YANG Groupings for Transmission Control Protocol (TCP) Configuration
draft-scharf-tcpm-yang-tcp-02

Abstract

This document specifies a YANG model for TCP on devices that are configured by network management protocols. The YANG model defines groupings for fundamental parameters that can be modified in many TCP implementations. The model extends a base model for TCP clients and servers [I-D.ietf-netconf-tcp-client-server].

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 https://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 January 8, 2020.

Copyright Notice

Copyright (c) 2019 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 (https://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
1. Introduction

The Transmission Control Protocol (TCP) [RFC0793] is used by many applications in the Internet, including control and management protocols. Therefore, TCP is implemented on network elements that can be configured via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. This document specifies a YANG model [RFC6020][RFC7950] for configuring TCP on network elements that support YANG data models. This document extends a base model for TCP clients and servers [I-D.ietf-netconf-tcp-client-server]. The model focuses on fundamental and standard TCP functions that are widely implemented. The model can be augmented to address more advanced or implementation-specific TCP features. Operational state and statistics are outside the scope of this memo.

Many protocol stacks on Internet hosts use other methods to configure TCP, such as operating system configuration or policies. Many TCP/IP stacks cannot be configured by network management protocols such as NETCONF or RESTCONF and they do not use YANG data models. Yet, such TCP implementations often also have means to configure the parameters listed in this document. All parameters defined in this document are optional.

This specification is orthogonal to a Management Information Base (MIB) for the Transmission Control Protocol (TCP) that has been standardized [RFC4022]. A MIB providing extended statistics for TCP...
is also available [RFC4898], and there are also MIBs for UDP [RFC4113] and SCTP [RFC3873]. It is possible to translate a MIB into a YANG model, for instance using the translation described in [RFC6643]. However, this approach is not used in this document, as such a translated model would not be up-to-date.

There are also other related YANG models. Examples are:

- Application protocol models may include TCP parameters, for example in case of BGP [I-D.ietf-idr-bgp-model].
- TCP header attributes are modeled in other models, such as [I-D.ietf-netmod-acl-model].
- TCP-related configuration of a NAT is defined in [I-D.ietf-opsawg-nat-yang].

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Model Overview

3.1. Modeling Scope

TCP is implemented on many different system architectures. As a result, there are many different and often implementation-specific ways to configure parameters of the TCP protocol engine. In addition, in many TCP/IP stacks configuration exists for different scopes:

- Global configuration: Many TCP implementations have configuration parameters that affect all TCP connections. Typical examples include the enabling or disabling optional protocol features.
- Interface configuration: It can be useful to use different TCP parameters on different interfaces, e.g., different device ports or IP interfaces. In that case, TCP parameters can be part of the interface configuration. Typical examples are the Maximum Segment Size (MSS) or configuration related to hardware offloading.
- Connection parameters: Many implementations have means to influence the behavior of each TCP connection, e.g., on the programming interface used by applications. A typical example are
socket options in the socket API, such as disabling the Nagle algorithm by TCP_NODELAY. In an application uses such an interface, it is possible that the configuration of the application or application protocol includes TCP-related parameters. An example is the YANG model for BGP configuration [I-D.ietf-idr-bgp-model].

- Policies: Setting of TCP parameters can also be part of system policies, templates, or profiles. An example would be the preferences defined in the TAPS interface [I-D.ietf-taps-interface].

There is no ground truth for setting certain TCP parameters, and traditionally different implementation have used different modeling approaches. For instance, one implementation may define a given configuration parameter globally, while another one uses per-interface settings, and both approaches work well for the corresponding use cases. Also, different systems may use different default values.

In addition to configuration of the TCP protocol engine, a TCP implementation typically also offers access to operational state and statistics. This includes amongst others:

- Statistics: Counters for the number of active/passive opens, sent and received segments, errors, and possibly other detailed debugging information
- TCP connection table: Access to status information for all TCP connections
- TCP listener table: Information about all TCP listening endpoints

This document focuses solely on modeling basic TCP configuration state. Operational state (see [RFC8342]) is outside the scope of this specification.

The YANG model defined in this document extends a base model for TCP clients and servers [I-D.ietf-netconf-tcp-client-server]. Similar to the base model, this specification only defines YANG groupings. This allows reuse of these groupings in different YANG data models. It is intended that these groupings will be used either standalone or for TCP-based protocols as part of a stack of protocol-specific configuration models.
3.2. Basic TCP Configuration Parameters

There are a number of basic system parameters that are configurable on many TCP implementations, even if not all TCP implementations may indeed have exactly all these settings. Also, the syntax, semantics and scope (e.g., global or interface-specific) can be different in different system architectures.

The following list of fundamental parameters considers both TCP implementations on hosts and on routers:

- Keepalives (see also [I-D.ietf-netconf-tcp-client-server])
  * Idle-time (in seconds): integer
  * Probe-interval (in seconds): integer
  * Max-probes: integer
- Maximum MSS (in byte): integer
- FIN timeout (in seconds): integer
- SACK (disable/enable): boolean
- Timestamps (disable/enable): boolean
- Path MTU Discovery (disable/enable): boolean
- ECN
  * Enabling (disable/passive/active): enumeration

Some other parameters are also common but not ubiquitously supported, or modeled in very different ways. Therefore, the following attributes are not considered in this document:

- Delayed ACK timeout (in ms)
- Initial RTO value (in ms)
- Maximum number of retransmissions
- Window scaling
- Maximum number of connections
TCP can be implemented in different ways and design choices by the protocol engine often affect configuration options. In a number of areas there are major differences between different software architectures. As a result, there are not many commonalities in the corresponding configuration parameters:

- **Window size**: TCP stacks can either store window state variables (such as the congestion window) in segments or in bytes.

- **Buffer sizes**: The memory management depends on the operating system. As the size of buffers can vary over several orders of magnitude, very different implementations exist. This typically influences TCP flow control.

- **Timers**: Timer implementation is another area in which TCP stacks may differ.

- **Congestion control algorithms**: Many congestion control algorithms have configuration parameters, but except for fundamental properties they often tie into the specific implementation.

This document only models fundamental system parameters that are configurable on many TCP implementations, and for which the configuration is reasonably similar.

### 3.3. Model Design

[Editor’s node: This section requires further work.]

This document extends the YANG model "ietf-tcp-common" defined in [I-D.ietf-netconf-tcp-client-server]. The exact modeling is TBD. The intention is to define YANG groupings for all parameters so that they can be used in different YANG models.

As an example, enabling the support of Selective Acknowledgements (SACK) can be modelled as follows:

```yang
grouping tcp-sack-grouping {
  description "Support of Selective Acknowledgements (SACK)";

  leaf sack {
    type boolean;
    default "true";
    description "Enable support of Selective Acknowledgements (SACK)";
  }
}
```
A YANG model could then, for instance, import the YANG model "ietf-tcp-common" as well as the model defined in this document as follows:

```yang
grouping example-tcp-config {
    description "Example TCP stack configuration";
    uses tcp-common-grouping;
    uses tcp-sack-grouping;
}
```

3.4.  Tree Diagram

This section provides a tree diagram [RFC8340] for the YANG module defined in this document.

4.  TCP Configuration YANG Model

5.  IANA Considerations

6.  Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.
7. References

7.1. Normative References

[I-D.ietf-netconf-tcp-client-server]


7.2. Informative References

[I-D.ietf-idr-bgp-model]

[I-D.ietf-netmod-acl-model]

[I-D.ietf-opsawg-nat-yang]

[I-D.ietf-taps-interface]


Internet-Draft            TCP Configuration            July 2019

           the Transmission Control Protocol (TCP)", RFC 4022,
           DOI 10.17487/RFC4022, March 2005,

[RFC4113]  Fenner, B. and J. Flick, "Management Information Base for
           the User Datagram Protocol (UDP)", RFC 4113,
           DOI 10.17487/RFC4113, June 2005,

           Extended Statistics MIB", RFC 4898, DOI 10.17487/RFC4898,

[RFC6643]  Schoenwaelder, J., "Translation of Structure of Management
           Information Version 2 (SMIv2) MIB Modules to YANG
           Modules", RFC 6643, DOI 10.17487/RFC6643, July 2012,

Appendix A.  Acknowledgements

Michael Scharf is supported by the StandICT.eu project, which is
funded by the European Commission under the Horizon 2020 Programme.

Appendix B.  Changes compared to previous versions

Changes compared to draft-scharf-tcpm-yang-tcp-01
  o Alignment with [I-D.ietf-netconf-tcp-client-server]
  o Removing backward-compatibility to the TCP MIB
  o Additional co-author

Changes compared to draft-scharf-tcpm-yang-tcp-00
  o Editorial improvements

Authors' Addresses

Michael Scharf
Hochschule Esslingen - University of Applied Sciences
Flandernstr. 101
Esslingen  73732
Germany

Email: michael.scharf@hs-esslingen.de