

Network Working Group
Request for Comments: 4614
Category: Informational

M. Duke
Boeing Phantom Works
R. Braden
USC Information Sciences Institute
W. Eddy
Verizon Federal Network Systems
E. Blanton
Purdue University Computer Science
September 2006

A Roadmap for Transmission Control Protocol (TCP) Specification Documents

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document contains a "roadmap" to the Requests for Comments (RFC) documents relating to the Internet's Transmission Control Protocol (TCP). This roadmap provides a brief summary of the documents defining TCP and various TCP extensions that have accumulated in the RFC series. This serves as a guide and quick reference for both TCP implementers and other parties who desire information contained in the TCP-related RFCs.

Table of Contents

1.	Introduction	2
2.	Basic Functionality	4
3.	Recommended Enhancements	6
3.1.	Congestion Control and Loss Recovery Extensions	7
3.2.	SACK-Based Loss Recovery and Congestion Control	8
3.3.	Dealing with Forged Segments	9
4.	Experimental Extensions	10
5.	Historic Extensions	13
6.	Support Documents	14
6.1.	Foundational Works	15
6.2.	Difficult Network Environments	16
6.3.	Implementation Advice	19
6.4.	Management Information Bases	20
6.5.	Tools and Tutorials	22
6.6.	Case Studies	22
7.	Undocumented TCP Features	23
8.	Security Considerations	24
9.	Acknowledgments	24
10.	Informative References	25
10.1.	Basic Functionality	25
10.2.	Recommended Enhancements	25
10.3.	Experimental Extensions	26
10.4.	Historic Extensions	27
10.5.	Support Documents	28
10.6.	Informative References Outside the RFC Series	31

[1.](#) Introduction

A correct and efficient implementation of the Transmission Control Protocol (TCP) is a critical part of the software of most Internet hosts. As TCP has evolved over the years, many distinct documents have become part of the accepted standard for TCP. At the same time, a large number of more experimental modifications to TCP have also been published in the RFC series, along with informational notes, case studies, and other advice.

As an introduction to newcomers and an attempt to organize the plethora of information for old hands, this document contains a "roadmap" to the TCP-related RFCs. It provides a brief summary of the RFC documents that define TCP. This should provide guidance to implementers on the relevance and significance of the standards-track extensions, informational notes, and best current practices that relate to TCP.

This document is not an update of [RFC 1122](#) and is not a rigorous standard for what needs to be implemented in TCP. This document is merely an informational roadmap that captures, organizes, and summarizes most of the RFC documents that a TCP implementer, experimenter, or student should be aware of. Particular comments or broad categorizations that this document makes about individual mechanisms and behaviors are not to be taken as definitive, nor should the content of this document alone influence implementation decisions.

This roadmap includes a brief description of the contents of each TCP-related RFC. In some cases, we simply supply the abstract or a key summary sentence from the text as a terse description. In addition, a letter code after an RFC number indicates its category in the RFC series (see [BCP 9](#) [[RFC2026](#)] for explanation of these categories):

- S - Standards Track (Proposed Standard, Draft Standard, or Standard)
- E - Experimental
- B - Best Current Practice
- I - Informational

Note that the category of an RFC does not necessarily reflect its current relevance. For instance, [RFC 2581](#) is nearly universally deployed although it is only a Proposed Standard. Similarly, some Informational RFCs contain significant technical proposals for changing TCP.

This roadmap is divided into four main sections. [Section 2](#) lists the RFCs that describe absolutely required TCP behaviors for proper functioning and interoperability. Further RFCs that describe strongly encouraged, but non-essential, behaviors are listed in [Section 3](#). Experimental extensions that are not yet standard practices, but that potentially could be in the future, are described in [Section 4](#).

The reader will probably notice that these three sections are broadly equivalent to MUST/SHOULD/MAY specifications (per [RFC 2119](#)), and although the authors support this intuition, this document is merely descriptive; it does not represent a binding standards-track position. Individual implementers still need to examine the standards documents themselves to evaluate specific requirement levels.

A small number of older experimental extensions that have not been widely implemented, deployed, and used are noted in [Section 5](#). Many other supporting documents that are relevant to the development, implementation, and deployment of TCP are described in [Section 6](#). Within each section, RFCs are listed in the chronological order of their publication dates.

A small number of fairly ubiquitous important implementation practices that are not currently documented in the RFC series are listed in [Section 7](#).

[2. Basic Functionality](#)

A small number of documents compose the core specification of TCP. These define the required basic functionalities of TCP's header parsing, state machine, congestion control, and retransmission timeout computation. These base specifications must be correctly followed for interoperability.

[RFC 793](#) S: "Transmission Control Protocol", STD 7 (September 1981)

This is the fundamental TCP specification document [[RFC0793](#)]. Written by Jon Postel as part of the Internet protocol suite's core, it describes the TCP packet format, the TCP state machine and event processing, and TCP's semantics for data transmission, reliability, flow control, multiplexing, and acknowledgment.

[Section 3.6 of RFC 793](#), describing TCP's handling of the IP precedence and security compartment, is mostly irrelevant today. [RFC 2873](#) changed the IP precedence handling, and the security compartment portion of the API is no longer implemented or used. In addition, [RFC 793](#) did not describe any congestion control mechanism. Otherwise, however, the majority of this document still accurately describes modern TCPs. [RFC 793](#) is the last of a series of developmental TCP specifications, starting in the Internet Experimental Notes (IENs) and continuing in the RFC series.

[RFC 1122](#) S: "Requirements for Internet Hosts - Communication Layers" (October 1989)

This document [[RFC1122](#)] updates and clarifies [RFC 793](#), fixing some specification bugs and oversights. It also explains some features such as keep-alives and Karn's and Jacobson's RTO estimation algorithms [[KP87](#)][[Jac88](#)][[JK92](#)]. ICMP interactions are mentioned, and some tips are given for efficient implementation. [RFC 1122](#) is an Applicability Statement, listing the various features that MUST, SHOULD, MAY, SHOULD NOT, and MUST NOT be present in

standards-conforming TCP implementations. Unlike a purely informational "roadmap", this Applicability Statement is a standards document and gives formal rules for implementation.

[RFC 2460](#) S: "Internet Protocol, Version 6 (IPv6) Specification (December 1998)

This document [[RFC2460](#)] is of relevance to TCP because it defines how the pseudo-header for TCP's checksum computation is derived when 128-bit IPv6 addresses are used instead of 32-bit IPv4 addresses. Additionally, [RFC 2675](#) describes TCP changes required to support IPv6 jumbograms.

[RFC 2581](#) S: "TCP Congestion Control" (April 1999)

Although [RFC 793](#) did not contain any congestion control mechanisms, today congestion control is a required component of TCP implementations. This document [[RFC2581](#)] defines the current versions of Van Jacobson's congestion avoidance and control mechanisms for TCP, based on his 1988 SIGCOMM paper [[Jac88](#)]. [RFC 2001](#) was a conceptual precursor that was obsoleted by [RFC 2581](#).

A number of behaviors that together constitute what the community refers to as "Reno TCP" are described in [RFC 2581](#). The name "Reno" comes from the Net/2 release of the 4.3 BSD operating system. This is generally regarded as the least common denominator among TCP flavors currently found running on Internet hosts. Reno TCP includes the congestion control features of slow start, congestion avoidance, fast retransmit, and fast recovery.

[RFC 1122](#) mandates the implementation of a congestion control mechanism, and [RFC 2581](#) details the currently accepted mechanism. [RFC 2581](#) differs slightly from the other documents listed in this section, as it does not affect the ability of two TCP endpoints to communicate; however, congestion control remains a critical component of any widely deployed TCP implementation and is required for the avoidance of congestion collapse and to ensure fairness among competing flows.

[RFC 2873](#) S: "TCP Processing of the IPv4 Precedence Field" (June 2000)

This document [[RFC2873](#)] removes from the TCP specification all processing of the precedence bits of the TOS byte of the IP header. This resolves a conflict over the use of these bits between [RFC 793](#) and Differentiated Services [[RFC2474](#)].

[RFC 2988](#) S: "Computing TCP's Retransmission Timer" (November 2000)

Abstract: "This document defines the standard algorithm that Transmission Control Protocol (TCP) senders are required to use to compute and manage their retransmission timer. It expands on the discussion in [section 4.2.3.1 of RFC 1122](#) and upgrades the requirement of supporting the algorithm from a SHOULD to a MUST." [[RFC2988](#)]

3. Recommended Enhancements

This section describes recommended TCP modifications that improve performance and security. RFCs 1323 and 3168 represent fundamental changes to the protocol. [RFC 1323](#), based on RFCs 1072 and 1185, allows better utilization of high bandwidth-delay product paths by providing some needed mechanisms for high-rate transfers. [RFC 3168](#) describes a change to the Internet's architecture, whereby routers signal end-hosts of growing congestion levels and can do so before packet losses are forced. [Section 3.1](#) lists improvements in the congestion control and loss recovery mechanisms specified in [RFC 2581](#). [Section 3.2](#) describes further refinements that make use of selective acknowledgments. [Section 3.3](#) deals with the problem of preventing forged segments.

[RFC 1323](#) S: "TCP Extensions for High Performance" (May 1992)

This document [[RFC1323](#)] defines TCP extensions for window scaling, timestamps, and protection against wrapped sequence numbers, for efficient and safe operation over paths with large bandwidth-delay products. These extensions are commonly found in currently used systems; however, they may require manual tuning and configuration. One issue in this specification that is still under discussion concerns a modification to the algorithm for estimating the mean RTT when timestamps are used.

[RFC 2675](#) S: "IPv6 Jumbograms" (August 1999)

IPv6 supports longer datagrams than were allowed in IPv4. These are known as Jumbograms, and use with TCP has necessitated changes to the handling of TCP's MSS and Urgent fields (both 16 bits). This document [[RFC2675](#)] explains those changes. Although it describes changes to basic header semantics, these changes should only affect the use of very large segments, such as IPv6 jumbograms, which are currently rarely used in the general Internet. Supporting the behavior described in this document does not affect interoperability with other TCP implementations when IPv4 or non-jumbogram IPv6 is used. This document states that jumbograms are to only be used when it can be guaranteed that all

receiving nodes, including each router in the end-to-end path, will support jumbograms. If even a single node that does not support jumbograms is attached to a local network, then no host on that network may use jumbograms. This explains why jumbogram use has been rare, and why this document is considered a performance optimization and not part of TCP over IPv6's basic functionality.

[RFC 3168](#) S: "The Addition of Explicit Congestion Notification (ECN) to IP" (September 2001)

This document [[RFC3168](#)] defines a means for end hosts to detect congestion before congested routers are forced to discard packets. Although congestion notification takes place at the IP level, ECN requires support at the transport level (e.g., in TCP) to echo the bits and adapt the sending rate. This document updates [RFC 793](#) to define two previously unused flag bits in the TCP header for ECN support. [RFC 3540](#) provides a supplementary (experimental) means for more secure use of ECN, and [RFC 2884](#) provides some sample results from using ECN.

[3.1.](#) Congestion Control and Loss Recovery Extensions

Two of the most important aspects of TCP are its congestion control and loss recovery features. TCP traditionally treats lost packets as indicating congestion-related loss, and cannot distinguish between congestion-related loss and loss due to transmission errors. Even when ECN is in use, there is a rather intimate coupling between congestion control and loss recovery mechanisms. There are several extensions to both features, and more often than not, a particular extension applies to both. In this sub-section, we group enhancements to either congestion control, loss recovery, or both, which can be performed unilaterally; that is, without negotiating support between endpoints. In the next sub-section, we group the extensions that specify or rely on the SACK option, which must be negotiated bilaterally. TCP implementations should include the enhancements from both sub-sections so that TCP senders can perform well without regard to the feature sets of other hosts they connect to. For example, if SACK use is not successfully negotiated, a host should use the NewReno behavior as a fall back.

[RFC 3042](#) S: "Enhancing TCP's Loss Recovery Using Limited Transmit" (January 2001)

Abstract: "This document proposes Limited Transmit, a new Transmission Control Protocol (TCP) mechanism that can be used to more effectively recover lost segments when a connection's congestion window is small, or when a large number of segments are lost in a single transmission window." [[RFC3042](#)] Tests from 2004 showed that Limited Transmit was deployed in roughly one third of the web servers tested [[MAF04](#)].

[RFC 3390](#) S: "Increasing TCP's Initial Window" (October 2002)

This document [[RFC3390](#)] updates [RFC 2581](#) to permit an initial TCP window of three or four segments during the slow-start phase, depending on the segment size.

[RFC 3782](#) S: "The NewReno Modification to TCP's Fast Recovery Algorithm" (April 2004)

This document [[RFC3782](#)] specifies a modification to the standard Reno fast recovery algorithm, whereby a TCP sender can use partial acknowledgments to make inferences determining the next segment to send in situations where SACK would be helpful but isn't available. Although it is only a slight modification, the NewReno behavior can make a significant difference in performance when multiple segments are lost from a single window of data.

[3.2.](#) SACK-Based Loss Recovery and Congestion Control

The base TCP specification in [RFC 793](#) provided only a simple cumulative acknowledgment mechanism. However, a selective acknowledgment (SACK) mechanism provides performance improvement in the presence of multiple packet losses from the same flight, more than outweighing the modest increase in complexity. A TCP should be expected to implement SACK; however, SACK is a negotiated option and is only used if support is advertised by both sides of a connection.

[RFC 2018](#) S: "TCP Selective Acknowledgment Options" (October 1996)

This document [[RFC2018](#)] defines the basic selective acknowledgment (SACK) mechanism for TCP.

[RFC 2883](#) S: "An Extension to the Selective Acknowledgement (SACK) Option for TCP" (July 2000)

This document [[RFC2883](#)] extends [RFC 2018](#) to cover the case of acknowledging duplicate segments.

[RFC 3517](#) S: "A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP" (April 2003)

This document [[RFC3517](#)] describes a relatively sophisticated algorithm that a TCP sender can use for loss recovery when SACK reports more than one segment lost from a single flight of data. Although support for the exchange of SACK information is widely implemented, not all implementations use an algorithm as sophisticated as that described in [RFC 3517](#).

[3.3](#). Dealing with Forged Segments

By default, TCP lacks any cryptographic structures to differentiate legitimate segments and those spoofed from malicious hosts. Spoofing valid segments requires correctly guessing a number of fields. The documents in this sub-section describe ways to make that guessing harder, or to prevent it from being able to affect a connection negatively.

The TCPM working group is currently in progress towards fully understanding and defining mechanisms for preventing spoofing attacks (including both spoofed TCP segments and ICMP datagrams). Some of the solutions being considered rely on TCP modifications, whereas others rely on security at lower layers (like IPsec) for protection.

[RFC 1948](#) I: "Defending Against Sequence Number Attacks" (May 1996)

This document [[RFC1948](#)] describes the TCP vulnerability that allows an attacker to send forged TCP packets, by guessing the initial sequence number in the three-way handshake. Simple defenses against exploitation are then described. Some variation is implemented in most currently used operating systems.

[RFC 2385](#) S: "Protection of BGP Sessions via the TCP MD5 Signature Option" (August 1998)

From document: "This document describes current existing practice for securing BGP against certain simple attacks. It is understood to have security weaknesses against concerted attacks.

This memo describes a TCP extension to enhance security for BGP. It defines a new TCP option for carrying an MD5 digest in a TCP segment. This digest acts like a signature for that segment, incorporating information known only to the connection end points. Since BGP uses TCP as its transport, using this option in the way described in this paper significantly reduces the danger from certain security attacks on BGP." [[RFC2385](#)]

TCP MD5 options are currently only used in very limited contexts, primarily for defending BGP exchanges between routers. Some deployment notes for those using TCP MD5 are found in the later [RFC 3562](#), "Key Management Considerations for the TCP MD5 Signature Option" [[RFC3562](#)]. [RFC 4278](#) deprecates the use of TCP MD5 outside BGP [[RFC4278](#)].

4. Experimental Extensions

The RFCs in this section are still experimental, but they may become proposed standards in the future. At least part of the reason that they are still experimental is to gain more wide-scale experience with them before a standards track decision is made. By their publication as experimental RFCs, it is hoped that the community of TCP researchers will analyze and test the contents of these RFCs. Although experimentation is encouraged, there is not yet formal consensus that these are fully logical and safe behaviors. Wide-scale deployment of implementations that use these features should be well thought-out in terms of consequences.

[RFC 2140](#) I: "TCP Control Block Interdependence" (April 1997)

This document [[RFC2140](#)] suggests how TCP connections between the same endpoints might share information, such as their congestion control state. To some degree, this is done in practice by a few operating systems; for example, Linux currently has a destination cache. Although this RFC is technically informational, the concepts it describes are in experimental use, so we include it in this section.

A related proposal, the Congestion Manager, is specified in [RFC 3124](#) [[RFC3124](#)]. The idea behind the Congestion Manager, moving congestion control outside of individual TCP connections, represents a modification to the core of TCP, which supports sharing information among TCP connections as well. Although a Proposed Standard, some pieces of the Congestion Manager support architecture have not been specified yet, and it has not achieved use or implementation beyond experimental stacks, so it is not listed among the standard TCP enhancements in this roadmap.

[RFC 2861](#) E: "TCP Congestion Window Validation" (June 2000)

This document [[RFC2861](#)] suggests reducing the congestion window over time when no packets are flowing. This behavior is more aggressive than that specified in [RFC 2581](#), which says that a TCP sender SHOULD set its congestion window to the initial window after an idle period of an RTT or greater.

[RFC 3465](#) E: "TCP Congestion Control with Appropriate Byte Counting (ABC)" (February 2003)

This document [[RFC3465](#)] suggests that congestion control use the number of bytes acknowledged instead of the number of acknowledgments received. This has been implemented in Linux. The ABC mechanism behaves differently from the standard method when there is not a one-to-one relationship between data segments and acknowledgments. ABC still operates within the accepted guidelines, but is more robust to delayed ACKs and ACK-division [[SCWA99](#)][RFC3449].

[RFC 3522](#) E: "The Eifel Detection Algorithm for TCP" (April 2003)

The Eifel detection algorithm [[RFC3522](#)] allows a TCP sender to detect a posteriori whether it has entered loss recovery unnecessarily.

[RFC 3540](#) E: "Robust Explicit Congestion Notification (ECN) signaling with Nonces" (June 2003)

This document [[RFC3540](#)] suggests a modified ECN to address security concerns and updates [RFC 3168](#).

[RFC 3649](#) E: "HighSpeed TCP for Large Congestion Windows" (December 2003)

This document [[RFC3649](#)] suggests a modification to TCP's steady-state behavior to use very large windows efficiently.

[RFC 3708](#) E: "Using TCP Duplicate Selective Acknowledgement (DSACKs) and Stream Control Transmission Protocol (SCTP) Duplicate Transmission Sequence Numbers (TSNs) to Detect Spurious Retransmissions" (February 2004)

Abstract: "TCP and Stream Control Transmission Protocol (SCTP) provide notification of duplicate segment receipt through Duplicate Selective Acknowledgement (DSACKs) and Duplicate Transmission Sequence Number (TSN) notification, respectively. This document presents conservative methods of using this information to identify unnecessary retransmissions for various applications." [[RFC3708](#)]

[RFC 3742](#) E: "Limited Slow-Start for TCP with Large Congestion Windows" (March 2004)

This document [[RFC3742](#)] describes a more conservative slow-start behavior to prevent massive packet losses when a connection uses a very large window.

[RFC 4015](#) S: "The Eifel Response Algorithm for TCP" (February 2005)

This document [[RFC4015](#)] describes the response portion of the Eifel algorithm, which can be used in conjunction with one of several methods of detecting when loss recovery has been spuriously entered, such as the Eifel detection algorithm in [RFC 3522](#), the algorithm in [RFC 3708](#), or F-RTT in [RFC 4138](#).

Abstract: "Based on an appropriate detection algorithm, the Eifel response algorithm provides a way for a TCP sender to respond to a detected spurious timeout. It adapts the retransmission timer to avoid further spurious timeouts, and can avoid - depending on the detection algorithm - the often unnecessary go-back-N retransmits that would otherwise be sent. In addition, the Eifel response algorithm restores the congestion control state in such a way that packet bursts are avoided."

[RFC 4015](#) is itself a Proposed Standard. The consensus of the TCPM working group was to place it in this section of the roadmap document due to three factors.

1. [RFC 4015](#) operates on the output of a detection algorithm, for which there is currently no available mechanism on the standards track.
2. The working group was not aware of any wide deployment and use of [RFC 4015](#).
3. The consensus of the working group, after a discussion of the known Intellectual Property Rights claims on the techniques described in [RFC 4015](#), identified this section of the roadmap as an appropriate location.

[RFC 4138](#) E: "Forward RTT-Recovery (F-RTT): An Algorithm for Detecting Spurious Retransmission Timeouts with TCP and the Stream Control Transmission Protocol" (August 2005)

The F-RTT detection algorithm [[RFC4138](#)] provides another option for inferring spurious retransmission timeouts. Unlike some similar detection methods, F-RTT does not rely on the use of any TCP options.

5. Historic Extensions

The RFCs listed here define extensions that have thus far failed to arouse substantial interest from implementers, or that were found to be defective for general use.

[RFC 1106](#) "TCP Big Window and NAK Options" (June 1989): found defective

This RFC [[RFC1106](#)] defined an alternative to the Window Scale option for using large windows and described the "negative acknowledgement" or NAK option. There is a comparison of NAK and SACK methods, and early discussion of TCP over satellite issues. [RFC 1110](#) explains some problems with the approaches described in [RFC 1106](#). The options described in this document have not been adopted by the larger community, although NAKs are used in the SCPS-TP adaptation of TCP for satellite and spacecraft use, developed by the Consultative Committee for Space Data Systems (CCSDS).

[RFC 1110](#) "A Problem with the TCP Big Window Option" (August 1989): deprecates [RFC 1106](#)

Abstract: "The TCP Big Window option discussed in [RFC 1106](#) will not work properly in an Internet environment which has both a high bandwidth * delay product and the possibility of disordering and duplicating packets. In such networks, the window size must not be increased without a similar increase in the sequence number space. Therefore, a different approach to big windows should be taken in the Internet." [[RFC1110](#)]

[RFC 1146](#) E "TCP Alternate Checksum Options" (March 1990): lack of interest

This document [[RFC1146](#)] defined more robust TCP checksums than the 16-bit ones-complement in use today. A typographical error in [RFC 1145](#) is fixed in [RFC 1146](#); otherwise, the documents are the same.

[RFC 1263](#) "TCP Extensions Considered Harmful" (October 1991) - lack of interest

This document [[RFC1263](#)] argues against "backwards compatible" TCP extensions. Specifically mentioned are several TCP enhancements that have been successful, including timestamps, window scaling, PAWS, and SACK. [RFC 1263](#) presents an alternative approach called "protocol evolution", whereby several evolutionary versions of TCP would exist on hosts. These distinct TCP versions would represent upgrades to each other and could be header-incompatible.

Interoperability would be provided by having a virtualization layer select the right TCP version for a particular connection. This idea did not catch on with the community, although the type of extensions [RFC 1263](#) specifically targeted as harmful did become popular.

[RFC 1379](#) I "Extending TCP for Transactions -- Concepts" (November 1992): found defective

See [RFC 1644](#).

[RFC 1644](#) E "T/TCP -- TCP Extensions for Transactions Functional Specification" (July 1994): found defective

The inventors of TCP believed that cached connection state could have been used to eliminate TCP's 3-way handshake, to support two-packet request/response exchanges. RFCs 1379 [[RFC1379](#)] and 1644 [[RFC1644](#)] show that this is far from simple. Furthermore, T/TCP floundered on the ease of denial-of-service attacks that can result. One idea pioneered by T/TCP lives on in [RFC 2140](#), in the sharing of state across connections.

[RFC 1693](#) E "An Extension to TCP: Partial Order Service" (November 1994): lack of interest

This document [[RFC1693](#)] defines a TCP extension for applications that do not care about the order in which application-layer objects are received. Examples are multimedia and database applications. In practice, these applications either accept the possible performance loss because of TCP's strict ordering or use more specialized transport protocols.

[6.](#) Support Documents

This section contains several classes of documents that do not necessarily define current protocol behaviors, but that are nevertheless of interest to TCP implementers. [Section 6.1](#) describes several foundational RFCs that give modern readers a better understanding of the principles underlying TCP's behaviors and development over the years. The documents listed in [Section 6.2](#) provide advice on using TCP in various types of network situations that pose challenges above those of typical wired links. Some implementation notes can be found in [Section 6.3](#). The TCP Management Information Bases are described in [Section 6.4](#). RFCs that describe tools for testing and debugging TCP implementations or that contain high-level tutorials on the protocol are listed [Section 6.5](#), and [Section 6.6](#) lists a number of case studies that have explored TCP performance.

[6.1.](#) Foundational Works

The documents listed in this section contain information that is largely duplicated by the standards documents previously discussed. However, some of them contain a greater depth of problem statement explanation or other context. Particularly, RFCs 813 - 817 (known as the "Dave Clark Five") describe some early problems and solutions ([RFC 815](#) only describes the reassembly of IP fragments and is not included in this TCP roadmap).

[RFC 813](#): "Window and Acknowledgement Strategy in TCP" (July 1982)

This document [[RFC0813](#)] contains an early discussion of Silly Window Syndrome and its avoidance and motivates and describes the use of delayed acknowledgments.

[RFC 814](#): "Name, Addresses, Ports, and Routes" (July 1982)

Suggestions and guidance for the design of tables and algorithms to keep track of various identifiers within a TCP/IP implementation are provided by this document [[RFC0814](#)].

[RFC 816](#): "Fault Isolation and Recovery" (July 1982)

In this document [[RFC0816](#)], TCP's response to indications of network error conditions such as timeouts or received ICMP messages is discussed.

[RFC 817](#): "Modularity and Efficiency in Protocol Implementation" (July 1982)

This document [[RFC0817](#)] contains implementation suggestions that are general and not TCP specific. However, they have been used to develop TCP implementations and to describe some performance implications of the interactions between various layers in the Internet stack.

[RFC 872](#): "TCP-ON-A-LAN" (September 1982)

Conclusion: "The sometimes-expressed fear that using TCP on a local net is a bad idea is unfounded." [[RFC0872](#)]

[RFC 896](#): "Congestion Control in IP/TCP Internetworks" (January 1984)

This document [[RFC0896](#)] contains some early experiences with congestion collapse and some initial thoughts on how to avoid it using congestion control in TCP.

[RFC 964](#): "Some Problems with the Specification of the Military Standard Transmission Control Protocol" (November 1985)

This document [[RFC0964](#)] points out several specification bugs in the US Military's MIL-STD-1778 document, which was intended as a successor to [RFC 793](#). This serves to remind us of the difficulty in specification writing (even when we work from existing documents!).

[RFC 1072](#): "TCP Extensions for Long-Delay Paths" (October 1988)

This document [[RFC1072](#)] contains early explanations of the mechanisms that were later described by RFCs 1323 and 2018, which obsolete it.

[RFC 1185](#): "TCP Extension for High-Speed Paths" (October 1990)

This document [[RFC1185](#)] builds on [RFC 1072](#) to describe more advanced strategies for dealing with sequence number wrapping and detecting duplicates from earlier connections. This document was obsoleted by [RFC 1323](#).

[RFC 2914](#) B: "Congestion Control Principles" (September 2000)

This document [[RFC2914](#)] motivates the use of end-to-end congestion control for preventing congestion collapse and providing fairness to TCP.

6.2. Difficult Network Environments

As the internetworking field has explored wireless, satellite, cellular telephone, and other kinds of link-layer technologies, a large body of work has built up on enhancing TCP performance for such links. The RFCs listed in this section describe some of these more challenging network environments and how TCP interacts with them.

[RFC 2488](#) B: "Enhancing TCP Over Satellite Channels using Standard Mechanisms" (January 1999)

From abstract: "While TCP works over satellite channels there are several IETF standardized mechanisms that enable TCP to more effectively utilize the available capacity of the network path. This document outlines some of these TCP mitigations. At this time, all mitigations discussed in this document are IETF standards track mechanisms (or are compliant with IETF standards)." [[RFC2488](#)]

[RFC 2757](#) I: "Long Thin Networks" (January 2000)

Several methods of improving TCP performance over long thin networks, such as geosynchronous satellite links, are discussed in this document [[RFC2757](#)]. A particular set of TCP options is developed that should work well in such environments and be safe to use in the global Internet. The implications of such environments have been further discussed in [RFC 3150](#) and [RFC 3155](#), and these documents should be preferred where there is overlap between them and [RFC 2757](#).

[RFC 2760](#) I: "Ongoing TCP Research Related to Satellites" (February 2000)

This document [[RFC2760](#)] discusses the advantages and disadvantages of several different experimental means of improving TCP performance over long-delay or error-prone paths. These include T/TCP, larger initial windows, byte counting, delayed acknowledgments, slow start thresholds, NewReno and SACK-based loss recovery, FACK [[MM96](#)], ECN, various corruption-detection mechanisms, congestion avoidance changes for fairness, use of multiple parallel flows, pacing, header compression, state sharing, and ACK congestion control, filtering, and reconstruction. Although [RFC 2488](#) looks at standard extensions, this document focuses on more experimental means of performance enhancement.

[RFC 3135](#) I: "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations" (June 2001)

From abstract: "This document is a survey of Performance Enhancing Proxies (PEPs) often employed to improve degraded TCP performance caused by characteristics of specific link environments, for example, in satellite, wireless WAN, and wireless LAN environments. Different types of Performance Enhancing Proxies are described as well as the mechanisms used to improve performance." [[RFC3135](#)]

[RFC 3150](#) B: "End-to-end Performance Implications of Slow Links" (July 2001)

From abstract: "This document makes performance-related recommendations for users of network paths that traverse "very low bit-rate" links....This recommendation may be useful in any network where hosts can saturate available bandwidth, but the design space for this recommendation explicitly includes connections that traverse 56 Kb/second modem links or 4.8 Kb/second wireless access links - both of which are widely deployed." [[RFC3150](#)]

[RFC 3155](#) B: "End-to-end Performance Implications of Links with Errors" (August 2001)

From abstract: "This document discusses the specific TCP mechanisms that are problematic in environments with high uncorrected error rates, and discusses what can be done to mitigate the problems without introducing intermediate devices into the connection." [[RFC3155](#)]

[RFC 3366](#) "Advice to link designers on link Automatic Repeat reQuest (ARQ)" (August 2002)

From abstract: "This document provides advice to the designers of digital communication equipment and link-layer protocols employing link-layer Automatic Repeat reQuest (ARQ) techniques. This document presumes that the designers wish to support Internet protocols, but may be unfamiliar with the architecture of the Internet and with the implications of their design choices for the performance and efficiency of Internet traffic carried over their links." [[RFC3366](#)]

[RFC 3449](#) B: "TCP Performance Implications of Network Path Asymmetry" (December 2002)

From abstract: "This document describes TCP performance problems that arise because of asymmetric effects. These problems arise in several access networks, including bandwidth-asymmetric networks and packet radio subnetworks, for different underlying reasons. However, the end result on TCP performance is the same in both cases: performance often degrades significantly because of imperfection and variability in the ACK feedback from the receiver to the sender.

The document details several mitigations to these effects, which have either been proposed or evaluated in the literature, or are currently deployed in networks." [[RFC3449](#)]

[RFC 3481](#) B: "TCP over Second (2.5G) and Third (3G) Generation Wireless Networks" (February 2003)

From abstract: "This document describes a profile for optimizing TCP to adapt so that it handles paths including second (2.5G) and third (3G) generation wireless networks." [[RFC3481](#)]

[RFC 3819](#) B: "Advice for Internet Subnetwork Designers" (July 2004)

This document [[RFC3819](#)] describes how TCP performance can be negatively affected by some particular lower-layer behaviors and provides guidance in designing lower-layer networks and protocols to be amicable to TCP.

6.3. Implementation Advice

[RFC 879](#): "The TCP Maximum Segment Size and Related Topics" (November 1983)

Abstract: "This memo discusses the TCP Maximum Segment Size Option and related topics. The purposes is to clarify some aspects of TCP and its interaction with IP. This memo is a clarification to the TCP specification, and contains information that may be considered as 'advice to implementers'." [[RFC0879](#)]

[RFC 1071](#): "Computing the Internet Checksum" (September 1988)

This document [[RFC1071](#)] lists a number of implementation techniques for efficiently computing the Internet checksum (used by TCP).

[RFC 1624](#) I: "Computation of the Internet Checksum via Incremental Update" (May 1994)

Incrementally updating the Internet checksum is useful to routers in updating IP checksums. Some middleboxes that alter TCP headers may also be able to update the TCP checksum incrementally. This document [[RFC1624](#)] expands upon the explanation of the incremental update procedure in [RFC 1071](#).

[RFC 1936](#) I: "Implementing the Internet Checksum in Hardware" (April 1996)

This document [[RFC1936](#)] describes the motivation for implementing the Internet checksum in hardware, rather than in software, and provides an implementation example.

[RFC 2525](#) I: "Known TCP Implementation Problems" (March 1999)

From abstract: "This memo catalogs a number of known TCP implementation problems. The goal in doing so is to improve conditions in the existing Internet by enhancing the quality of current TCP/IP implementations." [[RFC2525](#)]

[RFC 2923](#) I: "TCP Problems with Path MTU Discovery" (September 2000)

From abstract: "This memo catalogs several known Transmission Control Protocol (TCP) implementation problems dealing with Path Maximum Transmission Unit Discovery (PMTUD), including the long-standing black hole problem, stretch acknowledgements (ACKs) due to confusion between Maximum Segment Size (MSS) and segment size, and MSS advertisement based on PMTU." [[RFC2923](#)]

[RFC 3360](#) B: "Inappropriate TCP Resets Considered Harmful" (August 2002)

This document [[RFC3360](#)] is a plea that firewall vendors not send gratuitous TCP RST (Reset) packets when unassigned TCP header bits are used. This practice prevents desirable extension and evolution of the protocol and thus is potentially harmful to the future of the Internet.

[RFC 3493](#) I: "Basic Socket Interface Extensions for IPv6" (February 2003)

This document [[RFC3493](#)] describes the de facto standard sockets API for programming with TCP. This API is implemented nearly ubiquitously in modern operating systems and programming languages.

[6.4.](#) Management Information Bases

The first MIB module defined for use with Simple Network Management Protocol (SNMP) (in [RFC 1066](#) and its update, [RFC 1156](#)) was a single monolithic MIB module, called MIB-I. This evolved over time to be MIB-II ([RFC 1213](#)). It then became apparent that having a single monolithic MIB module was not scalable, given the number and breadth of MIB data definitions that needed to be included. Thus, additional MIB modules were defined, and those parts of MIB-II that needed to evolve were split off. Eventually, the remaining parts of MIB-II were also split off, the TCP-specific part being documented in [RFC 2012](#).

[RFC 2012](#) was obsoleted by [RFC 4022](#), which is the primary TCP MIB document today. MIB-I, defined in [RFC 1156](#), has been obsoleted by the MIB-II specification in [RFC 1213](#). For current TCP implementers, [RFC 4022](#) should be supported.

[RFC 1066](#): "Management Information Base for Network Management of TCP/IP-based Internets" (August 1988)

This document [[RFC1066](#)] was the description of the TCP MIB. It was obsoleted by [RFC 1156](#).

[RFC 1156](#) S: "Management Information Base for Network Management of TCP/IP-based Internets" (May 1990)

This document [[RFC1156](#)] describes the required MIB fields for TCP implementations, with minor corrections and no technical changes from [RFC 1066](#), which it obsoletes. This is the standards track document for MIB-I.

[RFC 1213](#) S: "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II" (March 1991)

This document [[RFC1213](#)] describes the second version of the MIB in a monolithic form. [RFC 2012](#) updates this document by splitting out the TCP-specific portions.

[RFC 2012](#) S: "SNMPv2 Management Information Base for the Transmission Control Protocol using SMIV2" (November 1996)

This document [[RFC2012](#)] defined the TCP MIB, in an update to [RFC 1213](#). It is now obsoleted by [RFC 4022](#).

[RFC 2452](#) S: "IP Version 6 Management Information Base for the Transmission Control Protocol" (December 1998)

This document [[RFC2452](#)] augments [RFC 2012](#) by adding an IPv6-specific connection table. The rest of 2012 holds for any IP version. [RFC 2012](#) is now obsoleted by [RFC 4022](#).

Although it is a standards track document, [RFC 2452](#) is considered a historic mistake by the MIB community, as it is based on the idea of parallel IPv4 and IPv6 structures. Although IPv6 requires new structures, the community has decided to define a single generic structure for both IPv4 and IPv6. This will aid in definition, implementation, and transition between IPv4 and IPv6.

[RFC 4022](#) S: "Management Information Base for the Transmission Control Protocol (TCP)" (March 2005)

This document [[RFC4022](#)] obsoletes [RFC 2012](#) and [RFC 2452](#) and specifies the current standard for the TCP MIB that should be deployed.

6.5. Tools and Tutorials

[RFC 1180](#) I: "TCP/IP Tutorial" (January 1991)

This document [[RFC1180](#)] is an extremely brief overview of the TCP/IP protocol suite as a whole. It gives some explanation as to how and where TCP fits in.

[RFC 1470](#) I: "FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices" (June 1993)

A few of the tools that this document [[RFC1470](#)] describes are still maintained and in use today; for example, `ttcp` and `tcpdump`. However, many of the tools described do not relate specifically to TCP and are no longer used or easily available.

[RFC 2398](#) I: "Some Testing Tools for TCP Implementors" (August 1998)

This document [[RFC2398](#)] describes a number of TCP packet generation and analysis tools. Although some of these tools are no longer readily available or widely used, for the most part they are still relevant and usable.

6.6. Case Studies

[RFC 1337](#) I: "TIME-WAIT Assassination Hazards in TCP" (May 1992)

This document [[RFC1337](#)] points out a problem with acting on received reset segments while one is in the TIME-WAIT state. The main recommendation is that hosts in TIME-WAIT ignore resets. This recommendation might not currently be widely implemented.

[RFC 2415](#) I: "Simulation Studies of Increased Initial TCP Window Size" (September 1998)

This document [[RFC2415](#)] presents results of some simulations using TCP initial windows greater than 1 segment. The analysis indicates that user-perceived performance can be improved by increasing the initial window to 3 segments.

[RFC 2416](#) I: "When TCP Starts Up With Four Packets Into Only Three Buffers" (September 1998)

This document [[RFC2416](#)] uses simulation results to clear up some concerns about using an initial window of 4 segments when the network path has less provisioning.

[RFC 2884](#) I: "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks" (July 2000)

This document [[RFC2884](#)] describes experimental results that show some improvements to the performance of both short- and long-lived connections due to ECN.

7. Undocumented TCP Features

There are a few important implementation tactics for the TCP that have not yet been described in any RFC. Although this roadmap is primarily concerned with mapping the TCP RFCs, this section is included because an implementer needs to be aware of these important issues.

SYN Cookies

A mechanism known as "SYN cookies" is widely used to thwart TCP SYN flooding attacks, in which an attacker sends a flood of SYNs to a victim but fails to complete the 3-way handshake. The result is exhaustion of resources at the server. The SYN cookie mechanism allows the server to return a cleverly chosen initial sequence number that has all the required state for the secure completion of the handshake. Then the server can avoid saving connection state during the 3-way handshake and thus survive a SYN flooding attack.

A web search for "SYN cookies" will reveal a number of useful descriptions of this mechanism, although there is currently no RFC on the matter.

Header Prediction

Header prediction is a trick to speed up the processing of segments. Van Jacobson and Mike Karels developed the technique in the late 1980s. The basic idea is that some processing time can be saved when most of a segment's fields can be predicted from previous segments. A good description of this was sent to the TCP-IP mailing list by Van Jacobson on March 9, 1988:

Quite a bit of the speedup comes from an algorithm that we ('we' refers to collaborator Mike Karels and myself) are calling "header prediction". The idea is that if you're in the middle of a bulk data transfer and have just seen a packet, you know what the next packet is going to look like: It will look just like the current packet with either the sequence number or ack number updated (depending on whether you're the sender or receiver). Combining this with the "Use hints" epigram from Butler Lampson's classic "Epigrams for System Designers", you start to think of the tcp state (rcv.nxt, snd.una, etc.) as "hints" about what the next packet should look like.

If you arrange those "hints" so they match the layout of a tcp packet header, it takes a single 14-byte compare to see if your prediction is correct (3 longword compares to pick up the send & ack sequence numbers, header length, flags and window, plus a short compare on the length). If the prediction is correct, there's a single test on the length to see if you're the sender or receiver followed by the appropriate processing. E.g., if the length is non-zero (you're the receiver), checksum and append the data to the socket buffer then wake any process that's sleeping on the buffer. Update rcv.nxt by the length of this packet (this updates your "prediction" of the next packet). Check if you can handle another packet the same size as the current one. If not, set one of the unused flag bits in your header prediction to guarantee that the prediction will fail on the next packet and force you to go through full protocol processing. Otherwise, you're done with this packet. So, the **total** tcp protocol processing, exclusive of checksumming, is on the order of 6 compares and an add.

8. Security Considerations

This document introduces no new security considerations. Each RFC listed in this document attempts to address the security considerations of the specification it contains.

9. Acknowledgments

This document grew out of a discussion on the end2end-interest mailing list, the public list of the End-to-End Research Group of the IRTF, and continued development under the IETF's TCP Maintenance and Minor Extensions (TCPM) working group. We thank Joe Touch, Reiner Ludwig, Pekka Savola, Gorrry Fairhurst, and Sally Floyd for their contributions, in particular. The chairs of the TCPM working group, Mark Allman and Ted Faber, have been instrumental in the development of this document. Keith McCloghrie provided some useful notes and clarification on the various MIB-related RFCs.

10. Informative References

10.1. Basic Functionality

- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.
- [RFC1122] Braden, R., "Requirements for Internet Hosts - Communication Layers", STD 3, [RFC 1122](#), October 1989.
- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", [BCP 9](#), [RFC 2026](#), October 1996.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", [RFC 2474](#), December 1998.
- [RFC2581] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", [RFC 2581](#), April 1999.
- [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", [RFC 2675](#), August 1999.
- [RFC2873] Xiao, X., Hannan, A., Paxson, V., and E. Crabbe, "TCP Processing of the IPv4 Precedence Field", [RFC 2873](#), June 2000.
- [RFC2988] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", [RFC 2988](#), November 2000.

10.2. Recommended Enhancements

- [RFC1323] Jacobson, V., Braden, R., and D. Borman, "TCP Extensions for High Performance", [RFC 1323](#), May 1992.
- [RFC1948] Bellovin, S., "Defending Against Sequence Number Attacks", [RFC 1948](#), May 1996.
- [RFC2018] Mathis, M., Mahdavi, J., Floyd, S., and A. Romanow, "TCP Selective Acknowledgment Options", [RFC 2018](#), October 1996.
- [RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", [RFC 2385](#), August 1998.

- [RFC2883] Floyd, S., Mahdavi, J., Mathis, M., and M. Podolsky, "An Extension to the Selective Acknowledgement (SACK) Option for TCP", [RFC 2883](#), July 2000.
- [RFC3042] Allman, M., Balakrishnan, H., and S. Floyd, "Enhancing TCP's Loss Recovery Using Limited Transmit", [RFC 3042](#), January 2001.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", [RFC 3168](#), September 2001.
- [RFC3390] Allman, M., Floyd, S., and C. Partridge, "Increasing TCP's Initial Window", [RFC 3390](#), October 2002.
- [RFC3517] Blanton, E., Allman, M., Fall, K., and L. Wang, "A Conservative Selective Acknowledgment (SACK)-based Loss Recovery Algorithm for TCP", [RFC 3517](#), April 2003.
- [RFC3562] Leech, M., "Key Management Considerations for the TCP MD5 Signature Option", [RFC 3562](#), July 2003.
- [RFC3782] Floyd, S., Henderson, T., and A. Gurtov, "The NewReno Modification to TCP's Fast Recovery Algorithm", [RFC 3782](#), April 2004.
- [RFC4015] Ludwig, R. and A. Gurtov, "The Eifel Response Algorithm for TCP", [RFC 4015](#), February 2005.
- [RFC4278] Bellovin, S. and A. Zinin, "Standards Maturity Variance Regarding the TCP MD5 Signature Option ([RFC 2385](#)) and the BGP-4 Specification", [RFC 4278](#), January 2006.

10.3. Experimental Extensions

- [RFC2140] Touch, J., "TCP Control Block Interdependence", [RFC 2140](#), April 1997.
- [RFC2861] Handley, M., Padhye, J., and S. Floyd, "TCP Congestion Window Validation", [RFC 2861](#), June 2000.
- [RFC3124] Balakrishnan, H. and S. Seshan, "The Congestion Manager", [RFC 3124](#), June 2001.
- [RFC3465] Allman, M., "TCP Congestion Control with Appropriate Byte Counting (ABC)", [RFC 3465](#), February 2003.

- [RFC3522] Ludwig, R. and M. Meyer, "The Eifel Detection Algorithm for TCP", [RFC 3522](#), April 2003.
- [RFC3540] Spring, N., Wetherall, D., and D. Ely, "Robust Explicit Congestion Notification (ECN) Signaling with Nonces", [RFC 3540](#), June 2003.
- [RFC3649] Floyd, S., "HighSpeed TCP for Large Congestion Windows", [RFC 3649](#), December 2003.
- [RFC3708] Blanton, E. and M. Allman, "Using TCP Duplicate Selective Acknowledgement (DSACKs) and Stream Control Transmission Protocol (SCTP) Duplicate Transmission Sequence Numbers (TSNs) to Detect Spurious Retransmissions", [RFC 3708](#), February 2004.
- [RFC3742] Floyd, S., "Limited Slow-Start for TCP with Large Congestion Windows", [RFC 3742](#), March 2004.
- [RFC4138] Sarolahti, P. and M. Kojo, "Forward RTO-Recovery (F-RTO): An Algorithm for Detecting Spurious Retransmission Timeouts with TCP and the Stream Control Transmission Protocol (SCTP)", [RFC 4138](#), August 2005.

10.4. Historic Extensions

- [RFC1106] Fox, R., "TCP big window and NAK options", [RFC 1106](#), June 1989.
- [RFC1110] McKenzie, A., "Problem with the TCP big window option", [RFC 1110](#), August 1989.
- [RFC1146] Zweig, J. and C. Partridge, "TCP alternate checksum options", [RFC 1146](#), March 1990.
- [RFC1263] O'Malley, S. and L. Peterson, "TCP Extensions Considered Harmful", [RFC 1263](#), October 1991.
- [RFC1379] Braden, R., "Extending TCP for Transactions -- Concepts", [RFC 1379](#), November 1992.
- [RFC1644] Braden, R., "T/TCP -- TCP Extensions for Transactions Functional Specification", [RFC 1644](#), July 1994.
- [RFC1693] Connolly, T., Amer, P., and P. Conrad, "An Extension to TCP : Partial Order Service", [RFC 1693](#), November 1994.

10.5. Support Documents

- [RFC0813] Clark, D., "Window and Acknowledgement Strategy in TCP", [RFC 813](#), July 1982.
- [RFC0814] Clark, D., "Name, addresses, ports, and routes", [RFC 814](#), July 1982.
- [RFC0816] Clark, D., "Fault isolation and recovery", [RFC 816](#), July 1982.
- [RFC0817] Clark, D., "Modularity and efficiency in protocol implementation", [RFC 817](#), July 1982.
- [RFC0872] Padlipsky, M., "TCP-on-a-LAN", [RFC 872](#), September 1982.
- [RFC0879] Postel, J., "TCP maximum segment size and related topics", [RFC 879](#), November 1983.
- [RFC0896] Nagle, J., "Congestion control in IP/TCP internetworks", [RFC 896](#), January 1984.
- [RFC0964] Sidhu, D. and T. Blumer, "Some problems with the specification of the Military Standard Transmission Control Protocol", [RFC 964](#), November 1985.
- [RFC1066] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", [RFC 1066](#), August 1988.
- [RFC1071] Braden, R., Borman, D., and C. Partridge, "Computing the Internet checksum", [RFC 1071](#), September 1988.
- [RFC1072] Jacobson, V. and R. Braden, "TCP extensions for long-delay paths", [RFC 1072](#), October 1988.
- [RFC1156] McCloghrie, K. and M. Rose, "Management Information Base for network management of TCP/IP-based internets", [RFC 1156](#), May 1990.
- [RFC1180] Socolofsky, T. and C. Kale, "TCP/IP tutorial", [RFC 1180](#), January 1991.
- [RFC1185] Jacobson, V., Braden, B., and L. Zhang, "TCP Extension for High-Speed Paths", [RFC 1185](#), October 1990.

- [RFC1213] McCloghrie, K. and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, [RFC 1213](#), March 1991.
- [RFC1337] Braden, R., "TIME-WAIT Assassination Hazards in TCP", [RFC 1337](#), May 1992.
- [RFC1470] Enger, R. and J. Reynolds, "FYI on a Network Management Tool Catalog: Tools for Monitoring and Debugging TCP/IP Internets and Interconnected Devices", FYI 2, [RFC 1470](#), June 1993.
- [RFC1624] Rijssinghani, A., "Computation of the Internet Checksum via Incremental Update", [RFC 1624](#), May 1994.
- [RFC1936] Touch, J. and B. Parham, "Implementing the Internet Checksum in Hardware", [RFC 1936](#), April 1996.
- [RFC2012] McCloghrie, K., "SNMPv2 Management Information Base for the Transmission Control Protocol using SMIV2", [RFC 2012](#), November 1996.
- [RFC2398] Parker, S. and C. Schmechel, "Some Testing Tools for TCP Implementors", [RFC 2398](#), August 1998.
- [RFC2415] Poduri, K. and K. Nichols, "Simulation Studies of Increased Initial TCP Window Size", [RFC 2415](#), September 1998.
- [RFC2416] Shepard, T. and C. Partridge, "When TCP Starts Up With Four Packets Into Only Three Buffers", [RFC 2416](#), September 1998.
- [RFC2452] Daniele, M., "IP Version 6 Management Information Base for the Transmission Control Protocol", [RFC 2452](#), December 1998.
- [RFC2488] Allman, M., Glover, D., and L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", [BCP 28](#), [RFC 2488](#), January 1999.
- [RFC2525] Paxson, V., Allman, M., Dawson, S., Fenner, W., Griner, J., Heavens, I., Lahey, K., Semke, J., and B. Volz, "Known TCP Implementation Problems", [RFC 2525](#), March 1999.
- [RFC2757] Montenegro, G., Dawkins, S., Kojo, M., Magret, V., and N. Vaidya, "Long Thin Networks", [RFC 2757](#), January 2000.

- [RFC2760] Allman, M., Dawkins, S., Glover, D., Griner, J., Tran, D., Henderson, T., Heidemann, J., Touch, J., Kruse, H., Ostermann, S., Scott, K., and J. Semke, "Ongoing TCP Research Related to Satellites", [RFC 2760](#), February 2000.
- [RFC2884] Hadi Salim, J. and U. Ahmed, "Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks", [RFC 2884](#), July 2000.
- [RFC2914] Floyd, S., "Congestion Control Principles", [BCP 41](#), [RFC 2914](#), September 2000.
- [RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", [RFC 2923](#), September 2000.
- [RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", [RFC 3135](#), June 2001.
- [RFC3150] Dawkins, S., Montenegro, G., Kojo, M., and V. Magret, "End-to-end Performance Implications of Slow Links", [BCP 48](#), [RFC 3150](#), July 2001.
- [RFC3155] Dawkins, S., Montenegro, G., Kojo, M., Magret, V., and N. Vaidya, "End-to-end Performance Implications of Links with Errors", [BCP 50](#), [RFC 3155](#), August 2001.
- [RFC3360] Floyd, S., "Inappropriate TCP Resets Considered Harmful", [BCP 60](#), [RFC 3360](#), August 2002.
- [RFC3366] Fairhurst, G. and L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)", [BCP 62](#), [RFC 3366](#), August 2002.
- [RFC3449] Balakrishnan, H., Padmanabhan, V., Fairhurst, G., and M. Sooriyabandara, "TCP Performance Implications of Network Path Asymmetry", [BCP 69](#), [RFC 3449](#), December 2002.
- [RFC3481] Inamura, H., Montenegro, G., Ludwig, R., Gurtov, A., and F. Khafizov, "TCP over Second (2.5G) and Third (3G) Generation Wireless Networks", [BCP 71](#), [RFC 3481](#), February 2003.
- [RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", [RFC 3493](#), February 2003.

- [RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", [BCP 89](#), [RFC 3819](#), July 2004.
- [RFC4022] Raghunarayan, R., "Management Information Base for the Transmission Control Protocol (TCP)", [RFC 4022](#), March 2005.

10.6. Informative References Outside the RFC Series

- [JK92] Jacobson, V. and M. Karels, "Congestion Avoidance and Control", This paper is a revised version of [[Jac88](#)], that includes an additional appendix. This paper has not been traditionally published, but is currently available at [ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z](http://ftp.ee.lbl.gov/papers/congavoid.ps.Z). 1992.
- [Jac88] Jacobson, V., "Congestion Avoidance and Control", ACM SIGCOMM 1988 Proceedings, in ACM Computer Communication Review, 18 (4), pp. 314-329, August 1988.
- [KP87] Karn, P. and C. Partridge, "Round Trip Time Estimation", ACM SIGCOMM 1987 Proceedings, in ACM Computer Communication Review, 17 (5), pp. 2-7, August 1987
- [MAF04] Medina, A., Allman, M., and S. Floyd, "Measuring the Evolution of Transport Protocols in the Internet", ACM Computer Communication Review, 35 (2), April 2005.
- [MM96] Mathis, M. and J. Mahdavi, "Forward Acknowledgement: Refining TCP Congestion Control", ACM SIGCOMM 1996 Proceedings, in ACM Computer Communication Review 26 (4), pp. 281-292, October 1996.
- [SCWA99] Savage, S., Cardwell, N., Wetherall, D., and T. Anderson, "TCP Congestion Control with a Misbehaving Receiver", ACM Computer Communication Review, 29 (5), pp. 71-78, October 1999.

Authors' Addresses

Martin H. Duke
The Boeing Company
PO Box 3707, MC 7L-49
Seattle, WA 98124-2207

Phone: 425-373-2852
EMail: martin.duke@boeing.com

Robert Braden
USC Information Sciences Institute
Marina del Rey, CA 90292-6695

Phone: 310-448-9173
EMail: braden@isi.edu

Wesley M. Eddy
Verizon Federal Network Systems
21000 Brookpark Rd, MS 54-5
Cleveland, OH 44135

Phone: 216-433-6682
EMail: weddy@grc.nasa.gov

Ethan Blanton
Purdue University Computer Science
250 N. University St.
West Lafayette, IN 47907

EMail: eblanton@cs.purdue.edu

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

