NETCONF Working Group K. Watsen
Internet-Draft Watsen Networks
Intended status: Standards Track M. Scharf
Expires: April 20, 2020 Hochschule Esslingen
October 18, 2019

YANG Groupings for TCP Clients and TCP Servers
draft-ietf-netconf-tcp-client-server-03

Abstract

This document defines three YANG modules: the first defines a grouping for configuring a generic TCP client, the second defines a grouping for configuring a generic TCP server, and the third defines a grouping common to the TCP clients and TCP servers.

Editorial Note (To be removed by RFC Editor)

This draft contains many placeholder values that need to be replaced with finalized values at the time of publication. This note summarizes all of the substitutions that are needed. No other RFC Editor instructions are specified elsewhere in this document.

Artwork in this document contains placeholder values for the date of publication of this draft. Please apply the following replacement:

o "2019-10-18" --> the publication date of this draft

The following Appendix section is to be removed prior to publication:

o Appendix A. Change Log

Status of This Memo

This Internet-Draft will expire on April 20, 2020.

Copyright Notice

Table of Contents

1. Introduction   3
2. Terminology   3
3. The TCP Common Model   3
  3.1. Model Scope   3
  3.2. Usage Guidelines for Configuring TCP Keep-Alives   3
  3.3. Tree Diagram   4
  3.4. Example Usage   5
  3.5. YANG Module   5
4. The TCP Client Model   8
  4.1. Tree Diagram   8
  4.2. Example Usage   9
  4.3. YANG Module   9
5. The TCP Server Model 12
  5.1. Tree Diagram 12
  5.2. Example Usage 13
  5.3. YANG Module 13
6. Security Considerations 15
7. IANA Considerations 16
  7.1. The IETF XML Registry 16
  7.2. The YANG Module Names Registry 17
8. References 17
  8.1. Normative References 17
  8.2. Informative References 18
Appendix A. Change Log 19
  A.1. 00 to 01 19
  A.2. 01 to 02 19
  A.3. 02 to 03 19
Authors' Addresses 19

1. Introduction

This document defines three YANG 1.1 [RFC7950] modules: the first defines a grouping for configuring a generic TCP client, the second defines a grouping for configuring a generic TCP server, and the third defines a grouping common to the TCP clients and TCP servers.

It is intended that these groupings will be used either standalone, for TCP-based protocols, as part of a stack of protocol-specific configuration models. For instance, these groupings could help define the configuration module for SSH, TLS, or HTTP based applications.

2. Terminology

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. The TCP Common Model

3.1. Model Scope

This document defines a common "grouping" statement for basic TCP connection parameters that matter to applications. In some TCP stacks, such parameters can also directly be set by an application using system calls, such as the socket API. The base YANG model in this document focuses on modeling TCP keep-alives. This base model can be extended as needed.

3.2. Usage Guidelines for Configuring TCP Keep-Alives

Network stacks may include "keep-alives" in their TCP implementations, although this practice is not universally accepted. If keep-alives are included, [RFC1122] [RFC793bis] mandates that the application MUST be able to turn them on or off for each TCP connection, and that they MUST default to off.

Keep-alive mechanisms exist in many protocols. Depending on the protocol stack, TCP keep-alives may only be one out of several alternatives. Which mechanism to use depends on the use case and application requirements. If keep-alives are needed by an application, it is RECOMMENDED that the aliveness check happens at the highest protocol layer possible that is meaningful to the application, in order to maximize the depth of the aliveness check.

[[TODO: Further guidance on keep-alives is provided in draft-xyz-tsvwg-...]]

A TCP keep-alive mechanism should only be invoked in server applications that might otherwise hang indefinitely and consume resources unnecessarily if a client crashes or aborts a connection during a network failure [RFC1122]. TCP keep-alives may consume significant resources both in the network and in endpoints (e.g., battery power). In addition, frequent keep-alives risk network congestion. The higher the frequency of keep-alives, the higher the overhead.

Given the cost of keep-alives, parameters have to be configured carefully:

- The default idle interval (leaf "idle-time") MUST default to no less than two hours, i.e., 7200 seconds [RFC1122]. A lower value MAY be configured, but keep-alive messages SHOULD NOT be transmitted more frequently than once every 15 seconds. Longer intervals SHOULD be used when possible.

- The maximum number of sequential keep-alive probes that can fail (leaf "max-probes") trades off responsiveness and robustness against packet loss. ACK segments that contain no data are not reliably transmitted by TCP. Consequently, if a keep-alive mechanism is implemented it MUST NOT interpret failure to respond to any specific probe as a dead connection [RFC1122]. Typically a single-digit number should suffice.

- TCP implementations may include a parameter for the number of seconds between TCP keep-alive probes (leaf "probe-interval"). In order to avoid congestion, the time interval between probes MUST NOT be smaller than one second. Significantly longer intervals SHOULD be used. It is important to note that keep-alive probes (or replies) can get dropped due to network congestion. Sending further probe messages into a congested path after a short interval, without backing off timers, could cause harm and result in a congestion collapse. Therefore it is essential to pick a large, conservative value for this interval.

3.3. Tree Diagram

This section provides a tree diagram [RFC8340] for the "ietf-tcp-common" module.

```
module: ietf-tcp-common

  grouping tcp-common-grouping
    +-- keepalives! {keepalives-supported}?
       +-- idle-time         uint16
       +-- max-probes        uint16
       +-- probe-interval    uint16
  grouping tcp-connection-grouping
    +-- keepalives! {keepalives-supported}?
       +-- idle-time         uint16
       +-- max-probes        uint16
       +-- probe-interval    uint16
```

3.4. Example Usage

This section presents an example showing the tcp-common-grouping populated with some data.

```
<tcp-common xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp-common">
  <keepalives>
    <idle-time>15</idle-time>
    <max-probes>3</max-probes>
    <probe-interval>30</probe-interval>
  </keepalives>
</tcp-common>
```

3.5. YANG Module

The ietf-tcp-common YANG module references [RFC6991].

<CODE BEGINS> file "ietf-tcp-common@2019-10-18.yang"

```
module ietf-tcp-common {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tcp-common";
  prefix tcpcmn;

  organization
    "IETF NETCONF (Network Configuration) Working Group and the
     IETF TCP Maintenance and Minor Extensions (TCPM) Working Group";

  contact
    "WG Web:   <http://datatracker.ietf.org/wg/netconf/>
               <http://datatracker.ietf.org/wg/tcpm/>
     WG List:  <mailto:netconf@ietf.org>
               <mailto:tcpm@ietf.org>
     Authors:  Kent Watsen <mailto:kent+ietf@watsen.net>
               Michael Scharf
```

```
                  <mailto:michael.scharf@hs-esslingen.de>";

     description
       "This module defines reusable groupings for TCP commons that
        can be used as a basis for specific TCP common instances.

        Copyright (c) 2019 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.;

        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 2019-10-18 {
       description
         "Initial version";
       reference
         "RFC XXXX: YANG Groupings for TCP Clients and TCP Servers";
     }

     // Features
     feature keepalives-supported {
       description
         "Indicates that keepalives are supported.";
     }

     // Groupings

     grouping tcp-common-grouping {
       description
         "A reusable grouping for configuring TCP parameters common
          to TCP connections as well as the operating system as a
          whole.";
       container keepalives {
```

```
         if-feature "keepalives-supported";
         presence "Indicates that keepalives are enabled.";
         description
           "Configures the keep-alive policy, to proactively test the
            aliveness of the TCP peer.  An unresponsive TCP peer is
            dropped after approximately (idle-time + max-probes
            * probe-interval) seconds.";
         leaf idle-time {
           type uint16 {
             range "1..max";
           }
           units "seconds";
           mandatory true;
           description
             "Sets the amount of time after which if no data has been
              received from the TCP peer, a TCP-level probe message
              will be sent to test the aliveness of the TCP peer.
              Two hours (7200 seconds) is safe value, per RFC 1122.";
           reference
             "RFC 1122:
               Requirements for Internet Hosts -- Communication Layers";
         }
         leaf max-probes {
           type uint16 {
             range "1..max";
           }
           mandatory true;
           description
             "Sets the maximum number of sequential keep-alive probes
              that can fail to obtain a response from the TCP peer
              before assuming the TCP peer is no longer alive.";
         }
         leaf probe-interval {
           type uint16 {
             range "1..max";
           }
           units "seconds";
           mandatory true;
           description
             "Sets the time interval between failed probes. The interval
              SHOULD be significantly longer than one second in order to
              avoid harm on a congested link.";
         }
       } // container keepalives
     } // grouping tcp-common-grouping


     grouping tcp-connection-grouping {
```

```
       description
         "A reusable grouping for configuring TCP parameters common
          to TCP connections.";
        uses tcp-common-grouping;
     }

   /*
     The following is for a future bis...
     This comment is here now so as support discussion with TCPM.
     This comment will be removed before publication.

     Should future system-level parameters be defined as a
     grouping or a container?

     grouping tcp-system-grouping {
       description
         "A reusable grouping for configuring TCP parameters common
          to the operating system as a whole.";

       // currently just a placeholder
     }
   */

   }

   <CODE ENDS>
```

4. The TCP Client Model

4.1. Tree Diagram

This section provides a tree diagram [RFC8340] for the "ietf-tcp-client" module.

```
   module: ietf-tcp-client

     grouping tcp-client-grouping
       +-- remote-address    inet:host
       +-- remote-port?      inet:port-number
       +-- local-address?    inet:ip-address {local-binding-supported}?
       +-- local-port?       inet:port-number {local-binding-supported}?
       +-- keepalives! {keepalives-supported}?
          +-- idle-time         uint16
          +-- max-probes        uint16
          +-- probe-interval    uint16
```

4.2. Example Usage

This section presents an example showing the tcp-client-grouping populated with some data.

```
<tcp-client xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp-client">
  <remote-address>www.example.com</remote-address>
  <remote-port>443</remote-port>
  <local-address>0.0.0.0</local-address>
  <local-port>0</local-port>
  <keepalives>
    <idle-time>15</idle-time>
    <max-probes>3</max-probes>
    <probe-interval>30</probe-interval>
  </keepalives>
</tcp-client>
```

4.3. YANG Module

The ietf-tcp-client YANG module references [RFC6991].

```
<CODE BEGINS> file "ietf-tcp-client@2019-10-18.yang"

module ietf-tcp-client {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tcp-client";
  prefix tcpc;

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

  import ietf-tcp-common {
    prefix tcpcmn;
    reference
      "RFC XXXX: YANG Groupings for TCP Clients and TCP Servers";
  }

  organization
    "IETF NETCONF (Network Configuration) Working Group and the
     IETF TCP Maintenance and Minor Extensions (TCPM) Working Group";

  contact
    "WG Web:   <http://datatracker.ietf.org/wg/netconf/>
               <http://datatracker.ietf.org/wg/tcpm/>
     WG List:  <mailto:netconf@ietf.org>
```

```
                  <mailto:tcpm@ietf.org>
        Authors:  Kent Watsen <mailto:kent+ietf@watsen.net>
                  Michael Scharf
                  <mailto:michael.scharf@hs-esslingen.de>";

     description
       "This module defines reusable groupings for TCP clients that
        can be used as a basis for specific TCP client instances.

        Copyright (c) 2019 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.;

        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 2019-10-18 {
       description
         "Initial version";
       reference
         "RFC XXXX: YANG Groupings for TCP Clients and TCP Servers";
     }

     // Features

     feature local-binding-supported {
       description
         "Indicates that the server supports configuring local
          bindings (i.e., the local address and local port) for
          TCP clients.";
     }

     feature tcp-client-keepalives {
       description
```

```
         "Per socket TCP keepalive parameters are configurable for
          TCP clients on the server implementing this feature.";
     }

     // Groupings

     grouping tcp-client-grouping {
       description
         "A reusable grouping for configuring a TCP client.

         Note that this grouping uses fairly typical descendent
          node names such that a stack of 'uses' statements will
          have name conflicts.  It is intended that the consuming
          data model will resolve the issue (e.g., by wrapping
          the 'uses' statement in a container called
          'tcp-client-parameters').  This model purposely does
          not do this itself so as to provide maximum flexibility
          to consuming models.";

       leaf remote-address {
         type inet:host;
         mandatory true;
         description
           "The IP address or hostname of the remote peer to
            establish a connection with.  If a domain name is
            configured, then the DNS resolution should happen on
            each connection attempt.  If the DNS resolution
            results in multiple IP addresses, the IP addresses
            are tried according to local preference order until
            a connection has been established or until all IP
            addresses have failed.";
       }
       leaf remote-port {
         type inet:port-number;
         default "0";
         description
           "The IP port number for the remote peer to establish a
            connection with.  An invalid default value (0) is used
            (instead of 'mandatory true') so that as application
            level data model may 'refine' it with an application
            specific default port number value.";
       }
       leaf local-address {
         if-feature "local-binding-supported";
         type inet:ip-address;
         description
           "The local IP address/interface (VRF?) to bind to for when
            connecting to the remote peer.  INADDR_ANY ('0.0.0.0') or
```

```
            INADDR6_ANY ('0:0:0:0:0:0:0:0' a.k.a. '::') MAY be used to
            explicitly indicate the implicit default, that the server
            can bind to any IPv4 or IPv6 addresses, respectively.";
       }
       leaf local-port {
         if-feature "local-binding-supported";
         type inet:port-number;
         default "0";
         description
           "The local IP port number to bind to for when connecting
            to the remote peer.  The port number '0', which is the
            default value, indicates that any available local port
            number may be used.";
       }
       uses tcpcmn:tcp-connection-grouping {
         augment "keepalives" {
           if-feature "tcp-client-keepalives";
           description
             "Add an if-feature statement so that implementations
              can choose to support TCP client keepalives.";
         }
       }
     }
   }

   <CODE ENDS>
```

5. The TCP Server Model

5.1. Tree Diagram

This section provides a tree diagram [RFC8340] for the "ietf-tcp-server" module.

```
   module: ietf-tcp-server

     grouping tcp-server-grouping
       +-- local-address    inet:ip-address
       +-- local-port?      inet:port-number
       +-- keepalives! {keepalives-supported}?
          +-- idle-time         uint16
          +-- max-probes        uint16
          +-- probe-interval    uint16
```

5.2. Example Usage

This section presents an example showing the tcp-server-grouping populated with some data.

```
<tcp-server xmlns="urn:ietf:params:xml:ns:yang:ietf-tcp-server">
  <local-address>10.20.30.40</local-address>
  <local-port>7777</local-port>
  <keepalives>
    <idle-time>15</idle-time>
    <max-probes>3</max-probes>
    <probe-interval>30</probe-interval>
  </keepalives>
</tcp-server>
```

5.3. YANG Module

The ietf-tcp-server YANG module references [RFC6991].

```
<CODE BEGINS> file "ietf-tcp-server@2019-10-18.yang"

module ietf-tcp-server {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-tcp-server";
  prefix tcps;

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

  import ietf-tcp-common {
    prefix tcpcmn;
    reference
      "RFC XXXX: YANG Groupings for TCP Clients and TCP Servers";
  }

  organization
    "IETF NETCONF (Network Configuration) Working Group and the
     IETF TCP Maintenance and Minor Extensions (TCPM) Working Group";

  contact
    "WG Web:   <http://datatracker.ietf.org/wg/netconf/>
               <http://datatracker.ietf.org/wg/tcpm/>
     WG List:  <mailto:netconf@ietf.org>
               <mailto:tcpm@ietf.org>
     Authors:  Kent Watsen <mailto:kent+ietf@watsen.net>
```

```
                  Michael Scharf
                  <mailto:michael.scharf@hs-esslingen.de>";

     description
       "This module defines reusable groupings for TCP servers that
        can be used as a basis for specific TCP server instances.

        Copyright (c) 2019 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.;

        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 2019-10-18 {
       description
         "Initial version";
       reference
         "RFC XXXX: YANG Groupings for TCP Clients and TCP Servers";
     }

     // Features

     feature tcp-server-keepalives {
       description
         "Per socket TCP keepalive parameters are configurable for
          TCP servers on the server implementing this feature.";
     }


     // Groupings

     grouping tcp-server-grouping {
       description
```

```
         "A reusable grouping for configuring a TCP server.

          Note that this grouping uses fairly typical descendent
          node names such that a stack of 'uses' statements will
          have name conflicts.  It is intended that the consuming
          data model will resolve the issue (e.g., by wrapping
          the 'uses' statement in a container called
          'tcp-server-parameters').  This model purposely does
          not do this itself so as to provide maximum flexibility
          to consuming models.";
       leaf local-address {
         type inet:ip-address;
         mandatory true;
         description
           "The local IP address to listen on for incoming
            TCP client connections.  INADDR_ANY (0.0.0.0) or
            INADDR6_ANY (0:0:0:0:0:0:0:0 a.k.a. ::) MUST be
            used when the server is to listen on all IPv4 or
            IPv6 addresses, respectively.";
       }
       leaf local-port {
         type inet:port-number;
         default "0";
         description
           "The local port number to listen on for incoming TCP
            client connections.  An invalid default value (0)
            is used (instead of 'mandatory true') so that an
            application level data model may 'refine' it with
            an application specific default port number value.";
       }
       uses tcpcmn:tcp-connection-grouping {
         augment "keepalives" {
           if-feature "tcp-server-keepalives";
           description
             "Add an if-feature statement so that implementations
              can choose to support TCP server keepalives.";
         }
       }
     }
   }

   <CODE ENDS>
```

6. Security Considerations

The YANG modules defined in this document are designed to be accessed via YANG based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. Both of these protocols have mandatory-to-

implement secure transport layers (e.g., SSH, TCP) with mutual authentication.

The NETCONF access control model (NACM) [RFC8341] provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

Since the modules defined in this document only define groupings, these considerations are primarily for the designers of other modules that use these groupings.

There are a number of data nodes defined in the YANG modules 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:

None of the writable/creatable/deletable data nodes   in the YANG modules defined in this document are considered more sensitive or vulnerable then standard configuration.

Some of the readable data nodes in the YANG modules 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:

None of the readable data nodes in the YANG modules defined in this document are considered more sensitive or vulnerable then standard configuration.

This document does not define any RPC actions and hence this section does not consider the security of RPCs.

7. IANA Considerations

7.1. The IETF XML Registry

This document registers two URIs in the "ns" subregistry of the IETF XML Registry [RFC3688]. Following the format in [RFC3688], the following registrations are requested:

```
      URI: urn:ietf:params:xml:ns:yang:ietf-tcp-client
      Registrant Contact: The NETCONF WG of the IETF.
      XML: N/A, the requested URI is an XML namespace.

      URI: urn:ietf:params:xml:ns:yang:ietf-tcp-server
      Registrant Contact: The NETCONF WG of the IETF.
      XML: N/A, the requested URI is an XML namespace.
```

7.2. The YANG Module Names Registry

This document registers two YANG modules in the YANG Module Names registry [RFC6020]. Following the format in [RFC6020], the following registrations are requested:

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

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

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

8. References

8.1. Normative References

[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>.

[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>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[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>.

[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>.

[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>.

8.2. Informative References

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

[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>.

[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>.

[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>.

Appendix A. Change Log

A.1. 00 to 01

- Added 'local-binding-supported' feature to TCP-client model.

- Added 'keepalives-supported' feature to TCP-common model.

- Added 'external-endpoint-values' container and 'external-endpoints' feature to TCP-server model.

A.2. 01 to 02

- Removed the 'external-endpoint-values' container and 'external-endpoints' feature from the TCP-server model.

A.3. 02 to 03

- Moved the common model section to be before the client and server specific sections.

- Added sections "Model Scope" and "Usage Guidelines for Configuring TCP Keep-Alives" to the common model section.

Authors' Addresses

Kent Watsen
Watsen Networks

EMail: kent+ietf@watsen.net

Michael Scharf
Hochschule Esslingen - University of Applied Sciences

EMail: michael.scharf@hs-esslingen.de