Network Working Group Name
Internet Draft
Document: draft-irtf-iccrg-wetzl-
congestion-control-open-research-00.txt

Michael Welzl
Dimitri Papadimitriou
Editors

Michael Scharf

Expires: December 2007
July 2007

Open Research Issues in Internet Congestion Control

draft-irtf-iccrg-welzl-congestion-control-open-research-00.txt

Status of this Memo

This Internet-Draft will expire on December 31, 2007.

Copyright Notice

Abstract

This document describes many of the open problems in Internet congestion control that are known today. This includes several new challenges that are becoming important as the network grows, as well

as some issues that have been known for many years. These challenges are generally considered to be open research topics that may require more study or application of innovative techniques before Internet-scale solutions can be confidently engineered and deployed.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [i].

Table of Contents

1. Introduction

This document describes many of the open research topics in the domain of Internet congestion control that are known today. We begin by reviewing some proposed definitions of congestion and congestion control based on current understandings.

Congestion is defined as the reduction in utility due to overload in networks that support both spatial and temporal multiplexing, but no reservation [Keshav]. Congestion control is a distributed algorithm to share network resources among competing traffic sources. Two components of congestion control have been defined: the primal and the dual [Kelly98]. Primal congestion control is based on the traffic sources algorithm controlling their sending rates or window sizes depending on the congestion indication feedback signals they get from routers (dynamic feedback-based adjustment). TCP algorithms carry out the primal iteration. Dual congestion control is implemented by the routers through gathering information from the traffic flows that are using them. Routers congestion control algorithm updates, implicitly or explicitly, a congestion measure and sends it back, implicitly or explicitly, to the traffic sources that use that link. Queue management algorithms such as Random Early Detection (RED) [Floyd93] or Random Exponential Marking (REM) [Ath01] carry out the dual iteration.

Congestion control provides for a fundamental set of mechanisms for maintaining the stability and efficiency of the Internet operations. Congestion control has been associated with TCP since Van Jacobson's work in 1988, but also outside of TCP (e.g. for real-time multimedia applications, multicast, and router-based mechanisms). The Van Jacobson end-to-end congestion control algorithms [Jacobson88] [RFC2581] are used by the Internet transport protocols TCP [RFC793]. They have been proven to be highly successful over many years but have begun to reach their limits. Indeed, heterogeneity of both data link/physical layer and applications are pulling TCP congestion control (that performs poorly as bandwidth or delay increases) outside of its natural operating regime. A side effect of these deficits is that there is an increasing share of hosts that use non-standardized congestion control enhancements (for instance, many Linux distributions are shipped with "CUBIC" as default TCP congestion control.)

From the original Jacobson algorithm requiring no congestion-related state in routers, more recent modifications have backed off from this purity. Active Queue Management (AQM) in routers, e.g., RED and all its variants, xCHOKE [Pan00], RED with In/Out (RIO) [Clark98], etc. improves performance by keeping queues small (implicit feedback), while Explicit Congestion Notification (ECN) [Floyd94] [RFC3168] passes one bit of congestion information back to senders. These measures do improve performance, but there is a limit to how much can be accomplished without more information from routers. The requirement of extreme scalability together with robustness has been a difficult hurdle to accelerating information flow. Primal-Dual TCP/AQM distributed algorithm stability and equilibrium properties have been extensively studied in [Low02] [Low03].

In addition, congestion control includes many new challenges that are becoming important as the network grows, in addition to the issues that have been known for many years. These are generally considered to be open research topics that may require more study or application of innovative techniques before Internet-scale solutions can be confidently engineered and deployed.

2. Global Challenges - Overview

3. Detailed Challenges

3.1 Challenge 1: Router Support

Routers can be involved in congestion control in two ways: First, they can implicitly optimize their functions, such as queue management and scheduling strategies, in order to support the operation of an end-to-end congestion control.

Various approaches have been proposed and also deployed, such as different AQM techniques. Even though these implicit techniques are known to improve network performance during congestion phases, they are still only partly deployed in the Internet. This may be due to the fact that finding optimal and robust parameterizations for these mechanisms is a non-trivial problem. Indeed, the problem with various AQM schemes is the difficulty to identify correct values of the parameter set that affects the performance of the queuing scheme (due to variation in the number of sources, the capacity and the feedback delay) [Fioriu00] [Hollot01] [Zhang03]. None of the AQM schemes (RED, REM, BLUE, PI-Controller but also Adaptive Virtual Queue (AVQ) define a systematic rule for setting its parameters.

Second, routers can participate in congestion control by explicit notification mechanisms. By such feedback from the network, connection endpoints can obtain more accurate information about the current network characteristics on the path. This allows endpoints to make more precise decisions that can better prevent packet loss and that can also improve fairness among different flows. Examples for explicit router feedback include Explicit Congestion Notification (ECN) [RFC3168], Quick-Start [RFC4782], and eXplicit Control Protocol (XCP) [Katabi02] [Falk07].

With increasing the per-flow bandwidth-delay product increases, TCP becomes inefficient and prone to instability, regardless of the queuing scheme. XCP, which generalizes ECN, has been developed to address these issues, using per-packet feedback. By decoupling resource utilization/congestion control from fairness control, XCP outperforms TCP in conventional and high bandwidth-delay environments, and remains efficient, fair, scalable, and stable regardless of the link capacity, the round trip delay, and the number of sources. XCP aims at achieving fair bandwidth allocation, high utilization, small standing queue size, and near-zero packet drops, with both steady and highly varying traffic. Importantly, XCP does not maintain any per-flow state in routers and requires few CPU cycles per packet, hence portable to high-speed routers. However, XCP is still subject to research efforts: [Andrew05] has recently pointed out cases where in which XCP is stable locally but unstable globally (when the maximum RTT of a flow is much larger than the mean RTT). This instability can be removed by setting the estimation interval to be the maximum observed RTT, rather than the mean RTT. Nevertheless, this makes the system vulnerable to erroneous RTT advertisements. [PAP02] shows that when flows with different RTTs are applied, XCP sometimes discriminates among heterogeneous traffic flows, even if XCP is generally fair to different flows even if they belong to significantly heterogeneous flows. [Low05] provides for a complete characterization of the XCP equilibrium properties.

In general, such router support raises many issues that have not been completely solved yet:

3.1.1 Performance and robustness

Congestion control requires some tradeoffs: On the one hand, it must allow high link utilizations and fair resource sharing. But on the other hand the algorithms must also be robust and conservative in particular during congestion phases.

Router support can help to improve performance and fairness, but it can also result in additional complexity and more control loops. This requires a careful design of the algorithms in order to ensure stability and avoid e.g. oscillations. A further challenge is the fact that information may be imprecise. For instance, severe congestion can delay feedback signals. Also, the measurement of parameters such as round-trip times (RTT) or data rates may contain estimation errors. Even though there has been significant progress in providing fundamental theoretical models for such effects, research has not completely explored the whole problem space yet.

Open questions are:

- How much can routers theoretically improve performance in the complete range of communication scenarios that exists in the Internet?
- Is it possible to design robust mechanisms that offer significant benefits without additional risks?

3.1.2 Granularity of router functions

There are several degrees of freedom concerning router involvement, ranging from some few additional functions in network management procedures one the one end, and additional per packet processing on the other end of the solution space. Furthermore, different amounts of state can be kept in routers (no per-flow state, partial per-flow state, soft state per flows, hard state per flow). The additional router processing a challenge for Internet scalability and could also increase the end-to-end latencies.

There are many solutions that do not require per-flow state and thus do not cause a large processing overhead. However, scalability issues could also be caused, for instance, by synchronization mechanisms for state information among parallel processing entities, which are e. g. used in high-speed router hardware designs.

Open questions are:

- What granularity of router processing can be realized without affecting the Internet scalability?

- How can additional processing efforts be kept at a minimum?

3.1.3 Information acquisition

In order to support congestion control, routers have to obtain at least a subset of the following information. Obtaining that information may result in complex tasks.

1. Capacity of (outgoing) links

Link characteristics depend on the realization of lower protocol layers. Routers do not necessarily know the link layer network topology and link capacities, and these are not necessarily constant (e. g., on shared wireless links). Difficulties also arise when using IP-in-IP tunnels [RFC 2003] or MPLS [RFC3031] [RFC3032]. In these cases, link information could be determined by cross-layer information exchange, but this requires link layer technology specific interfaces. An alternative could be online measurements, but this can cause significant additional network overhead.

2. Traffic carried over (outgoing) links

Accurate online measurement of data rates is challenging when traffic is bursty. For instance, it is impossible to define and measure a current link load. This is a challenge for proposals that require knowledge e.g. about the current link utilization.

3. Internal buffer statistics

Some proposals use buffer statistics such as a virtual queue length to trigger feedback. However, routers can include multiple distributed buffer stages that make it difficult to obtain such metrics.

Open questions are: Can this information be made available, e.g., by additional interfaces or protocols?

3.1.4 Feedback signaling

Explicit notification mechanisms can be realized either by in-band signaling or by out-of-band signaling. The latter case requires additional protocols and can be further subdivided into path-coupled and path-decoupled approaches.

In-band signaling can be considered to be an appropriate choice: Since notifications are piggy-packet along with data traffic, there is less overhead and implementation complexity remains limited. Path-coupled out-of-band signaling could however be possible, too.

Open questions concerning feedback signaling include:

- At which protocol layer should the feedback occur (IP/network layer assisted, transport layer assisted, hybrid solutions, shim layer /intermediate sub-layer, etc.)?
- What is the optimal frequency of feedback (only in case of congestion events, per RTT, per packet, etc.)?

3.2 Challenge 2: Dynamic Range of Requirements

The Internet encompasses a large variety of heterogeneous IP networks that are realized by a multitude of technologies, which result in a tremendous variety of link and path characteristics: capacity can be either scarce in very slow speed radio links (several kbps), or there may be an abundant supply in high-speed optical links (several gigabit per second). Concerning latency, scenarios range from local interconnects (much less than a millisecond) to certain wireless and satellite links with very large latencies (up to a second). Even higher latencies can occur in interstellar communication. As a consequence, both the available bandwidth and the end-to-end delay in the Internet may vary over many orders of magnitude, and it is likely that the range of parameters will further increase in future.

Additionally, neither available bandwidth nor end-to-end delays are constant. At the IP layer, competing cross-traffic, traffic management in routers, and dynamic routing can result in sudden changes of the characteristics of the path followed from the source to the destination. Additional dynamics can be caused by link layer mechanisms, such as shared media access (e.g., in wireless networks), changes of links (horizontal/vertical handovers), topology modifications (e. g., in ad-hoc networks), link layer error correction, dynamic bandwidth provisioning schemes, etc. From this follows that path characteristics can be subject to substantial changes within short time frames.

The congestion control algorithms have to deal with this variety in an efficient way. The congestion control principles introduced by V. Jacobson assume a rather static scenario and implicitly target at configurations where the bandwidth−delay product is of the order of some dozens of packets at most. While these principles have proved to work well in the Internet for almost two decades, much larger bandwidth−delay products and increased dynamics challenge them more and more. There are many situations where today’s congestion control algorithms react in a suboptimal way, resulting in low resource utilization, non−optimal congestion avoidance, or unfairness.

This gave rise to a multitude of new proposals for congestion control algorithms. For instance, since the additive−increase multiplicative decrease (AIMD) principle of TCP does not scale well to large congestion window sizes, several high−speed congestion control extensions have been developed recently, such as High−Speed TCP, Scalable TCP, Fast TCP and BIC/CUBIC. However, these new algorithms raise fairness issues, and they may be less robust in certain situations for which they have not been designed.

However, there is still no common agreement in the IETF on which algorithm and protocol to choose. For instance, XCP could solve some problems caused by high bandwidth−delay products, at the cost of some additional complexity in routers. Also note that XCP may have some problems with dynamic changes of link layer characteristics as they are discussed in this section (shared media etc.). Similarly, proprietary congestion control mechanisms have been proposed for other specific environments, e.g., to cope with highly variable data rates.

It is always possible to tune congestion control parameters based on some knowledge about the environment and the application scenario. However, the fundamental question is whether it is possible to define one congestion control mechanism that operates reasonable well in the whole range of scenarios that exist in the Internet. Hence, it is an open research question how such a "unified" congestion control would have to be designed, and which maximum degree of dynamics it could efficiently handle.

3.3 Challenge 3: Corruption Loss

It is common for congestion control mechanisms to interpret packet loss as a sign of congestion. This is appropriate when packets are dropped in routers because of a queue that overflows, but there are other possible reasons for packet drops. In particular, in wireless networks, packets can be dropped because of corruption, rendering the typical reaction of a congestion control mechanism inappropriate.

TCP over wireless and satellite is a topic that has been investigated for a long time [Krishnan04]. There are some proposals where the congestion control mechanism would react as if a packet had not been dropped in the presence of corruption (cf. TCP HACK [MW1]), but discussions in the IETF have shown that there is no agreement that this type of reaction is appropriate. It has been said that congestion can manifest itself as corruption on shared wireless links, and in any case it is questionable whether a source that sends packets that are continuously impaired by link noise should keep sending at a high rate.

Generally, two questions must be addressed when designing congestion control mechanism that would take corruption into account:

1. How is corruption detected?

2. What should be the reaction?

In addition to question 1 above, it may be useful to consider detecting the reason for corruption, but this has not yet been done to the best of our knowledge.

Corruption detection can be done using an in-band or out-of-band signaling mechanism, much in the same way as described for Challenge 1. Additionally, implicit detection can be considered: link layers sometimes retransmit erroneous frames, which can cause the end-to-end delay to increase - but, from the perspective of a sender at the transport layer, there are many other possible reasons for such an effect.

Header checksums provide another implicit detection possibility: if a checksum covers all necessary headers only and this checksum does not show an error, it is possible for errors to be found in the payload using a second checksum. Such error detection is possible with UDP-Lite and DCCP, and it was found to work well over a GPRS network in a study [MW2] and poorly over a WiFi network in another study [MW3]. Note that, while UDP-Lite and DCCP enable the detection of corruption, the specifications of these protocols do not foresee any specific reaction to it for the time being.

The idea of having a transport endpoint detect and accordingly react to corruption poses a number of interesting questions regarding cross-layer interactions. As IP is designed to operate over arbitrary link layers, it is therefore difficult to design a congestion control mechanism on top of it, which appropriately reacts to corruption - especially as the specific data link layers that are in use along an end-to-end path are typically unknown to entities at the transport layer.

The IETF has not yet specified how a congestion control mechanism should react to corruption.

3.4 Challenge 4: Small Packets

With multimedia streaming flows becoming common, an increasingly large fraction of the bytes transmitted belong to control traffic. Compounding the congestion control, small packets may excessively contribute to lower network efficiency in terms of full-size packet transfer performance.

For small packets, the Nagle algorithm allows to avoid congestion collapse and pathological congestion [RFC896]. The Nagle algorithm can dramatically reduce the number of small packets. However, aggregation implies delay for packets. Applications that are jitter-sensitive typically disable the Nagle algorithm. For applications that exchange small packets, variants for the small packet to the TCP-friendly rate control (TFRC) [RFC3448] in the Datagram Congestion Control Protocol (DCCP) [RFC4340] have been designed. DCCP enables unreliable but congestion-controlled data transmission. TFRC is a congestion control mechanism for unicast flows operating in a best-effort Internet environment, and is designed for DCCP that controls the sending rate based on a stochastic Markov model for TCP Reno. Consistent with the use of end-to-end congestion control, versions of the Congestion Control Identifier (CCID) have dealt with DCCP flows that would like to receive as much bandwidth as possible over the long term (CCID 2) [RFC4241], or flows that minimize the abrupt rate changes in the sending rate (CCID 3) [RFC4242].

In its version number 4 [draft-floyd-ccid4-00.txt], CCID is being designed either to applications programs that use a small fixed segment size, or to application programs that change their sending rate by varying the segment size.

In some stable and unstable conditions, it appears that the congestion control mechanisms for small packets must be further enhanced, tightly coordinated, and controlled over wide-area networks.

3.5 Challenge 5: Pseudo-Wires

Pseudowires (PW) may carry non-TCP data flows e.g. TDM traffic. Structure Agnostic TDM over Packet (SATOP) [RFC4553], Circuit Emulation over Packet Switched Networks (CESoPSN), TDM over IP, are not responsive to congestion control in a TCP-friendly manner as prescribed by [RFC2914]. Moreover, it is not possible to simply reduce the flow rate of a TDM PW when facing packet loss.

Carrying TDM PW over an IP network poses a real problem. Indeed, providers can rate control corresponding incoming traffic but it may not be able to detect that a PW carries TDM traffic. This can be illustrated with the following example.

Sources S1, S2, S3 and S4 are originating TDM over IP traffic. P1 provider edges E1, E2, E3, and E4 are respectively rate limiting such traffic. Provider P1 SLA with transit provider P2 is such that the latter assumes a BE traffic pattern and that the distribution shows the typical properties of common BE traffic (elastic, non-real time, non-interactive).

The problem rises for transit provider P2 that is not able to detect that IP packets are carrying constant-bit rate service traffic that is by definition unresponsive to any congestion control mechanisms.

```
              ...........       ............
             .           .     .
      S1 --- E1 ---      .     .
             .     |     .     .
             .      === E5 === E7 ---
             .     |     .     .     |
      S2 --- E2 ---      .     .     |
             .           .     .     |      |
              ...........      .     |      v
                               .      ----- R --->
              ...........      .     |      ^
             .           .     .     |      |
      S3 --- E3 ---      .     .     |
             .     |     .     .     |
             .      === E6 === E8 ---
             .     |     .     .
      S4 --- E4 ---      .     .
             .           .     .
              ...........       ............

             \---- P1 ---/     \---------- P2 -----
```

Assuming P1 providers are rate limiting BE traffic, a transit P2 provider router R may be subject to serious congestion as all TDM PWs cross the same router. TCP-friendly traffic would follow existing TCP's Additive-Increase Multiplicative-Decrease (AIMD) algorithm of reducing the sending rate in half in response to each packet drop. Nevertheless, the TDM PWs will take all available capacity leaving no room for any other type of traffic. Note that the situation may simply occur because S4 suddenly turns up a TDM PW.

As it is not possible to assume that edge routers will soon have the ability to detect the type of the carried traffic, it is important for transit routers (P2 provider) to be able to apply a fair, robust, responsive and efficient congestion control technique such as to prevent impacting normal-behaving Internet traffic. However, it is still an open question how the corresponding mechanisms in data and control plane have to be designed.

3.6 Challenge 6: Multi-domain Congestion Control

Transport protocols such as TCP operate over the Internet that is divided into autonomous systems. These systems are characterized by their heterogeneity as IP networks are realized by a multitude of technologies. Variety of conditions (see also Challenge 2) and their variations leads to correlation effects between policers that regulate traffic against certain conformance criteria.

With the advent of techniques allowing for early detection of congestion, packet loss is no longer the solely metric of congestion. ECN (Explicit Congestion Notification) marks packets - set by active queue management techniques - to convey congestion information trying to prevent packet losses (packet loss and the number of packets marked gives you an indication of the level of congestion). Using TCP ACKs to feed back that information allows the hosts to realign their transmission rate and thus encourage them to efficiently use of the network. In IP, ECN uses the two unused bits of the TOS field [RFC2474]. Further, ECN in TCP uses two bits in the TCP header that were previously defined as reserved [RFC793].

ECN [RFC3168] is an example of a congestion feedback mechanism from the network toward hosts, while the policer must sit at every potential point of congestion. The congestion-based feedback scheme has, however limitations when applied inter-domain. Indeed, the same congestion feedback mechanism is required on the entire path for optimal control at end-systems.

Another solution in multi-domain environment may be the TCP rate controller (TRC), as traffic conditioner, that regulates the TCP flow at the ingress node in each domain by controlling packet drops and RTT of the packets in a flow. The outgoing traffic from a TRC controlled domain is shaped in a way that no packets are dropped at the policer. However, the TRC depends on the TCP end-to-end model, and thus the diversity of TCP implementations is a general problem.

Another challenge in multi−domain operation is security. At some domain boundaries, an increasing number of application layer gateways (e. g., proxies) is deployed, which split up end−to−end connections and prevent end−to−end congestion control. Furthermore, authentication and authorization issues can arise at domain boundaries, whenever information is exchanged, and so far the Internet does not have a single general security architecture that could be used in all cases. Many autonomous systems also only exchange some limited amount of information about their internal state (topology hiding principle), even though having more precise information could be highly beneficial for congestion control. The future evolution of the Internet inter−domain operation has to show whether more multi−domain information exchange can be realized.

3.7 Challenge 7: Precedence for Elastic Traffic

Elastic traffic initiated by so−called elastic data applications adapt to available bandwidth via a feedback control loop such as the TCP congestion control. There are two types of "as−soon−as−possible" traffic types: short−lived flows and flows with an expected average throughput. For all those flows the application dynamically adjusts the data generation rate. Examples of short−lived elastic traffic include HTTP and instant messaging traffic. Examples of average throughput requiring elastic traffic are FTP and emailing. In brief, elastic data applications can show extremely different requirements and traffic characteristics.

The idea to distinguish several classes of best−effort traffic dates is rather old, since it would be beneficial to address the relative delay sensitivities of different elastic applications. The notion of traffic precedence was introduced in [RFC791], and it was broadly defined as "An independent measure of the importance of this datagram."

For instance, low precedence traffic will experience lower average throughput than higher precedence traffic. Several questions arise, however. What is the meaning of "relative"? What is the role of the Transport Layer in providing the respective considerations for precedence wrt to serviced applicative traffic?

The preferential treatment of higher precedence traffic with appropriate congestion control mechanisms is still an open issue that may, depending on the proposed solution, impact both the host and the network precedence awareness, and thereby the congestion control.

DiffServ [RFC2474] [RFC2475] related aspects will be addressed in a future release of this document.

3.8 Challenge 8: Misbehaving Senders and Receivers

TBD.

3.9 Other challenges

TBD.

4. Security Considerations

5. Contributors

This document is the result of a collective effort to which the following people have contributed:

Dimitri Papadimitriou <Dimitri.Papadimitriou@alcatel-lucent.be>
Michael Welzl <michael.welzl@uibk.ac.at>
Wesley Eddy <weddy@grc.nasa.gov>
Bela Berde <bela.berde@gmx.de>
Paulo Loureiro <loureiro.pjg@gmail.com>
Chris Christou <christou_chris@bah.com>
Michael Scharf <michael.scharf@ikr.uni-stuttgart.de>

6. References

7.1 Normative References

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC793, September 1981.

[RFC896] Nagle, J., "Congestion Control in IP/TCP", RFC 896, January 1984.

[RFC2309] Braden, B., et al., "Recommendations on queue management and congestion avoidance in the Internet", RFC 2309, April 1998.

[RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 1633, October 1996.

[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.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[RFC2581] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", RFC 2581, April 1999.

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, September 2000.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC3448] Handley, M., Floyd, S., Padhye, J., and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, January 2003.

[RFC3985] Bryant, S. and P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.

[RFC4340] Kohler, E., Handley, M., and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, March 2006.

[RFC4341] Floyd, S. and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control", RFC 4341, March 2006.

[RFC4342] Floyd, S., Kohler, E., and J. Padhye, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC)", RFC 4342, March 2006.

[RFC4553] Vainshtein, A. and Y. Stein, "Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)", RFC 4553, June 2006.

[RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, Jan. 2007.

7.2 Informative References

[Andrew00] L. Andrew, B. Wydrowski and S. Low, "An Example of Instability in XCP", Manuscript available at < http://netlab.caltech.edu/maxnet/XCP_instability.pdf>

[Ath01] S. Athuraliya, S. Low, V. Li, and Q. Yin, "REM: Active queue management," IEEE Network Magazine, vol.15, no.3, pp. 48-53, May 2001.

[Bonald00] T. Bonald, M. May, and J.-C. Bolot, "Analytic Evaluation of RED Performance," In Proceedings of IEEE INFOCOM, Tel Aviv, Israel, March 2000.

[Clark98] D. Clark and W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service," IEEE/ACM Transactions on Networking, vol.6, no.4, pp.362-373, August 1998

[Floyd93] S. Floyd and V. Jacobson, M-^SRandom early detection gateways for congestion avoidance,M-^T IEEE/ACM Trans. on Networking, vol.1, no.4, pp. 397-413, Aug. 1993.

[Falk07] A. Falk et al "Specification for the Explicit Control Protocol (XCP)", Work in Progress, draft-falk-xcp-spec-03.txt, July 2007.

[Firoiu00] V. Firoiu and M. Borden, "A Study of Active Queue Management for Congestion Control," In Proceedings of IEEE INFOCOM, Tel Aviv, Israel, March 2000.

[Floyd94] S. Floyd, "TCP and Explicit Congestion Notification", ACM Computer Communication Review, vol.24, no.5, October 1994, pp. 10-23.

[Hollot01] C. Hollot, V. Misra, D. Towsley, and W.-B. Gong, "A Control Theoretic Analysis of RED," In Proceedings of IEEE INFOCOM, Anchorage, Alaska, April 2001.

[Jacobson88] V. Jacobson, "Congestion Avoidance and Control", Proc. of the ACM SIGCOMM ’88 Symposium, pp. 314-329, August 1988.

[Katabi02] D. Katabi, M. Handley, and C. Rohr, "Internet Congestion Control for Future High Bandwidth-Delay Product Environments", Proceedings of the ACM SIGCOMM ’02 Symposium, pp. 89-102, August 2002.

[Kelly98] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness, and stability," Journal of the Operational Research Society, vol.49, pp. 237M-^V252, 1998.

[Keshav] S. Keshav, "What is congestion and what is congestion control", Presentation at IRTF ICCRG Workshop, Pfldnet 2007, (Los Angeles), California, February 2007.

[Krishnan04] R. Krishnan, J. Sterbenz, W. Eddy, C. Partridge, and M. Allman, "Explicit Transport Error Notification (ETEN) for Error-Prone Wireless and Satellite Networks", Computer Networks, vol.46, no.3, October 2004.

[Low05] S. Low, L. Andrew and B. Wydrowski. "Understanding XCP: equilibrium and fairness", Proceedings of IEEE Infocom, Miami, USA, March 2005.

[Low03.2] S. Low, F. Paganini, J. Wang, and J. Doyle, "Linear stability of TCP/RED and a scalable control", Computer Networks Journal, vol.43, no.5, pp.633-647, December 2003.

[Low03.1] S. Low, "A duality model of TCP and queue management algorithms", IEEE/ACM Trans. on Networking, vol.11, no.4, pp.525M-^V536, August 2003.

[Low02] S. Low, F. Paganini, J. Wang, S. Adlakha, and J. C. Doyle, "Dynamics of TCP/RED and a Scalable Control", Proceedings of IEEE Infocom, New York, USA, June 2002.

[Pan00] R. Pan, B. Prabhakar, and K. Psounis, "CHOKe: a stateless AQM scheme for approximating fair bandwidth allocation", In Proceedings of IEEE Infocom, Tel Aviv, Israel, March 2000.

[Zhang03] H. Zhang, C. Hollot, D. Towsley, and V. Misra. "A Self-Tuning Structure for Adaptation in TCP/AQM Networks", SIGMETRICSM-^R03, June 10M-^V14, 2003, San Diego, California, USA.

Acknowledgments

The authors would like to thank Jan Vandenabeele for its comments on the document.

Author’s Addresses

Michael Welzl
University of Innsbruck
Technikerstr 21a
A-6020 Innsbruck, Austria
Phone: +43 (512) 507-6110
Email: michael.welzl@uibk.ac.at

Dimitri Papadimitriou
Alcatel-Lucent

Copernicuslaan, 50
B-2018 Antwerpen, Belgium
Phone : +32 3 240 8491
Email: dimitri.papadimitriou@alcatel-lucent.be

Michael Scharf
University of Stuttgart
Pfaffenwaldring 47
D-70569 Stuttgart
Germany
Phone: +49 711 685 69006
Email: michael.scharf@ikr.uni-stuttgart.de

Full Copyright Statement

Copyright (C) The Internet Society (2007).

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, THE IETF TRUST 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.

Acknowledgment

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