

TAPS  
Internet-Draft  
Intended status: Informational  
Expires: December 24, 2015

S. Gjessing  
M. Welzl  
University of Oslo  
June 22, 2015

**A Minimal Set of Transport Services for TAPS Systems**  
**draft-gjessing-taps-minset-00**

Abstract

This draft will eventually recommend a minimal set of IETF Transport Services offered by end systems supporting TAPS, and give guidance on choosing among the available mechanisms and protocols. As a starting point for discussion, it currently only gives an overview of some ways to categorize the set of transport services in the first TAPS document (version 4: [draft-ietf-taps-transport-04](#)), assuming that the eventual minimal set of transport services will be based on a similar form of categorization.

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 December 24, 2015.

Copyright Notice

Copyright (c) 2015 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 the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Terminology (as defined by <a href="#">draft-ietf-taps-transport-04</a> ) . .	<a href="#">4</a>
<a href="#">3.</a>	A list of all features in the considered IETF Transport Protocols . . . . .	<a href="#">5</a>
<a href="#">4.</a>	Grouping of the features . . . . .	<a href="#">6</a>
<a href="#">4.1.</a>	Functional vs. non-functional . . . . .	<a href="#">7</a>
<a href="#">4.1.1.</a>	Functional features . . . . .	<a href="#">7</a>
<a href="#">4.1.2.</a>	Non-functional features . . . . .	<a href="#">7</a>
<a href="#">4.2.</a>	Components from <a href="#">draft-ietf-taps-transport-04</a> that are not specified or seen by the applications . . . . .	<a href="#">9</a>
<a href="#">5.</a>	Acknowledgements . . . . .	<a href="#">9</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">10</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">10</a>
<a href="#">8.</a>	Normative References . . . . .	<a href="#">10</a>
	Authors' Addresses . . . . .	<a href="#">10</a>



## 1. Introduction

An application has an intended usage and demands for transport services, and the task of any system that implements TAPS is to offer these services to its applications, i.e. the applications running on top of TAPS, without binding an application to a particular transport protocol.

The present draft is based on [\[TAPS1\]](#) and follows the same terminology (also listed below). We include an "inversion" (in the database sense) of [\[TAPS1\]](#), in that, based on the lists of protocol components in [\[TAPS1\]](#) we list all protocol features, and for each feature, we list the Transport Protocols that support this feature (as a component). The resulting list is very long. If the list of Transport Services that a TAPS system offers to applications was a simple copy of this list, the resulting system would be very hard to use. It is therefore necessary to minimize the number of services that are offered. We begin this by grouping these transport features.

The groups of features offered to applications are divided as follows:

1. functional vs. non-functional
2. static vs. initialization vs. dynamic
3. single-sided vs. both-sided

Because QoS is out of scope of TAPS, this document assumes a "best effort" service model [\[RFC5290, RFC7305\]](#). Applications using a TAPS system can therefore not make any assumptions about e.g. the time it will take to send a message. There are however certain requirements that are strictly kept by transport protocols today, and a TAPS system.

Functional features use components that cannot be used without the application knowing about them, or else they violate assumptions that might cause the application to break. Components implementing non-functional features may be used without involving the application. For example, unordered message delivery is a functional feature: it cannot be used without the application knowing about it because the application's assumption could be that messages arrive in-order, and in this case unordered delivery could cause the application to break. Multihoming and data bundling (Nagle in TCP) are non-functional features: if a TAPS system autonomously decides to enable or disable them, an application will not break (but a TAPS system may be able to communicate more efficiently if the application is in control of this feature).

If a transport protocol offers a feature that can not be changed or



opted out, this feature is called static in this protocol. An application uses a static feature either because it has requested it (and TAPS decide to fulfill this request by a protocol that has a component that implements this feature as static), or the static feature is offered to the application implicitly because TAPS chooses a protocol that implements it. For example, if an application chooses byte-stream-oriented delivery, it automatically also gets reliable delivery.

Initialization features can be chosen when communication begins but not adjusted later; this assumes that a TAPS system does not change protocols during a communication session. Examples of initialization features are flow control and congestion control.

Dynamic features are changeable during runtime. An example of a dynamic feature is data bundling (Nagle in TCP).

Single-sided features can be provided via components that are implemented only on the side where they applications requests the Transport Service. An example of a single-sided feature is data bundling (Nagle in TCP). Both-sided features can only be provided via components that are implemented on both sides. An example is error detection (checksum). Possibly certain features could benefit from, but do not need to be, implemented on both sides. Since the point of categorization is to determine the minimal set of Transport Services that a TAPS system must provide, the essential property of such features is that they *\*can\** be implemented on only one side.

## 2. Terminology (as defined by [draft-ietf-taps-transports-04](#))

The following terms are defined throughout this document, and in subsequent documents produced by TAPS describing the composition and decomposition of transport services.

Transport Service Feature: a specific end-to-end feature that a transport service provides to its clients. Examples include confidentiality, reliable delivery, ordered delivery, message-versus-stream orientation, etc.

Transport Service: a set of transport service features, without an association to any given framing protocol, which provides a complete service to an application.

Transport Protocol: an implementation that provides one or more different transport services using a specific framing and header format on the wire.



Transport Protocol Component: an implementation of a transport service feature within a protocol.

Transport Service Instance: an arrangement of transport protocols with a selected set of features and configuration parameters that implements a single transport service, e.g. a protocol stack (RTP over UDP).

Application: an entity that uses the transport layer for end-to-end delivery data across the network (this may also be an upper layer protocol or tunnel encapsulation).

### **3. A list of all features in the considered IETF Transport Protocols**

[TAPS1] provides a list of known IETF transport protocols and transport protocols frameworks. Here all features from [TAPS1] are listed:

- o unicast: TCP SCTP UDP-Lite DCCP NORM
- o IPv6 multicast and anycast: UDP
- o IPv4 broadcast, multicast and anycast: UDP UDP-Lite
- o multicast: NORM
  
- o unidirectional: UDP
- o bidirectional: TCP
  
- o message-oriented delivery: SCTP UDP UDP-Lite DCCP
- o byte-stream-oriented delivery: TCP
  
- o user message fragmentation and reassembly: SCTP
- o IPv6 jumbograms: UDP
  
- o connection setup with feature negotiation and application-to-port mapping: TCP SCTP DCCP
  
- o port multiplexing: TCP SCTP UDP UDP-Lite DCCP
- o port multiplexing (UDP ports): NORM
- o 2-tuple endpoints: UDP
  
- o transport layer multihoming for resilience: SCTP
  
- o transport layer mobility: SCTP
  
- o non-reliable delivery: UDP-Lite DCCP
- o reliable delivery: TCP NORM
- o reliable or partially reliable delivery: SCTP
- o drop notification: DCCP





- o ordered delivery: DCCP
- o ordered delivery for each byte stream: TCP
- o ordered delivery for each byte or message stream: NORM
- o ordered and unordered delivery within a stream (of messages): SCTP
- o unordered delivery: UDP-Lite UDP
- o unordered delivery of in-memory data or file bulk content objects: NORM
- o object-oriented delivery of discrete data or file items: NORM
  
- o partial integrity protection: UDP-Lite DCCP
- o checksum optional: UDP
- o error detection (checksum): TCP UDP
- o error detection (UDP checksum): NORM
- o strong error detection (CRC32C): SCTP
- o packet erasure coding (both proactively and as part of ARQ): NORM
  
- o flow control: TCP SCTP
- o flow control (slow receiver function): DCCP
- o flow control (timer-based and/or ack-based): NORM
  
- o segmentation: TCP NORM
  
- o stream-oriented delivery in a single stream: TCP NORM
- o support for multiple concurrent streams: SCTP
- o support for stream scheduling prioritization: SCTP
  
- o data bundling (Nagle's algorithm): TCP NORM
- o user message bundling: SCTP
  
- o congestion control: TCP SCTP NORM
- o no congestion control: UDP
  
- o timestamps: DCCP

#### **4. Grouping of the features**

This section presents a grouping of the features from [[TAPS1](#)] according to the three categories: functional vs. non-functional; static vs. initialization vs. dynamic; one-sided vs. two-sided.

The current version only includes the functional vs. non-functional categorization. The other categories will be included in future versions.



## **4.1. Functional vs. non-functional**

### **4.1.1. Functional features**

What is delivered (delivery type):

- o message-oriented delivery: SCTP UDP UDP-Lite DCCP
- o byte-stream-oriented delivery: TCP
- o object-oriented delivery of discrete data or file items: NORM

Reliability of delivery:

- o non-reliable delivery: UDP UDP-Lite DCCP
- o reliable delivery: TCP NORM
- o reliable or partially reliable delivery: SCTP
- o drop notification: DCCP

Direction of communication:

- o unidirectional: UDP
- o bidirectional: TCP

Ordered or unordered delivery:

- o ordered delivery: DCCP
- o ordered delivery for each byte stream: TCP
- o ordered delivery for each byte or message stream: NORM
- o ordered and unordered delivery within a stream (of messages): SCTP
- o unordered delivery: UDP-Lite UDP
- o unordered delivery of in-memory data or file bulk content objects:  
NORM

Unicast, anycast, multicast or broadcast:

- o unicast: TCP SCTP UDP UDP-Lite DCCP NORM
- o IPv6 multicast and anycast: UDP
- o IPv4 broadcast, multicast and anycast: UDP UDP-Lite
- o multicast: NORM

### **4.1.2. Non-functional features**

These are features that the application can optionally specify. If a feature is not specified by the application it is undefined, i.e. the TAPS system may choose an implementation of any of the features



listed.

Congestion control:

- o congestion control: TCP SCTP NORM
- o no congestion control: UDP

Resilience:

- o multihoming: SCTP
- o no resilience: UDP UDP-lite TCP

Connection oriented (or not):

- o connection oriented: TCP DCCP SCTP
- o not connection oriented: UDP UDP-Lite

Message sizes and fragmentation:

- o user message fragmentation and reassembly: SCTP
- o IPv6 jumbograms: UDP

Setup negotiation:

- o connection setup with feature negotiation and application-to-port mapping: TCP SCTP DCCP

Flow control:

- o flow control: TCP SCTP
- o flow control (slow receiver function): DCCP
- o flow control (timer-based and/or ack-based): NORM
- o no flow control: UDP

Multiplexing:

- o multistreaming: SCTP
- o port multiplexing: TCP SCTP UDP UDP-Lite DCCP
- o port multiplexing (UDP ports): NORM
- o 2-tuple endpoints: UDP



Transport layer mobility:

- o transport layer mobility: SCTP

Error detection, protection, FEC and integrity (divided into more groups?):

- o partial integrity protection: UDP-Lite DCCP
- o checksum optional: UDP
- o error detection (checksum): TCP UDP
- o error detection (UDP checksum): NORM
- o strong error detection (CRC32C): SCTP
- o packet erasure coding (both proactively and as part of ARQ): NORM
- o content privacy to in-path devices: ? (intro of [[TAPS1](#)])

Streams:

- o stream-oriented delivery in a single stream: TCP NORM
- o support for multiple concurrent streams: SCTP
- o support for stream scheduling prioritization: SCTP

Bundling:

- o data bundling (Nagle's algorithm): TCP NORM
- o user message bundling: SCTP

#### **4.2. Components from [draft-ietf-taps-transports-04](#) that are not specified or seen by the applications**

Segmentation:

- o segmentation: TCP NORM

Timestamps:

- o timestamps: DCCP
- o Service Codes: DCCP

## **5. Acknowledgements**

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 644334 (NEAT). The views expressed are solely those of the author(s).





## **6. IANA Considerations**

XX RFC ED - PLEASE REMOVE THIS SECTION XXX

This memo includes no request to IANA.

## **7. Security Considerations**

Security will be considered in future versions of this document.

## **8. Normative References**

[TAPS1] Fairhurst, G., Trammell, B., and M. Kuehlewind, "Services provided by IETF transport protocols and congestion control mechanisms", [draft-ietf-taps-transports-04](#) (work in progress), May 2015.

### Authors' Addresses

Stein Gjessing  
University of Oslo  
PO Box 1080 Blindern  
Oslo, N-0316  
Norway

Phone: +47 22 85 24 44  
Email: [steing@ifi.uio.no](mailto:steing@ifi.uio.no)

Michael Welzl  
University of Oslo  
PO Box 1080 Blindern  
Oslo, N-0316  
Norway

Phone: +47 22 85 24 20  
Email: [michawe@ifi.uio.no](mailto:michawe@ifi.uio.no)

