Transport Area Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: October 6, 2007

L. Eggert
Nokia

April 4, 2007

UDP Usage Guidelines for Application Designers
draft-eggert-tsvwg-udp-guidelines-02

Status of this Memo

Copyright Notice

Abstract

The User Datagram Protocol (UDP) provides a minimal, message-passing transport that has no inherent congestion control mechanisms. Because congestion control is critical to the stable operation of the Internet, applications and upper-layer protocols that choose to use

UDP as an Internet transport must employ mechanisms to prevent congestion collapse and establish some degree of fairness with concurrent traffic. This document provides guidelines on the use of UDP for the designers of such applications and upper-layer protocols that cover congestion-control and other topics, including message sizes and reliability.

Table of Contents

1. Introduction

The User Datagram Protocol (UDP) [RFC0768] provides a minimal, unreliable, message-passing transport to applications and upper-layer protocols (both simply called "applications" in the remainder of this document). Compared to other transport protocols, UDP is unique in that it does not establish end-to-end connections between communicating end systems. UDP communication consequently does not incur connection establishment and teardown overheads and there is no associated end system state. Because of these characteristics, UDP can offer a very efficient communication transport to some applications.

A second unique characteristic of UDP is that it provides no inherent congestion control mechanisms. [RFC2914] describes the best current practice for congestion control in the Internet. It identifies two major reasons why congestion control mechanisms are critical for the stable operation of the Internet:

1. The prevention of congestion collapse, i.e., a state where an increase in network load results in a decrease in useful work done by the network.

2. The establishment of a degree of fairness, i.e., allowing multiple flows to share the capacity of a path reasonably equitably.

Because UDP itself provides no congestion control mechanisms, it is up to the applications that use UDP for Internet communication to employ suitable mechanisms to prevent congestion collapse and establish a degree of fairness. [RFC2309] discusses the dangers of congestion-unresponsive flows and states that "all UDP-based streaming applications should incorporate effective congestion avoidance mechanisms." This is an important requirement, even for applications that do not use UDP for streaming. For example, an application that generates five 1500-byte UDP packets in one second

can already exceed the capacity of a 56 Kb/s path. For applications that can operate at higher, potentially unbounded data rates, congestion control becomes vital. Section 3 describes a number of simple guidelines for the designers of such applications.

A UDP message is carried in a single IP packet and is hence limited to a maximum payload of 65,487 bytes. The transmission of large IP packets frequently requires IP fragmentation, which decreases communication reliability and efficiency and should be avoided [I-D.heffner-frag-harmful]. Some of the guidelines in Section 3 describe how applications should determine appropriate message sizes.

This document provides guidelines to designers of applications that use UDP for unicast transmission. A special class of applications uses UDP for IP multicast transmissions. Congestion control, flow control or reliability for multicast transmissions is more difficult to establish than for unicast transmissions, because a single sender may transmit to multiple receivers across potentially very heterogeneous paths at the same time. Designing multicast applications requires expertise that goes beyond the simple guidelines given in this document. The IETF has defined a reliable multicast framework [RFC3048] and several building blocks to aid the designers of multicast applications, such as [RFC3738] or [RFC4654].

2. Terminology

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 BCP 14, RFC 2119 [RFC2119].

3. UDP Usage Guidelines

The RECOMMENDED alternative to the UDP usage guidelines described in this section is the use of a transport protocol that is congestion-controlled, such as TCP [RFC0793], SCTP [RFC2960] or DCCP [RFC4340] with its different congestion control types [RFC4341][RFC4342][I-D.floyd-dccp-ccid4]. Congestion control mechanisms are difficult to implement correctly, and for most

applications, the use of one of the existing, congestion-controlled protocols is the simplest method of satisfying [RFC2914]. The same is true for message size determination and reliability mechanisms.

If used correctly, congestion-controlled transport protocols are not as "heavyweight" as often claimed. For example, TCP with SYN cookies [I-D.ietf-tcpm-syn-flood], which are available on many platforms, does not require a server to maintain per-connection state until the connection is established. TCP also requires the end that closes a connection to maintain the TIME-WAIT state that prevents delayed segments from one connection instance to interfere with a later one. Applications that are aware of this behavior can shift maintenance of the TIME-WAIT state to conserve resources. Finally, TCP's built-in capacity-probing and PMTU awareness results in efficient data transmission that quickly compensates for the initial connection setup delay, for transfers that exchange more than a few packets.

3.1. Congestion Control Guidelines

If an application or upper-layer protocol chooses not to use a congestion-controlled transport protocol, it SHOULD control the rate at which it sends UDP messages to a destination host. It is important to stress that an application SHOULD perform congestion control over all UDP traffic it sends to a destination, independent of how it generates this traffic. For example, an application that forks multiple worker processes or otherwise uses multiple sockets to generate UDP messages SHOULD perform congestion control over the aggregate traffic. The remainder of this section discusses several approaches for this purpose.

It is important to note that congestion control should not be viewed as an add-on to a finished application. Many of the mechanisms discussed in the guidelines below require application support to operate correctly. Application designers need to consider congestion control throughout the design of their application, similar to how they consider security aspects throughout the design process.

3.1.1. Bulk Transfer Applications

Applications that perform bulk transmission of data to a peer over UDP SHOULD consider implementing TCP-Friendly Rate Control (TFRC) [RFC3448], window-based, TCP-like congestion control, or otherwise ensure that the application complies with the congestion control principles.

TFRC has been designed to provide both congestion control and fairness in a way that is compatible with the IETF's other transport protocols. TFRC is currently being updated [I-D.ietf-dccp-rfc3448bis], and application designers SHOULD always evaluate whether the latest published specification fits their needs. If an application implements TFRC, it need not follow the remaining guidelines in Section 3.1, but SHOULD still follow the guidelines on message sizes in Section 3.2 and reliability in Section 3.2.

Bulk transfer applications that choose not to implement TFRC or TCP-like windowing SHOULD implement a congestion control scheme that results in bandwidth use that competes fairly with TCP within an order of magnitude. [RFC3551] suggests that applications SHOULD monitor the packet loss rate to ensure that it is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path under the same network conditions would achieve an average throughput, measured on a reasonable timescale, that is not less than that of the UDP flow. The comparison to TCP cannot be specified exactly, but is intended as an "order-of-magnitude" comparison in timescale and throughput.

Finally, some bulk transfer applications chose not to implement any congestion control mechanism and instead rely on transmitting across reserved path capacity. This might be an acceptable choice for a subset of restricted networking environments, but is by no means a safe practice for operation in the Internet. When the UDP traffic of such applications leaks out on unprovisioned paths, results are detrimental.

3.1.2. Low Data-Volume Applications

Applications that exchange only a small number of messages with a destination at any time may not benefit from implementing TFRC or one of the other congestion control schemes in Section 3.1.1. Such applications SHOULD still control their transmission behavior by not sending more than one UDP message per round-trip time (RTT) to a

destination. Similar to the recommendation in [RFC1536], an application SHOULD maintain an estimate of the RTT for any destination it communicates with. Applications SHOULD implement the algorithm specified in [RFC2988] to compute a smoothed RTT (SRTT) estimate. A lost response from the peer SHOULD be treated as a very large RTT sample, instead of being ignored, in order to cause a sufficiently large (exponential) back-off. When implementing this scheme, applications need to choose a sensible initial value for the RTT. This value SHOULD generally be as conservative as possible for the given application. For example, SIP [RFC3261] and GIST [I-D.ietf-nsis-ntlp] use an initial value of 500 ms, and shorter values are likely problematic in many cases.

Some applications cannot maintain a reliable RTT estimate for a destination. The first case is applications that exchange too few messages with a peer to establish a statistically accurate RTT estimate. Such applications MAY use a fixed transmission interval that is exponentially backed-off during loss. For example, SIP [RFC3261] and GIST [I-D.ietf-nsis-ntlp] use an interval of 500 ms, and shorter values are likely problematic in many cases.

A second class of applications cannot maintain an RTT estimate for a destination, because the destination does not send return traffic. Such applications SHOULD NOT send more than one UDP message every 3 seconds. The 3-second interval was chosen based on TCP's retransmission timeout when the RTT is unknown [RFC2988], and shorter values are likely problematic in many cases. Note that this interval must be more conservative than above, because the lack of return traffic prevents the detection of packet loss, i.e., congestion events, and the application therefore cannot perform exponential back-off to reduce load.

Applications that communicate bidirectionally SHOULD employ

congestion control for both directions of the communication. For example, for a client-server, request-response-style application, clients SHOULD congestion control their request transmission to a server, and the server SHOULD congestion control its responses to the clients. Congestion in the forward and reverse direction is uncorrelated and an application SHOULD independently detect and respond to congestion along both directions.

3.2. Message Size Guidelines

Because IP fragmentation lowers the efficiency and reliability of Internet communication [I-D.heffner-frag-harmful], an application SHOULD NOT send UDP messages that result in IP packets that exceed the MTU of the path to the destination. Consequently, an application SHOULD either use the path MTU information provided by the IP layer or implement path MTU discovery itself [RFC1191][RFC1981][RFC4821] to determine whether the path to a destination will support its desired message size without fragmentation.

Applications that choose not to do so SHOULD NOT send UDP messages that exceed the minimum PMTU. The minimum PMTU depends on the IP version used for transmission, and is the lesser of 576 bytes and the first-hop MTU for IPv4 [RFC1122] and 1280 bytes for IPv6 [RFC2460]. To determine an appropriate UDP payload size, applications must subtract IP header and option lengths as well as the length of the UDP header from the PMTU size. Transmission of minimum-sized messages is inefficient over paths that support a larger PMTU, which is a second reason to implement PMTU discovery.

Applications that do not send messages that exceed the minimum PMTU of IPv4 or IPv6 need not implement any of the above mechanisms.

3.3. Reliability Guidelines

Application designers are generally aware that UDP does not provide any reliability. Often, this is a main reason to consider UDP as a transport. Applications that do require reliable message delivery SHOULD implement an appropriate mechanism themselves.

UDP also does not protect against message duplication, i.e., an application may receive multiple copies of the same message. Application designers SHOULD consider whether their application handles message duplication gracefully, and may need to implement mechanisms to detect duplicates. Even if message reception triggers idempotent operations, applications may want to suppress duplicate messages to reduce load.

Finally, UDP messages may be reordered in the network and arrive at

the receiver in an order different from the send order. Applications

that require ordered delivery SHOULD reestablish message ordering themselves.

4. Security Considerations

[RFC2309] and [RFC2914] discuss the dangers of congestion-unresponsive flows to the Internet. This document provides guidelines to designers of UDP-based applications to congestion-control to their transmissions. As such, it does not raise any additional security concerns.

5. IANA Considerations

This document raises no IANA considerations.

6. Acknowledgments

Thanks to Mark Allman, Gorry Fairhurst, Sally Floyd, Joerg Ott, Colin Perkins, Pasi Sarolahti and Magnus Westerlund for their comments on this document.

7. References

7.1. Normative References

[RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

[RFC2960] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L., and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[RFC2988] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, November 2000.

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

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

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

7.2. Informative References

[I-D.floyd-dccp-ccid4]
Floyd, S. and E. Kohler, "Profile for Datagram Congestion Control Protocol (DCCP) Congestion ID 4: TCP-Friendly Rate Control for Small Packets (TFRC-SP)", draft-floyd-dccp-ccid4-00 (work in progress), November 2006.

[I-D.heffner-frag-harmful]
Heffner, J., "IPv4 Reassembly Errors at High Data Rates", draft-heffner-frag-harmful-04 (work in progress), January 2007.

[I-D.ietf-dccp-rfc3448bis]
Handley, M., "TCP Friendly Rate Control (TFRC): Protocol Specification", draft-ietf-dccp-rfc3448bis-01 (work in progress), March 2007.

[I-D.ietf-nsis-ntlp]
Schulzrinne, H. and R. Hancock, "GIST: General Internet Signalling Transport", draft-ietf-nsis-ntlp-13 (work in progress), April 2007.

[I-D.ietf-tcpm-syn-flood]
Eddy, W., "TCP SYN Flooding Attacks and Common Mitigations", draft-ietf-tcpm-syn-flood-02 (work in progress), March 2007.

[RFC1122] Braden, R., "Requirements for Internet Hosts -

Communication Layers", STD 3, RFC 1122, October 1989.

[RFC1536] Kumar, A., Postel, J., Neuman, C., Danzig, P., and S. Miller, "Common DNS Implementation Errors and Suggested Fixes", RFC 1536, October 1993.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC3048] Whetten, B., Vicisano, L., Kermode, R., Handley, M., Floyd, S., and M. Luby, "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer", RFC 3048, January 2001.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[RFC3738] Luby, M. and V. Goyal, "Wave and Equation Based Rate Control (WEBRC) Building Block", RFC 3738, April 2004.

[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.

[RFC4654] Widmer, J. and M. Handley, "TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification", RFC 4654, August 2006.

Author's Address

Lars Eggert
Nokia Research Center
P.O. Box 407
Nokia Group 00045
Finland

Phone: +358 50 48 24461
Email: lars.eggert@nokia.com
URI: http://research.nokia.com/people/lars_eggert/

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).