TCPM   M. Scharf
Internet-Draft   Hochschule Esslingen
Intended status: Standards Track   M. Jethanandani
Expires: 28 April 2022   Kloud Services
                                                                V. Murgai
                                                                  Samsung
                                                          25 October 2021

A YANG Model for Transmission Control Protocol (TCP) Configuration

draft-ietf-tcpm-yang-tcp-04

Abstract

This document specifies a minimal YANG model for TCP on devices that are configured by network management protocols. The YANG model defines a container for all TCP connections and groupings of authentication parameters that can be imported and reused in TCP implementations or by other models that need to configure TCP parameters. The model also includes basic TCP statistics. The model is compliant with Network Management Datastore Architecture (NMDA) (RFC 8342)~~ compliant~~.

Commenté [BMI1]: To be consistent with rfc8407#section-4.13

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Transmission Control Protocol (TCP) ~~Specification~~specification [I-D.ietf-tcpm-rfc793bis] is used by many applications in the Internet, including control and management protocols. ~~Therefore~~As such, 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 minimal YANG 1.1 [RFC7950] ~~1.1~~ model for configuring TCP on network elements that support YANG ~~data models,~~. This YANG module ~~and~~ is compliant with Network Management Datastore Architecture (NMDA) [RFC8342] ~~compliant~~. The ~~model~~ YANG module has a narrow scope and focuses on a subset of fundamental TCP functions and basic statistics. It defines a container for TCP connection that includes definitions from YANG Groupings for TCP Clients and TCP Servers [I-D.ietf-netconf-tcp-client-server]. The ~~model~~ module also ~~enables~~supports configuration of TCP-AO [RFC5925], which is a relevant TCP ~~feature~~ functionality on network elements such as routers. The ~~model~~ module can be augmented or updated to address more advanced or implementation-specific TCP features in the future.

Many protocol stacks on ~~Internet~~ IP 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 [RFC6241] or RESTCONF [RFC8040]. Moreover, many existing TCP/IP stacks do not use YANG data models. Such TCP implementations often have other means to configure the parameters listed in this document. Such other means ~~, which~~ are outside the scope of this document.

This specification is orthogonal to the Management Information Base (MIB) for the Transmission Control Protocol (TCP) [RFC4022]. The basic statistics defined in this document follow the model of the TCP MIB. An TCP Extended Statistics MIB [RFC4898] is also available, but this document does not cover such extended statistics. It is possible also to translate a MIB into a YANG ~~model~~module, for instance using Translation of Structure of Management Information Version 2 (SMIv2) MIB Modules to YANG Modules [RFC6643]. However, this approach is not used in this document~~, as such~~ because a translated model would not be up-to-date.

There are other existing TCP-related YANG models, which are orthogonal to this specification. Examples are:

- TCP header attributes are modeled in other models, such as YANG Data Model for Network Access Control Lists (ACLs) [RFC8519] and Distributed Denial-of-Service Open Thread Signaling (DOTS) Data Channel Specification [RFC8783].

Commenté [BMI2]: Start a new paragraph.

Mis en forme : Anglais (États-Unis)

Commenté [BMI3]: To avoid confusion with “features” used in YANG

- TCP-related configuration of a NAT (e.g., NAT44, NAT64, Destination NAT~~, ...~~) is defined in “A YANG Module for Network Address Translation (NAT) and Network Prefix Translation (NPT)” [RFC8512] and “A YANG Data Model for Dual-Stack Lite (DS-Lite)” [RFC8513].

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.

2.1. Note to RFC Editor

This document uses several placeholder values throughout the document. Please replace them as follows and remove this note before publication.

RFC XXXX, where XXXX is the number assigned to this document at the time of publication.

Likewise, RFC CCCC should be updated to refer the RFC number assigned to [I-D.ietf-netconf-tcp-client-server].

2021-10-25 with the actual date of the publication of this document.

3. ~~Model~~ YANG Module Overview

3.1. ~~Modeling~~ Scope

TCP is implemented on ~~many~~ different system architectures. As a result, there are ~~may~~ 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 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~~is the use of

Commenté [BMI4]: Redundant with « P » of tcP socket options in the socket API, such as disabling the Nagle algorithm by TCP_NODELAY. If 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 BGP YANG Model for Service Provider Networks [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 An Abstract Application Layer Interface to Transport Services [I-D.ietf-taps-interface].

As a result, there is no ground truth for setting certain TCP parameters, and traditionally different TCP 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, TCP can be implemented in different ways and design choices by the protocol engine often affect configuration options.

Nonetheless, a number of TCP stack parameters require configuration by YANG models. This document therefore defines a minimal YANG model with fundamental parameters directly following from TCP standards.

An important use case is the TCP configuration on network elements such as routers, which often use YANG data models. The model therefore specifies TCP parameters that are important on such TCP stacks.

This in particular applies to the support of TCP-AO [RFC5925]. TCP Authentication Option (TCP-AO) is used on routers to secure routing protocols such as BGP. In that case, a YANG model for TCP-AO configuration is required. The model defined in this document includes the required parameters for TCP-AO configuration, such as the values of SendID and RecvID. The key chain for TCP-AO can be modeled by the YANG Data Model for Key Chains [RFC8177].

Given an installed base, the model also allows enabling of the legacy TCP MD5 [RFC2385] signature option. As the TCP MD5 signature option is obsoleted by TCP-AO, it is strongly RECOMMENDED to use TCP-AO.

Similar to the TCP MIB [RFC4022], this document also specifies basic statistics and a TCP connection table.

- 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. Note, the connection table is modeled as a list that is read-writeable, even though a connection cannot be created by adding entries to the table. Similarly, deletion of connections from this list is implementation-specific.

Commenté [BMI5]: You may indicate that MPTCP-specifics are not covered in this version.

This allows implementations of TCP MIB [RFC4022] to migrate to the YANG model defined in this memo. Note that the TCP MIB does not include means to reset statistics, which are defined in this document. This is not a major addition, as a reset can simply be implemented by storing offset values for the counters.

3.2. Model Design

The YANG model defined in this document includes definitions from ~~the YANG Groupings for TCP Clients and TCP Servers~~ [I-D.ietf-netconf-tcp-client-server]. Similar to that model, this specification defines reusable 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. An example could be ~~the BGP YANG Model for Service Provider Networks~~ [I-D.ietf-idr-bgp-model].

3.3. Tree Diagram

This section provides a ~~abridged~~ tree diagram for the YANG module defined in this document. Annotations used in the diagram are defined in YANG Tree Diagrams [RFC8340].

```
module: ietf-tcp
  +--rw tcp!
     +--rw connections
     |  +--rw connection*
     |          [local-address remote-address local-port remote-port]
     |     +--rw local-address     inet:ip-address
     |     +--rw remote-address    inet:ip-address
     |     +--rw local-port        inet:port-number
     |     +--rw remote-port       inet:port-number
     |     +--rw common
     |        +--rw keepalives! {keepalives-supported}?
     |        |  +--rw idle-time         uint16
     |        |  +--rw max-probes        uint16
     |        |  +--rw probe-interval    uint16
     |        +--rw (authentication)?
     |           +--:(ao)
     |           |  +--rw enable-ao?             boolean
     |           |  +--rw send-id?               uint8
     |           |  +--rw recv-id?               uint8
     |           |  +--rw include-tcp-options?   boolean
     |           |  +--rw accept-key-mismatch?   boolean
     |           +--:(md5)
     |              +--rw enable-md5?            boolean
     +--ro statistics {statistics}?
```

Commenté [BMI6]: Isn't useful to retrieve the state of a connection?

Commenté [BMI7]: This may not be easily augmented in the future to support MPTCP.

```
        +--ro active-opens?              yang:counter32
        +--ro passive-opens?             yang:counter32
        +--ro attempt-fails?             yang:counter32
        +--ro establish-resets?          yang:counter32
        +--ro currently-established?     yang:gauge32
        +--ro in-segments?               yang:counter64
        +--ro out-segments?              yang:counter64
        +--ro retransmitted-segments?    yang:counter32
        +--ro in-errors?                 yang:counter32
        +--ro out-resets?                yang:counter32
        +---x reset
           +---w input
           |  +---w reset-at?    yang:date-and-time
           +--ro output
              +--ro reset-finished-at?    yang:date-and-time~~   module: ietf-tcp~~
~~     +--rw tcp!~~
~~        +--rw connections~~
~~        |     ...~~
~~        +--ro statistics {statistics}?~~
~~              ...~~
```

4. TCP YANG Model

The module imports types defined in [RFC6991].

```
   <CODE BEGINS> file "ietf-tcp@2021-10-25.yang"
   module ietf-tcp {
     yang-version "1.1";
     namespace "urn:ietf:params:xml:ns:yang:ietf-tcp";
     prefix "tcp";

     import ietf-yang-types {
       prefix "yang";
       reference
         "RFC 6991: Common YANG Data Types, Section 3.";
     }
     import ietf-tcp-common {
       prefix "tcpcmn";
       reference
         "RFC CCCC: YANG Groupings for TCP Clients and TCP Servers.";

     }
     import ietf-inet-types {
       prefix "inet";
       reference
         "RFC 6991: Common YANG Data Types, Section 4.";

     }

     organization
       "IETF TCPM Working Group";

     contact
       "WG Web:   <https://datatracker.ietf.org/wg/tcpm/about>
        WG List:  <tcpm@ietf.org>

        Authors: Michael Scharf (michael.scharf at hs-esslingen dot de)
                 Mahesh Jethanandani (mjethanandani at gmail dot com)
                 Vishal Murgai (vmurgai at gmail dot com)";

     description
       "This module focuses on fundamental TCP functions and basic
        statistics. The model can be augmented to address more advanced
        or implementation specific TCP features.

        Copyright (c) 2021 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject to
        the license terms contained in, the Simplified BSD License set
        forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (https://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC XXXX
        (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
        for full legal notices.
```

Commenté [BMI8]: Please run “pyang -f yang --yang-canonical”

```
        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 (RFC 2119) (RFC 8174) when, and only when,
        they appear in all capitals, as shown here.";

     revision "2021-10-25" {
       description
         "Initial Version";
       reference
         "RFC XXXX~~, ~~:  A YANG Model for Transmission Control Protocol (TCP)
                    Configuration.";
     }

     // Features
     feature statistics {
       description
         "This implementation supports statistics reporting.";
     }

     // TCP-AO Groupings

     grouping ao {
       leaf enable-ao {
         type boolean;
         default "false";
         description
           "Enable support of TCP-Authentication Option (TCP-AO).";
       }

       leaf send-id {
         type uint8 {
           range "0..255";
         }
         must "../enable-ao = 'true'";
         description
           "The SendID is inserted as the KeyID of the TCP-AO option
            of outgoing segments. The SendID must match the RecvID
            at the other endpoint.";
         reference
           "RFC 5925: The TCP Authentication Option.";
       }

       leaf recv-id {
         type uint8 {
           range "0..255";
         }
         must "../enable-ao = 'true'";
```

Commenté [BMI9]: Please add a reference to the section where this is defined

```
         description
           "The RecvID is matched against the TCP-AO KeyID of incoming
            segments. The RecvID must match the SendID at the other
            endpoint.";
         reference
           "RFC 5925: The TCP Authentication Option.";
       }

       leaf include-tcp-options {
         type boolean;
         must "../enable-ao = 'true'";
         default true;
         description
           "Include TCP options in MAC calculation.";
       }

       leaf accept-key-mismatch {
         type boolean;
         must "../enable-ao = 'true'";
         description
           "Accept TCP segments with a Master Key Tuple (MKT) that is
            not configured.";
       }
       description
         "Authentication Option (AO) for TCP.";
       reference
         "RFC 5925: The TCP Authentication Option.";
     }

     // MD5 grouping

     grouping md5 {
       description
         "Grouping for use in authenticating TCP sessions using MD5.";
       reference
         "RFC 2385: Protection of BGP Sessions via the TCP MD5
                    Signature.";

       leaf enable-md5 {
         type boolean;
         default "false";
         description
           "Enables, when set to true, support of MD5 to authenticate a TCP
session.";
       }
     }

     // TCP configuration
```

Commenté [BMI10]: Idem as above

Commenté [BMI11]: Please consider adding a reference statement

Commenté [BMI12]: Please consider adding a reference statement

```
     container tcp {
       presence "The container for TCP configuration.";

       description
         "TCP container.";

       container connections {
         list connection {
           key "local-address remote-address local-port remote-port";

           leaf local-address {
             type inet:ip-address;
             description
               "Local IP address that forms the connection identifier.";
           }

           leaf remote-address {
             type inet:ip-address;
             description
               "Remote address that forms the connection identifier.";
           }

           leaf local-port {
             type inet:port-number;
             description
               "Local TCP port number that forms the connection identifier.";
           }

           leaf remote-port {
             type inet:port-number;
             description
               "Remote TCP port number that forms the connection identifier.";
           }

           container common {
             uses tcpcmn:tcp-common-grouping;

             choice authentication {
               case ao {
                 uses ao;
                 description
                   "Use TCP-AO to secure the connection.";
               }

               case md5 {
                 uses md5;
                 description
                   "Use TCP-MD5 to secure the connection.";
```

Commenté [BMI13]: See the comment about future augmentations to support MPTCP connections

Commenté [BMI14]: This isn’t sufficient in its own to identify a connection.

Mis en forme : Surlignage

Mis en forme : Surlignage

Mis en forme : Surlignage

```
               }
               description
                 "Choice of ~~how to secure the ~~TCP connection authentication.";
             }
             description
               "Common definitions of TCP configuration. This includes
                parameters such as how to secure the connection,
                that can be part of either the client or server.";
           }
           description
             "List of TCP connections with their parameters. The list
              is modeled as writeable, but implementations may not
              allow creation of new TCP connections by adding entries to
              the list. Furthermore, the behavior upon removal is
              implementation-specific. Implementations may support
              closing or resetting a TCP connection upon an operation
              that removes the entry from the list.";
         }
         description
           "A container of all TCP connections.";
       }

       container statistics {
         if-feature statistics;
         config false;

         leaf active-opens {
           type yang:counter32;
           description
             "The number of times that TCP connections have made a
              direct transition to the SYN-SENT state from the CLOSED
              state.";
         }

         leaf passive-opens {
           type yang:counter32;
           description
             "The number of times TCP connections have made a direct
              transition to the SYN-RCVD state from the LISTEN state.";
         }

         leaf attempt-fails {
           type yang:counter32;
           description
             "The number of times that TCP connections have made a
              direct transition to the CLOSED state from either the
              SYN-SENT state or the SYN-RCVD state, plus the number of
              times that TCP connections have made a direct transition
              to the LISTEN state from the SYN-RCVD state.";
         }

         leaf establish-resets {
           type yang:counter32;
           description
             "The number of times that TCP connections have made a
              direct transition to the CLOSED state from either the
              ESTABLISHED state or the CLOSE-WAIT state.";
         }

         leaf currently-established {
           type yang:gauge32;
           description
             "The number of TCP connections for which the current state
              is either ESTABLISHED or CLOSE-WAIT.";
         }

         leaf in-segments {
           type yang:counter64;
           description
             "The total number of segments received, including those
              received in error.  This count includes segments received
              on currently established connections.";
         }

         leaf out-segments {
           type yang:counter64;
           description
             "The total number of segments sent, including those on
              current connections but excluding those containing only
              retransmitted octets.";
         }

         leaf retransmitted-segments {
           type yang:counter32;
           description
             "The total number of segments retransmitted; that is, the
              number of TCP segments transmitted containing one or more
              previously transmitted octets.";
         }

         leaf in-errors {
           type yang:counter32;
           description
             "The total number of segments received in error (e.g., bad
              TCP checksums).";
         }
         leaf out-resets {
           type yang:counter32;
           description
             "The number of TCP segments sent containing the RST flag.";
         }

         action reset {
           description
             "Reset statistics action command.";
           input {
             leaf reset-at {
               type yang:date-and-time;
               description
                 "Time when the reset action needs to be
                  executed.";
             }
           }
           output {
             leaf reset-finished-at {
               type yang:date-and-time;
               description
                 "Time when the reset action command completed.";
             }
           }
         }
         description
           "Statistics across all connections.";
       }
     }
   }
   <CODE ENDS>
```

5. IANA Considerations

5.1. The IETF XML Registry

This document registers an URI in the "ns" subregistry of the IETF XML Registry [RFC3688]. Following the format in IETF XML Registry [RFC3688], the following registration is requested:

URI: urn:ietf:params:xml:ns:yang:ietf-tcp
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.

5.2. The YANG Module Names Registry

This document registers a YANG module in the "YANG Module Names" registry YANG - A Data Modeling Language [RFC6020]. Following the format in YANG - A Data Modeling Language [RFC6020], the following registrations are requested:

```
   name:         ietf-tcp
   namespace:    urn:ietf:params:xml:ns:yang:ietf-tcp
   prefix:       tcp
   reference:    RFC XXXX
```

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) described in Using the NETCONF protocol over 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.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., "config true", which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

* Common configuration included from NETCONF Client and Server Models [I-D.ietf-netconf-tcp-client-server]. Unrestricted access to all the nodes, e.g., keepalive idle-timer, can cause connections to fail or to timeout prematurely.

* Authentication configuration. Unrestricted access to the nodes under authentication configuration can prevent the use of authenticated communication and cause connection setups to fail. This can result in massive security vulnerabilities and service disruption for the traffic requiring authentication.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

* Unrestricted access to connection information of the client or server can be used by a malicious user to launch an attack, e.g. MITM.

* Similarly, unrestricted access to statistics of the client or server can be used by a malicious user to exploit any vulnerabilities of the system.

Some of the RPC operations in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to these operations. These are the operations and their sensitivity/vulnerability:

Commenté [BMI15]: No rpc is defined in the module.

* The YANG module allows for the statistics to be cleared by executing the reset action. This action should be restricted to users with the right permission.

7. References

7.1. Normative References

[I-D.ietf-netconf-tcp-client-server]
Watsen, K. and M. Scharf, "YANG Groupings for TCP Clients and TCP Servers", Work in Progress, Internet-Draft, draft-ietf-netconf-tcp-client-server-10, 18 May 2021, <https://www.ietf.org/archive/id/draft-ietf-netconf-tcp-client-server-10.txt>.

[I-D.ietf-tcpm-rfc793bis]
Eddy, W. M., "Transmission Control Protocol (TCP) Specification", Work in Progress, Internet-Draft, draft-ietf-tcpm-rfc793bis-25, 7 September 2021, <https://www.ietf.org/archive/id/draft-ietf-tcpm-rfc793bis-25.txt>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2385] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, DOI 10.17487/RFC2385, August 1998, <https://www.rfc-editor.org/info/rfc2385>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <https://www.rfc-editor.org/info/rfc5925>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8177] Lindem, A., Ed., Qu, Y., Yeung, D., Chen, I., and J. Zhang, "YANG Data Model for Key Chains", RFC 8177, DOI 10.17487/RFC8177, June 2017, <https://www.rfc-editor.org/info/rfc8177>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

7.2. Informative References

[I-D.ietf-idr-bgp-model]
Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP YANG Model for Service Provider Networks", Work in Progress, Internet-Draft, draft-ietf-idr-bgp-model-11, 11 July 2021, <https://www.ietf.org/archive/id/draft-ietf-idr-bgp-model-11.txt>.

[I-D.ietf-taps-interface]
Trammell, B., Welzl, M., Enghardt, T., Fairhurst, G., Kuehlewind, M., Perkins, C., Tiesel, P. S., Wood, C. A., Pauly, T., and K. Rose, "An Abstract Application Layer Interface to Transport Services", Work in Progress, Internet-Draft, draft-ietf-taps-interface-13, 12 July 2021, <https://www.ietf.org/archive/id/draft-ietf-taps-interface-13.txt>.

[I-D.touch-tcpm-ao-test-vectors]
Touch, J. and J. Kuusisaari, "TCP-AO Test Vectors", Work in Progress, Internet-Draft, draft-touch-tcpm-ao-test-vectors-02, 23 December 2020, <https://www.ietf.org/archive/id/draft-touch-tcpm-ao-test-vectors-02.txt>.

[RFC4022] Raghunarayan, R., Ed., "Management Information Base for the Transmission Control Protocol (TCP)", RFC 4022, DOI 10.17487/RFC4022, March 2005, <https://www.rfc-editor.org/info/rfc4022>.

[RFC4898] Mathis, M., Heffner, J., and R. Raghunarayan, "TCP Extended Statistics MIB", RFC 4898, DOI 10.17487/RFC4898, May 2007, <https://www.rfc-editor.org/info/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, <https://www.rfc-editor.org/info/rfc6643>.

[RFC8512] Boucadair, M., Ed., Sivakumar, S., Jacquenet, C., Vinapamula, S., and Q. Wu, "A YANG Module for Network Address Translation (NAT) and Network Prefix Translation (NPT)", RFC 8512, DOI 10.17487/RFC8512, January 2019, <https://www.rfc-editor.org/info/rfc8512>.

[RFC8513] Boucadair, M., Jacquenet, C., and S. Sivakumar, "A YANG Data Model for Dual-Stack Lite (DS-Lite)", RFC 8513, DOI 10.17487/RFC8513, January 2019, <https://www.rfc-editor.org/info/rfc8513>.

[RFC8519] Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <https://www.rfc-editor.org/info/rfc8519>.

[RFC8783] Boucadair, M., Ed. and T. Reddy.K, Ed., "Distributed Denial-of-Service Open Threat Signaling (DOTS) Data Channel Specification", RFC 8783, DOI 10.17487/RFC8783, May 2020, <https://www.rfc-editor.org/info/rfc8783>.

Appendix A. Acknowledgements

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

The following persons have contributed to this document by reviews: Mohamed Boucadair

Appendix B. Changes compared to previous versions

Changes compared to draft-scharf-tcpm-yang-tcp-04

- Removed congestion control

- Removed global stack parameters

Changes compared to draft-scharf-tcpm-yang-tcp-03

- Updated TCP-AO grouping

- Added congestion control

Changes compared to draft-scharf-tcpm-yang-tcp-02

- Initial proposal of a YANG model including base configuration parameters, TCP-AO configuration, and a connection list

- Editorial bugfixes and outdated references reported by Mohamed Boucadair

- Additional co-author Mahesh Jethanandani

Changes compared to draft-scharf-tcpm-yang-tcp-01

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

- Removing backward-compatibility to the TCP MIB

- Additional co-author Vishal Murgai

Changes compared to draft-scharf-tcpm-yang-tcp-00

- Editorial improvements

Appendix C. Examples

C.1. Keepalive Configuration

This particular example demonstrates how both a particular connection can be configured for keepalives.

[note: '\' line wrapping for formatting only]

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
This example shows how TCP keepalive can be configured for
a given connection. An idle connection is dropped after
idle-time + (max-probes * probe-interval).
-->
<tcp
    xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp">
  <connections>
    <connection>
      <local-address>192.168.1.1</local-address>
      <remote-address>192.168.1.2</remote-address>
      <local-port>1025</local-port>
      <remote-port>80</remote-port>
      <common>
        <keepalives>
          <idle-time>5</idle-time>
          <max-probes>5</max-probes>
          <probe-interval>10</probe-interval>
        </keepalives>
      </common>
    </connection>
  </connections>
</tcp>
```

C.2. TCP-AO Configuration

The following example demonstrates how to model a TCP-AO [RFC5925] configuration for the example in TCP-AO Test Vectors [I-D.touch-tcpm-ao-test-vectors], Section 5.1.1.

```
[note: '\' line wrapping for formatting only]

<?xml version="1.0" encoding="UTF-8"?>
<!--
This example sets TCP-AO configuration parameters as
demonstrated by examples in draft-touch-tcpm-ao-test-vectors.
-->

<tcp
    xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp">
  <connections>
    <connection>
      <local-address>192.168.1.1</local-address>
      <remote-address>192.168.1.2</remote-address>
      <local-port>1025</local-port>
      <remote-port>80</remote-port>
      <common>
        <enable-ao>true</enable-ao>
      </common>
    </connection>
  </connections>
</tcp>

<key-chains
    xmlns="urn:ietf:params:xml:ns:yang:ietf-key-chain">
  <key-chain>
    <name>ao-config</name>
    <description>"An example for TCP-AO configuration."</description>\

    <key>
      <key-id>61</key-id>
      <crypto-algorithm>hmac-sha-1</crypto-algorithm>
      <key-string>
        <hexadecimal-string>01:23:a5:93:b9:db:70:62:9b:be:2c:a6:77:cd:fd:ea:\
6f:e0:ac:ad</hexadecimal-string>
      </key-string>
    </key>
  </key-chain>
</key-chains>
```

Appendix D. Complete Tree Diagram

Here is the complete tree diagram for the TCP YANG model.

```
   module: ietf-tcp
     +--rw tcp!
        +--rw connections
        |  +--rw connection*
        |          [local-address remote-address local-port remote-port]
        |     +--rw local-address     inet:ip-address
        |     +--rw remote-address    inet:ip-address
        |     +--rw local-port        inet:port-number
        |     +--rw remote-port       inet:port-number
        |     +--rw common
        |        +--rw keepalives!
        |        |  +--rw idle-time         uint16
        |        |  +--rw max-probes        uint16
        |        |  +--rw probe-interval    uint16
        |        +--rw (authentication)?
        |           +--:(ao)
        |           |  +--rw enable-ao?             boolean
        |           |  +--rw send-id?               uint8
        |           |  +--rw recv-id?               uint8
        |           |  +--rw include-tcp-options?   boolean
        |           |  +--rw accept-key-mismatch?   boolean
        |           +--:(md5)
        |              +--rw enable-md5?            boolean
        +--ro statistics {statistics}?
           +--ro active-opens?            yang:counter32
           +--ro passive-opens?           yang:counter32
           +--ro attempt-fails?           yang:counter32
           +--ro establish-resets?        yang:counter32
           +--ro currently-established?   yang:gauge32
           +--ro in-segments?             yang:counter64
           +--ro out-segments?            yang:counter64
           +--ro retransmitted-segments?  yang:counter32
           +--ro in-errors?               yang:counter32
           +--ro out-resets?              yang:counter32
           +---x reset
              +---w input
              |  +---w reset-at?   yang:date-and-time
              +--ro output
                 +--ro reset-finished-at?   yang:date-and-time
```

Authors' Addresses

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

Email: michael.scharf@hs-esslingen.de

Mahesh Jethanandani
Kloud Services

Email: mjethanandani@gmail.com

Vishal Murgai
Samsung

Email: vmurgai@gmail.com