

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

Michael Scharf <michael.scharf@hs-esslingen.de>

Vishal Murgai <vmurgai@cisco.com>

July 25, 2019

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

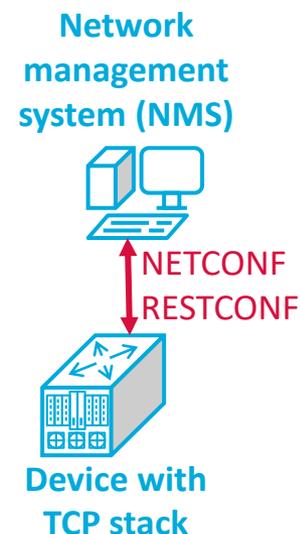
Motivation for a TCP YANG model

- Emerging related YANG models
 - TCP functions such as keepalives, e.g. draft-ietf-netconf-tcp-client-server-02
 - TCP parameters such as MSS, e.g. draft-ietf-idr-bgp-model-06
- Mostly parameters exposed by socket options
- How to deal with other TCP configuration?
 - Global configuration for all TCP connections
 - Interface-specific configuration (possibly MSS/MTU)
 - Connection-specific parameters (e.g., TCP_NODELAY)
 - Policies / profiles / templates

```
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

draft-ietf-netconf-tcp-client-server-02
```



Configuration similarities among TCP Stacks?

High-level informal survey on 4 different (anonymous) TCP stacks

Configuration example	Good match	Similar parameter but deviations	Sum	Comments
Keepalives				
• Idle-time (in seconds): integer	1	1 in msec	2	Also supported as socket option
• Probe-interval (in seconds): integer	1	1 in msec	2	Also supported as socket option
• Max-probes: integer	2		2	Also supported as socket option
Maximum MSS (in byte): integer	1	1 as input to PMTUD	2	
FIN timeout (in seconds): integer	3		3	
SACK (disable/enable): boolean	3		3	Modeled as type empty in 1 stack
Timestamps (disable/enable): boolean	3	1 as enumeration	4	Modeled as type empty in 1 stack
Path MTU Discovery (disable/enable): boolean	3	1 as enumeration	4	Modeled as type empty in 1 stack
ECN				
• Enabling (disable/passive/active): enumeration	2	1 as boolean	3	

Not included: Connection-specific parameters (i.e., socket options such as TCP_NODELAY)

→ Configuration similarities for fundamental TCP functionality

Example for a potential model

Potential YANG grouping for SACK:

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

  leaf sack {
    type boolean;
    default "true";
    description
      "Enable support of Selective Acknowledgements (SACK)";
  }
}
```

Potential use of such a YANG grouping for a TCP stack configuration:

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

Other TCP parameters

■ Modeling differences between stacks

- Delayed ACK timeout (in ms)
- Initial RTO value (in ms)
- Maximum number of retransmissions
- Window scaling
- Maximum number of connections
- ... (others following from the spec)

Could any of these ones be included (optionally)?

■ Significant dependency on stack internals

- Window size (segment vs. bytes)
- Buffer sizes and flow control
- Timers
- Congestion control algorithms
- ...

Outside the scope of the document?

Summary and next steps

- **Major changes in draft-scharf-tcpm-yang-tcp-02**
 - Alignment with draft-ietf-netconf-tcp-client-server
 - New list of (somewhat) common TCP configuration parameters
 - Vishal as new co-author
- **Potential next step: Actual YANG model**
 - Based on draft-ietf-netconf-tcp-client-server
 - Possibly only YANG groupings (like draft-ietf-netconf-tcp-client-server)
 - Definitions could be used in different context (system, interface, protocol , ...)
 - TCP configuration not always clearly separated from rest of TCP/IP stack
 - All parameters optional
- **Questions to TCPM**
 1. Is the new scope of the document realistic?
 2. If so, what TCP parameters should be included in a model?