Services provided by IETF transport protocols and congestion control mechanisms
draft-fairhurst-taps-transports-00

Abstract

This document describes services provided by existing IETF protocols and congestion control mechanisms. It is designed to help application and network stack programmers and to inform the work of the IETF TAPS Working Group.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 30, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of
Internet-Draft TAPS transport protocols and CC mechanisms   October 2014

the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

1. Introduction

Most Internet applications make use of the Transport Services
provided by TCP (a reliable, in-order stream protocol) or UDP (an
unreliable datagram protocol). We use the term "Transport Service"
to mean an end-to-end facility provided by the transport layer. That
service can only be provided correctly if information is supplied
from the application. The application may determine the information
to be supplied at design time, compile time, or run time and may
include guidance on whether an aspect of the service is required, a
preference by the application, or something in between. Examples of
Transport service facilities are reliable delivery, ordered delivery,
content privacy to in-path devices, integrity protection, and minimal
latency.

Transport protocols such as SCTP, DCCP, MPTCP, UDP and UDP-Lite have
been defined at the transport layer.

In addition, a transport service may be built on top of these
transport protocols, using a framework such as WebSockets, or RTP.
Service built on top of UDP or UDP-Lite typically also need to
specify a congestion control mechanism, such as TFRC or the LEDBAT
congestion control mechanism. This extends the set of available
Transport Services beyond those provided to applications by TCP and UDP.

Transport services can also be differentiated by the services they
provide: for instance, SCTP offers a message-based service that does
not suffer head-of-line blocking when used with multiple stream,
because it can accept blocks of data out of order, UDP-Lite provides
partial integrity protection when used over link-layer services that
can support this, and LEDBAT can provide low-priority "scavenger"
communication.

2. Terminology

This section presents the terminology used in this document.

[EDITOR’S NOTE: Terminology to be discussed in Honolulu. We need to
determine what a "service" as used by the IETF, as opposed to a
"service component", "property", an "aspect", "dimension", etc.]
3. Transport Protocols

This section provides a list of known IETF transport protocol and transport protocol frameworks.

[EDITOR’S NOTE: combine these tables into one? Also, reorder them to match the sections below.]

<table>
<thead>
<tr>
<th>Section</th>
<th>Benefit</th>
<th>Setup</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Transmission Control Protocol (TCP)</td>
<td>CO</td>
<td>Unicast</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Multipath-TCP (MPTCP)</td>
<td>CO</td>
<td>Unicast</td>
</tr>
<tr>
<td>3.2</td>
<td>SCTP</td>
<td>CO</td>
<td>Unicast</td>
</tr>
<tr>
<td>3.2.1</td>
<td>SCTP-PR</td>
<td>CO</td>
<td>Unicast</td>
</tr>
<tr>
<td>3.3</td>
<td>User Datagram Protocol (UDP)</td>
<td>DG</td>
<td>Unicast/Multicst</td>
</tr>
<tr>
<td>3.4</td>
<td>UDP-Lite</td>
<td>DG</td>
<td>Unicast/Multicst</td>
</tr>
<tr>
<td>3.5</td>
<td>DCCP</td>
<td>CO</td>
<td>Unicast</td>
</tr>
<tr>
<td>3.X</td>
<td>More as needed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Key IETF Transport Protocol - by communication mode

<table>
<thead>
<tr>
<th>Section</th>
<th>Benefit</th>
<th>Style</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Transmission Control Protocol (TCP)</td>
<td>Str</td>
<td>Ordered Byte Stream</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Multipath-TCP (MPTCP)</td>
<td>Str</td>
<td>Ordered Byte Stream</td>
</tr>
<tr>
<td>3.2</td>
<td>SCTP</td>
<td>Mess</td>
<td>Message Streams</td>
</tr>
<tr>
<td>3.2.1</td>
<td>SCTP-PR</td>
<td>Mess</td>
<td>Partial M Streams</td>
</tr>
<tr>
<td>3.3</td>
<td>User Datagram Protocol (UDP)</td>
<td>Mess</td>
<td>Datagram Message</td>
</tr>
<tr>
<td>3.4</td>
<td>UDP-Lite</td>
<td>Mess</td>
<td>Error Tolerant DG</td>
</tr>
<tr>
<td>3.5</td>
<td>DCCP</td>
<td>Mess</td>
<td>Unrel Message Stream</td>
</tr>
<tr>
<td>3.X</td>
<td>More as needed</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Key IETF Transport Protocol - by reliability

"Setup" defines whether the protocol performs a connection-oriented protocol handshake prior to communication or is datagram-based. This provides reliable negotiation of options, including negotiation of a suitable congestion control mechanism. This property can impact the ability of the protocol to traverse firewalls.
### Table 3: Key IETF Transport Protocol - by congestion control

<table>
<thead>
<tr>
<th>Section</th>
<th>Benefit</th>
<th>Congestion Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Transmission Control Protocol (TCP)</td>
<td>Yes</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Multipath-TCP (MPTCP)</td>
<td>Yes (Multipath)</td>
</tr>
<tr>
<td>3.2</td>
<td>SCTP</td>
<td>Yes</td>
</tr>
<tr>
<td>3.2.1</td>
<td>SCTP-PR</td>
<td>Yes</td>
</tr>
<tr>
<td>3.3</td>
<td>User Datagram Protocol (UDP)</td>
<td>At application layer</td>
</tr>
<tr>
<td>3.4</td>
<td>UDP-Lite</td>
<td>At application layer</td>
</tr>
<tr>
<td>3.5</td>
<td>DCCP</td>
<td>Yes, Various CCIDs defined</td>
</tr>
<tr>
<td>3.X</td>
<td>More as needed</td>
<td></td>
</tr>
</tbody>
</table>

Some other protocol frameworks that may potentially be considered for inclusion in future versions of this document. Examples are:

- Multicast - RMT
- RTP-based methods
- HTTP-based methods
- TLS
- DTLS

The following subsections describes each of these transports.

#### 3.1. Transport Control Protocol (TCP)

TCP provides a bidirectional byte-oriented stream over a connection-oriented protocol. The protocol and API use the byte-stream model.

[EDITOR’S NOTE: Describe the aspects(?) of TCP: reliable, connection-oriented, congestion-controlled, single-stream-oriented, non-boundary-preserving... Note that we want to describe the characteristics of the SOCK_STREAM API as well as just the wire protocol.]

#### 3.1.1. Multipath TCP (MPTCP)

[EDITOR’S NOTE: aspects of MPTCP beyond TCP.]
3.2. Stream Control Transmission Protocol (SCTP)

This section will describe SCTP.

SCTP provides a bidirectional set of logical unicast streams over one connection-oriented protocol. The protocol and API use messages, rather than a byte-stream. Each stream of messages is independently managed, therefore retransmission does not hold back data sent using other logical streams.

3.2.1. Partial Reliability SCTP (PR-CTP)

SCTP-PR [RFC3758] is a variant of SCTP that provides partial reliability.

3.3. User Datagram Protocol (UDP)

The User Datagram Protocol (UDP) provides a unidirectional minimal message-passing transport that has no inherent congestion control mechanisms. The service may be multicast and/or unicast.

[EDITOR’S NOTE: Describe the aspects(?) of UDP: unreliable, congestion control to be applied above the transport, datagram-oriented, connectionless, boundary-preserving... Note that we want to describe the characteristics of the SOCK_DGRAM API as well as just the wire protocol.]

Using UDP robustly requires each application to implement a raft of functions (mostly re-inventing or adapting mechanisms already found in TCP, SCTP and DCCP). [EDITOR’S NOTE: reference RFC 5405/bis]

3.4. UDP-Lite

A special class of applications can derive benefit from having partially-damaged payloads delivered, rather than discarded, when using paths that include error-prone links. Such applications can tolerate payload corruption and may choose to use the Lightweight User Datagram Protocol (UDP-Lite) The service may be multicast and/or unicast.

[EDITOR’S NOTE: compare to UDP]

[RFC3828] and [RFC 5405/bis]
3.5. Datagram Congestion Control Protocol (DCCP)

The Datagram Congestion Control Protocol (DCCP) [RFC4340] is a bidirectional transport protocol that provides unicast connections of congestion-controlled unreliable messages. DCCP is suitable for applications that transfer fairly large amounts of data and that can benefit from control over the tradeoff between timeliness and reliability.

[EDITOR’S NOTE: Describe the aspects (?) of DCCP...]

[FC4340 et al]

3.6. Realtime Transport Protocol (RTP)

RTP provides an end-to-end network transport service, suitable for applications transmitting real-time data, such as audio, video or data, over multicast or unicast network services, including TCP, UDP, UDP-Lite, DCCP.

[EDITOR’S NOTE: Describe the aspects (?) of RTP...]

3.7. Hypertext Transport Protocol (HTTP) as a pseudotransport

HTTP provides end-to-end network unicast transport service.

[EDITOR’S NOTE: Reference BCP 56, note that this implies TCP but also brings with it object semantics you may not want.]

3.7.1. WebSockets

[EDITOR’S NOTE: point out how websockets kind of fixes this.]

4. Transport service components

Aspects as derived from the subsections above.

This section is blank for now.

5. Acknowledgements

The authors were part-funded by the European Community under its Seventh Framework Programme. The views expressed are solely those of the authors.

Comments are welcome to the authors or via the IETF TAPS mailing lists.
6. IANA Considerations

XXX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

7. Security Considerations

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

8. References

8.1. Normative References


8.2. Informative References


Authors’ Addresses

Godred Fairhurst
University of Aberdeen
School of Engineering, Fraser Noble Building
Aberdeen AB24 3UE
UK

Email: gorry@erg.abdn.ac.uk

Brian Trammell
ETH Zurich
Gloriastrasse 35
Zurich 8092
CH

Email: ietf@trammell.ch