networks, where communication between neighboring cells is cheap and freezing a connection during a handover is undesirable.

[table of contents](#)

3.3. Snooping Protocols

In the original Snoop Protocol [Balakrishnan95] a router intercepts packets and keeps them in a buffer until the wireless node acknowledges them. If the wireless node does not receive the packet then the snooping router will receive three duplicate acknowledgments. It drops those duplicate acknowledgments and retransmits the lost packet.

WTCP [Ratnam98] is a similar protocol however, it further uses the timestamp option of TCP to trick the sender in such a way that retransmission timeouts do not occur. This protocol is very similar to Airmail, except that the latter pushes functionality into the base station to reduce the

processing load for the wireless node which is assumed to be a device with scarce resources [[Ayanoglu95](#)].

The delayed dupACKs protocol [[Vaidya99](#)] delays the returning duplicate acknowledgments instead of dropping them. Several attempts at retransmitting the lost packets from the router to the wireless node are made. The duplicate acknowledgments are not delayed long enough to cause a retransmission timeout which would be more costly than a fast recovery. Of course, the delayed acknowledgments are not sent at all if the retransmission of the lost packet is received and acknowledged within the time when the duplicate acknowledgments are being delayed.

[table of contents](#)

3.4. Forward Error Correction

Forward Error Correction (FEC) is the use of encoding methods which transmit redundant bits along with a message in such a way that if there is corruption along the path, the original message can still be recovered. The use of FEC allows such feats as the reading of scratched Compact Discs. It is a very powerful technique which is becoming widespread in wireless networks. All digital cellular networks make use of FEC in the transmission of voice communications. Unfortunately, the use of FEC comes at a price. The addition of the redundant bits can be significant, and can increase the size of a message substantially. For this reason, it is desirable to use the minimum amount of FEC required to reconstruct the message.

The use of FEC can be adjusted dynamically to account for variations in link conditions [[Tickoo05](#)]. This reduces the overhead from using too much FEC while maintaining an acceptable loss rate.

[table of contents](#)

3.5. Link Level Retransmission

A common way of reducing losses over wireless links is to buffer incoming packets at the wireless access point until they are received at the wireless node. In addition to FEC this is the standard way of reducing losses over 3G cellular networks [[3GPP99](#), [RLP00](#)]. However, a wireless link with a non-negligible delay can cause large variation in the RTT and spurious retransmission timeouts, which are very costly to TCP performance.

The problem with the standard way of computing RTO is that it assumes that the RTT varies in a Gaussian way and sets it equal to the RTT plus 3 standard deviations of the RTT. However, the assumption of a Gaussian distribution is blatantly untrue when factors such as link level retransmissions cause delays. The delay from a retransmission is either present or not, causing a bimodal or multimodal distribution. This can easily cause a timeout in the TCP connection which is worse than the three duplicate acknowledgments triggered by the loss of a packet. This becomes a major issue when the wireless hop in the connection has a substantial delay. For example, a link level retransmission causing a 1 ms delay of a packet is unlikely to cause problems, while a 20 ms delay would cause problems in a number of reasonable circumstances. Since the main problem is just the spacing of packets, an ACK regulator which

smoothens the delay jitter can be used to dramatically improve the performance of TCP [Chan04]. This can be implemented at the wireless access router and requires very little state maintenance.

[table of contents](#)

3.6. Summary of Protocols

Table 2 provides a summary of a number of protocols, it is reproduced from [Elaarag02].

Table 2: Summary of Protocol Attributes

	RLP	Airmail	Snoop	FR	EBSN	MTCP	I-TCP	M-TCP
High BER	Yes	Yes	Yes		Yes	Yes	Yes	
Bursty Error		Yes	Yes		Yes	Yes		
Handoff		Yes	Yes	Yes			Yes	Yes
Long Disconnections							Yes	Yes
Frequent Disconnections							Yes	Yes
Bandwidth	Yes	Yes						Yes
Cell Size				Yes				
Power Scarcity		Yes			Yes			Yes
Serial Timeouts					Yes	Yes	Yes	Yes
Packet Size Variation					Yes	Yes		
End-to-end TCP semantics	Yes	Yes	Yes	Yes	Yes			Yes
Compatibility	Yes	Yes	Yes			Yes	Yes	Yes

[table of contents](#)

4. Router Unaware Solutions

It is sometimes infeasible to include router support since it requires modifications to existing routers. Furthermore, a solution that does not include routers would be cheaper and easier to deploy. The degradation to performance for TCP over a wireless link in this case is mainly from the misinterpretation of corruption losses on the wireless link as caused by congestion. This causes the TCP connection to scale back its transmission rate when it should not. Therefore, TCP over wireless without router support basically hinges on being able to classify losses that occur due to corruption from losses that occur due to congestion. This is in stark contrast to router based solutions which receive this information explicitly.

The early work done in this area is nicely summarized in [Cen03]. Of major importance is the observation that a misclassification rate of 18% will allow a connection to remain TCP fair. Although some classifiers do remarkably well under certain circumstances, there are other circumstances under which they do poorly. There is no classifier that performs predictably well under all conditions. However, picking classifiers and learning what classifiers do well when is a standard machine learning type problem which has been studied extensively. This work can be leveraged to produce much better classifiers [Khayat04, Khayat05]. However, the use of the better machine learned classifiers causes its own problems. Namely, calculating the classifiers is now a computationally expensive operation. This operation must be performed either at the receiver, which might be a wireless embedded system with scarce resources, or it must be performed at the server, causing scalability problems.

[table of contents](#)

5. Practical Applications

The solutions described thus far have been general techniques and recommendations that are suited for all kinds of wireless communications. When they have been particular to a certain kind of wireless network it has been to cellular networks and 3G technology specifically. However, there are unique problems in Local Area Network (LAN) communications such as 802.11 and Bluetooth and satellite communication. This section will address these situations specifically. [table of contents](#)

5.1. Satellite Communication

Satellite communication is problematic for different reasons than terrestrial wireless communication. The end points of satellite links are tightly controlled with respect to antenna direction and encoding. The use of Forward Error Correction (FEC) codes fine tuned to the actual link further allow for a very reliable channel. Therefore, modern satellite connections suffer a bit error rate (BER) on the order of 1 per 10 million [RFC 2488]. Although in some frequencies one must worry about the signal being attenuated by rain, under normal conditions loss rates are very low.

The intrinsic propagation delay required to send signals to and from a satellite are formidable, causing a high bandwidth-delay product. Most research in high bandwidth-delay product networks has focused on private networks or transatlantic links, however, the use of a geostationary satellite link in a connection will add at least 558 ms to its Round Trip Time (RTT) [RFC 2488].

Scalable TCP [Kelly03] uses a Multiplicative Increase Multiplicative Decrease (MIMD) algorithm when transmitting at high speeds instead of the traditional Additive Increase Multiplicative Decrease (AIMD) algorithm which TCP uses. However, at low speeds Scalable TCP acts exactly as regular TCP does and hence remains TCP fair as long as regular TCP would be able to utilize the network resources. The use of an MIMD algorithm provides much better capacity scalability, however, this is achieved at the cost of inter-flow fairness. Two competing Scalable TCP flows transmitting at high rates in MIMD mode, will not converge to a fair rate under assumptions of synchronous feedback. It has yet to be shown whether they converge in practice, especially under low level of statistical multiplexing. A number of similar

proposals are circulating, with no clearly superior design.

High Speed TCP uses a similar approach of adjusting the scaling factor depending on the transmission rate [RFC3649]. However, instead of using an MIMD scheme directly it uses a sequence of AIMD schemes each with a greater additive increase factor. A table lookup on the current congestion window size is used to find the appropriate algorithm. Essentially this creates a discretized MIMD algorithm and hence the differences between this and Scalable-TCP are small.

There are a number of minor recommendations for transmitting over satellite links. The discovery of the Maximum Transmittable Unit (MTU) is vital, as satellite links may not have standard values for this quantity. The use of Forward Error Correction (FEC) on the link level is very useful for lowering the user seen Bit Error Rate (BER). As in all high bandwidth-delay product networks TCP SACK can increase performance dramatically.

[table of contents](#)

5.2. Bluetooth

Recently a number of standard solutions have been applied to specific technologies. This shows the maturation of the subject matter from a purely theoretical endeavor to a practical one. Bluetooth is a personal area wireless network technology. Chen et al. have analyzed the various Forward Error Correction (FEC) methods and packet length for optimizing TCP performance over Bluetooth connections [Chen04].

[table of contents](#)

5.3. 802.11

TCP makes very general assumptions about the networks it is transmitting over. Among these assumptions is the assumption that the forward and reverse paths are independent and do not interfere. Obviously 802.11 violates this assumption since it is a broadcast network in which both the forward and reverse data is sent over the same channel. In an attempt at fairness 802.11 splits resources evenly between senders and receivers. Unfortunately, this interacts very poorly with TCP's own fairness mechanisms causing the possibility of massive unfairness in 802.11 networks [Pilosof03]. To solve this, we can use the receiver window (rwnd) parameter of the TCP connection to throttle the sending rate.

[table of contents](#)

6. Conclusion

The use of wireless links causes several problems for TCP connections. Among them are high loss rates, high delay latency, high delay jitter, reverse path contention, dynamic link capacity, and link capacity dependent on transmission rate. A number of modifications to TCP have been proposed to improve its performance over a wireless channel. These solutions fall into two general categories: those that utilize an intermediate router and those that rely only on the

end hosts. Router assisted solutions are generally better since they do not need to guess whether a loss is due to congestion or corruption. However care must be taken to not break the end-to-end semantics of TCP. While some modifications to TCP are acceptable, it should be noted that such modifications must remain fair to legacy TCP connections which do not use the modifications.

[table of contents](#)

7. References

References are ordered with more relevant sources listed first.

[Elaarag02] Hala Elaarag, "Improving TCP performance over mobile networks," ACM Computing Surveys, 2002.

http://portal.acm.org/ft_gateway.cfm?id=568524&type=pdf&coll=ACM&dl=ACM&CFID=74421280&CFTOKEN=52319095

[RFC 2914] S. Floyd, "RFC 2914: Congestion Control Principles," September 2000

<http://www.ietf.org/rfc/rfc2914.txt>

[Bakre97] Ajay V. Bakre and B. R. Badrinath, "Implementation and Performance Evaluation of Indirect TCP," IEEE Transactions on Computers.

<http://citeseer.ist.psu.edu/rd/23081651%2C416569%2C1%2C0.25%2CDownload/http://coblitz.codeen.org:3125/citeseer.ist.psu.edu/cache/papers/cs/20580/http:zS>

[Yavatkar94] R. Yavatkar and N. Bhagwat, "Improving end-to-end performance of TCP over mobile Internetworks," IEEE Workshop on Mobile Computing, 1994.

[Cen03] S. Cen, P. Cosmanm, G. Voelker, "End-to-end differentiation of congestion and wireless Losses," IEEE/ACM Transactions on Networking, 2003.

<http://www.cs.ucsd.edu/~voelker/pubs/lda-ton03.pdf>

[Haas97] Zygmunt J. Haas, "Mobile-TCP: An Asymmetric Transport Protocol Design For Mobile Systems," Mobile Multimedia Communications, 1997.

<http://citeseer.ist.psu.edu/rd/23081651%2C232076%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/10247/http:zSzzSzwww.ee.cornell.edu7EhaaszSzPublicationszSzmmc.pdf/haas97mobiletcp.pdf>

[RFC 2488] M. Allman, D. Glover, L. Sanchez, "RFC 2488: Enhancing TCP Over Satellite Channels using Standard Mechanisms," January 1999.

<http://www.ietf.org/rfc/rfc2488.txt>

[Kelly03] Tom Kelly, "Scalable TCP: improving performance in highspeed wide area networks," SIGCOMM 2003.

http://portal.acm.org/ft_gateway.cfm?id=956989&type=pdf&coll=GUIDE&dl=GUIDE&CFID=73584078&CFTOKEN=22809677

[RFC 3649] S. Floyd "RFC 3649: HighSpeed TCP for Large Congestion Windows," December

2003.

<http://www.ietf.org/rfc/rfc3649.txt>

[Brown97] Kevin Brown and Suresh Singh, "M-TCP: TCP for Mobile Cellular Networks," ACM Computer Communication Review, 1997.

<http://citeseer.ist.psu.edu/rd/23081651%2C100145%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/5856/ftp:zSzzSzftp.ece.orst.eduzSzpul>

[Khayat05] Ibtissam El Khayat, Pierre Geurts, and Guy Leduc, "Improving TCP in Wireless Networks with an Adaptive Machine-Learnt Classifier of Packet Loss Causes," IFIP International Networking Conference, 2005.

<ftp://ftp.run.montefiore.ulg.ac.be/pub/RUN-PP05-03.pdf>

[Tickoo05] Omesh Tickoo, Vijaynarayanan Subramanian, Shivkumar Kalyanaraman and K. K. Ramakrishnan, "LT-TCP: End-to-End Framework to Improve TCP Performance over Networks with Lossy Channels," IEEE International Workshop on Quality of Service, 2005.

<http://www.ecse.rpi.edu/Homepages/shivkuma/research/papers/tcp-fec2005.pdf>

[Hadjiefthymiades02] Stathes Hadjiefthymiades, Stamatis Papayiannis, and Lazaros Merakos, "Using Path Prediction to Improve TCP Performance in Wireless/Mobile Communications," IEEE Communications Magazine, 2002.

<http://ieeexplore.ieee.org/iel5/35/22021/01024415.pdf?isnumber=&arnumber=1024415>

[Balakrishnan97] Hari Balakrishnan, Venkat Padmanabhan, Srinivasan Seshan, and Randy H. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," IEEE/ACM Transactions on Networking, 1997.

<http://nms.lcs.mit.edu/~hari/papers/ton.ps>

[Chan04] Mun Choon Chan and Ramachandran Ramjee, "Improving TCP/IP Performance over Third Generation Wireless Networks," INFOCOM, 2004.

http://www.ieee-infocom.org/2004/Papers/39_3.PDF

[Ratnam98] Karunaharan Ratnam and Ibrahim Matta, "WTCP: An Efficient Mechanism for Improving TCP Performance over Wireless Links," IEEE Symposium on Computers and Communications (ISCC), 1998.

<http://citeseer.ist.psu.edu/rd/23081651%2C314964%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/14145/http:zSzzSzwww.ece.neu.eduz>

[RFC 2018] M. Mathis, J. Mahdavi, S. Floyd, A. Romanow, "RFC 2018: TCP Selective Acknowledgment Options," October 1996

<http://www.ietf.org/rfc/rfc2018.txt>

[RFC 3168] K. Ramakrishnan, S. Floyd, D. Black, "RFC 3168: The Addition of Explicit Congestion Notification (ECN) to IP," September 2001

<http://www.ietf.org/rfc/rfc3168.txt>

[RFC 3782] S. Floyd, T. Henderson, A. Gurtov, "RFC 3782: The NewReno Modification to TCP's Fast Recovery Algorithm," April 2004

<http://www.ietf.org/rfc/rfc3782.txt>

[RFC 2883] S. Floyd, J. Mahdavi, M. Mathis, M. Podolsky "RFC 2883: An Extension to the Selective Acknowledgement (SACK) Option for TCP," July 2000

<http://www.ietf.org/rfc/rfc2883.txt>

[RFC 2581] M. Allman, V. Paxson, W. Stevens, "RFC 2581: TCP Congestion Control," April 1999

<http://www.ietf.org/rfc/rfc2581.txt>

[RFC 793] "RFC 793: Transmission Control Protocol," September 1981

<http://www.ietf.org/rfc/rfc793.txt>

[Bakre95] Ajay Bakre and B.R. Badrinath, "Handoff and System Support for Indirect TCP/IP," Proceedings of the 2nd USENIX Symposium on Mobile and Location-Independent Computing, 1995.

<http://citeseer.ist.psu.edu/rd/23081651%2C33194%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/884/ftp:zSzzSzpaul.rutgers.eduzSzput>

[Chen04] Ling-Jyh Chen, Rohit Kapoor, M. Y. Sanadidi, Mario Gerla, "Enhancing Bluetooth TCP Throughput via Link Layer Packet Adaptation," ICC, 2004.

<http://citeseer.ist.psu.edu/rd/23081651%2C721551%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs2/587/http:zSzzSzwww.cs.ucla.eduzSz>

[Balakrishnan95] Hari Balakrishnan, Srinivasan Seshan, and Randy H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," ACM Wireless Networks, 1995.

<http://nms.lcs.mit.edu/~hari/papers/ton.ps>

[Ayanoglu95] Ender Ayanoglu, Sanjoy Paul, Thomas F. LaPorta, Krishan K. Sabnani, and Richard D. Gitlin, "AIRMAIL: A link-layer protocol for wireless networks," ACM Wireless Networks 1995.

<http://citeseer.ist.psu.edu/rd/23081651%2C137704%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/2897/http:zSzzSzremus.rutgers.eduzS>

[7EpsanjoyzSzairmail.pdf/ayanoglu95airmail.pdf](http://citeseer.ist.psu.edu/cache/papers/cs/2897/http:zSzzSzremus.rutgers.eduzS7EpsanjoyzSzairmail.pdf/ayanoglu95airmail.pdf)

[Vaidya99] Nitin Vaidya, Miten Mehta, Charles Perkins, and Gabriel Montenegro, "Delayed Duplicate Acknowledgements: A TCP-Unaware Approach to Improve Performance of TCP over Wireless," Technical Report, 1999.

<http://citeseer.ist.psu.edu/rd/0%2C101437%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/3483/http:zSzzSzwww.cs.tamu.eduzS>

[-computingzSz99-003.pdf/vaidya99delayed.pdf](http://citeseer.ist.psu.edu/cache/papers/cs/3483/http:zSzzSzwww.cs.tamu.eduzS-computingzSz99-003.pdf/vaidya99delayed.pdf)

[Khayat04] Ibtissam El Khayat, Pierre Geurts, and Guy Leduc, "Enhancement of TCP over Wired/Wireless Networks with Packet Loss Classifiers Inferred by Supervised Learning," Technical Report, 2004.

<ftp://ftp.run.montefiore.ulg.ac.be/pub/RUN-PP04-06.pdf>

[Pilosof03] "Saar Pilosof, Ramachandran Ramjee, Danny Raz, Yuval Shavitt, and Prasun Sinha, "Understanding TCP fairness over Wireless LAN," INFOCOM, 2003.

http://www.ieee-infocom.org/2003/papers/21_04.PDF

[3GPP99] Third Generation Partnership Project, "RLC Protocol Specification (3G TS 25.322:)", 1999.

<http://www.3gpp.org/ftp/Specs/html-info/25322.htm>

[RLP00] TIA/EIA/IS-707-A-2.10 "Data Service Options for Spread Spectrum Systems: Radio Link Protocol Type 3" http://www.3gpp2.org/Public_html/specs/C.S0017-010-A_v2.0_050907.pdf

[table of contents](#)

8. Glossary

ACK - Acknowledgment packet

AIMD - Additive Increase Multiplicative Decrease, a distributed resource allocation scheme which is known for its fairness.

BER - Bit Error Rate

Congestion Control - The process of scaling back the transmission rate to avoid excessive traffic in the network.

cwnd - Congestion Window

ECN - Explicit Congestion Notification

Fast Recovery - The practice of retransmitting a packet immediately upon receiving a partial acknowledgment during a congestion event. Also called the NewReno Modification to TCP.

Fast Retransmit - Retransmitting a packet when 3 duplicate Acknowledgments are received (instead of waiting for a timeout) also called the Reno modification to TCP.

FEC - Forward Error Correction

IETF - Internet Engineering Task Force

IP - Internet Protocol

MIMD - Multiplicative Increase Multiplicative Decrease, a distributed resource allocation scheme which is known for its scalability.

NACK - Negative Acknowledgment

NewReno Modification - See Fast Recovery

Reno Modification - See Fast Retransmit

RFC - Request For Comments, the IETF standardization documents.

RTO - Retransmission Timeout, the time to wait for a packet before triggering a timeout.

RTT - Round Trip Time (the delay between sending a packet and receiving its acknowledgment)

rwnd - Receiver Window

SACK - Selective Acknowledgment

SRP - Selective Repeat Protocol

ssthresh - Slow Start Threshold

TCP - Transmission Control Protocol

[table of contents](#)

Last Modified: Tue May 2 00:22:31 CDT 2006
