

**A YANG Data Model for basic IP and ICMP Statistics
draft-baill-netmod-yang-ip-stats-01**

Abstract

This document defines a YANG data model for basic IP and ICMP statistics for monitoring IPv4 and IPv6 implementations.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/1id-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

Copyright and License 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

1	Introduction	3
1.1	Terminology	3
1.2	Tree Diagrams	3
2	IP Stats Model	4
3	Relationship to IP-MIB	8
4	IP Statistics YANG Module	11
5	Security Considerations	35
6	IANA Considerations	35
7	References	36
7.1	Normative References	36
7.2	Informative References	37
	Authors' Addresses	37

1 Introduction

This document defines a YANG [[RFC6020](#)] data model for collecting basic IP and ICMP statistics for monitoring IPv4 and IPv6 implementations.

The data model defines the following IP layer statistics:

- o ICMPv4 system statistics
- o ICMPv6 system statistics
- o IP interface statistics

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

The following terms are defined in [[RFC6241](#)] and are not redefined here:

- o client
- o configuration data
- o server
- o state data
- o statistics

The following terms are defined in [[RFC6020](#)] and are not redefined here:

- o augment
- o data model
- o data node
- o presence container

1.2. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration data (read-write), and "ro" means state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

2 IP Stats Model

This document defines the YANG module "ietf-ip-stats" which provides the following IP layer statistics:

- o ICMPv4 system statistics
- o ICMPv6 system statistics
- o IP interface statistics

The ICMPv4 system statistics are defined in the "icmpv4" container. It provides a small set of statistics about ICMPv4 traffic processed by the entire system. It is an optional node and it MAY be present only if IPv4 is enabled on the system.

The ICMPv6 system statistics are defined in the "icmpv6" container. It provides a small set of statistics about ICMPv6 traffic (including MLDv2 [[RFC 3810](#)]) processed by the entire system. It is an optional node and it MAY be present only if IPv6 is enabled on the system.

Both the "icmpv4" and "icmpv6" containers are child nodes of the top-level "system-stats" container that is intended to hold any system wide statistics. The "system-stats" container has the following tree structure:


```
+--ro system-stats
  +--ro icmpv4!
    | +--ro in-msgs?          yang:counter32
    | +--ro in-errors?       yang:counter32
    | +--ro in-echo-requests? yang:counter32
    | +--ro in-echo-replies? yang:counter32
    | +--ro in-dest-unreachs? yang:counter32
    | +--ro in-param-problems? yang:counter32
    | +--ro in-time-exceeds?  yang:counter32
    | +--ro in-others?       yang:counter32
    | +--ro out-msgs?        yang:counter32
    | +--ro out-errors?      yang:counter32
    | +--ro out-echo-requests? yang:counter32
    | +--ro out-echo-replies? yang:counter32
    | +--ro out-dest-unreachs? yang:counter32
    | +--ro out-param-problems? yang:counter32
    | +--ro out-time-exceeds?  yang:counter32
    | +--ro out-others?      yang:counter32
  +--ro icmpv6!
    +--ro in-msgs?          yang:counter32
    +--ro in-errors?       yang:counter32
    +--ro in-echo-requests? yang:counter32
    +--ro in-echo-replies? yang:counter32
    +--ro in-dest-unreachs? yang:counter32
    +--ro in-param-problems? yang:counter32
    +--ro in-time-exceeds?  yang:counter32
    +--ro in-redirects?     yang:counter32
    +--ro in-pkt-too-bigs?  yang:counter32
    +--ro in-router-solicits? yang:counter32
    +--ro in-router-adverts? yang:counter32
    +--ro in-neighbor-solicits? yang:counter32
    +--ro in-neighbor-adverts? yang:counter32
    +--ro in-MLDv2-queries? yang:counter32
    +--ro in-MLDv2-reports? yang:counter32
    +--ro in-others?       yang:counter32
    +--ro out-msgs?        yang:counter32
    +--ro out-errors?      yang:counter32
    +--ro out-echo-requests? yang:counter32
    +--ro out-echo-replies? yang:counter32
    +--ro out-dest-unreachs? yang:counter32
    +--ro out-param-problems? yang:counter32
    +--ro out-time-exceeds?  yang:counter32
    +--ro out-redirects?     yang:counter32
    +--ro out-pkt-too-bigs?  yang:counter32
    +--ro out-router-solicits? yang:counter32
    +--ro out-router-adverts? yang:counter32
    +--ro out-neighbor-solicits? yang:counter32
    +--ro out-neighbor-adverts? yang:counter32
```



```

+--ro out-MLDv2-queries?      yang:counter32
+--ro out-MLDv2-reports?     yang:counter32
+--ro out-others?            yang:counter32
```

The IP interface statistics are defined in the "stats" containers belongs to a reusable "ip-if-stats" group. Such containers augment the YANG IP Management module [[RFC 7277](#)] and provide a basic set of statistics about IPv4 and IPv6 traffic received on and transmitted by an interface. The statistics are for a specific IP address family. Up to two separate set of IP statistics are supported per interface, one for each IP version. The IP implementation MAY also decides to support a single IP version per interface. The interface can be any IP capable interface defined in the IANA Interface Type YANG module [[RFC 7224](#)]. For instance, it can be:

- o an Ethernet interface (if:type = 'ianaift:ethernetCsmacd')
- o a link aggregate interface (if:type = 'ianaift:ieee8023adLag')
- o a L3 IP VLAN interface (if:type = 'ianaift:l3ipvlan')
- o a loopback interface (if:type = 'ianaift:softwareLoopback')

The "stats" container is an optional node and it MAY be present only if IPv4 and/or IPv6 is enabled on the system. The "stats" containers has the following tree structure:

```
augment /if:interfaces-state/if:interface/ip:ipv4:
  +--ro stats
    +--ro in-receives?      yang:counter64
    +--ro in-octets?        yang:counter64
    +--ro in-hdr-errors?    yang:counter32
    +--ro in-mcast-pkts?    yang:counter64
    +--ro in-mcast-octets?  yang:counter64
    +--ro in-bcast-pkts?    yang:counter64
    +--ro in-truncates?     yang:counter32
    +--ro in-dest-addr-errors? yang:counter32
    +--ro in-scr-addr-errors? yang:counter32
    +--ro in-forwards?      yang:counter64
    +--ro in-no-routes?     yang:counter32
    +--ro in-reasm-requires? yang:counter64
    +--ro in-reasm-fails?   yang:counter32
    +--ro in-reasm-OKs?     yang:counter64
    +--ro in-unknown-protos? yang:counter32
    +--ro in-discards?      yang:counter32
    +--ro in-delivers?      yang:counter64
    +--ro out-requests?     yang:counter64
```



```

    +--ro out-no-routes?          yang:counter32
    +--ro out-forwards?           yang:counter64
    +--ro out-frag-requires?      yang:counter64
    +--ro out-frag-fails?        yang:counter32
    +--ro out-frag-OKs?          yang:counter64
    +--ro out-frag-creates?       yang:counter64
    +--ro out-mcast-pkts?         yang:counter64
    +--ro out-mcast-octets?       yang:counter64
    +--ro out-bcast-pkts?        yang:counter64
    +--ro out-discards?          yang:counter32
    +--ro out-transmits?         yang:counter64
    +--ro out-octets?            yang:counter64
augment /if:interfaces-state/if:interface/ip:ipv6:
  +--ro stats
    +--ro in-receives?           yang:counter64
    +--ro in-octets?             yang:counter64
    +--ro in-hdr-errors?         yang:counter32
    +--ro in-mcast-pkts?         yang:counter64
    +--ro in-mcast-octets?       yang:counter64
    +--ro in-bcast-pkts?        yang:counter64
    +--ro in-truncates?          yang:counter32
    +--ro in-dest-addr-errors?   yang:counter32
    +--ro in-scr-addr-errors?    yang:counter32
    +--ro in-forwards?          yang:counter64
    +--ro in-no-routes?         yang:counter32
    +--ro in-reasm-requires?     yang:counter64
    +--ro in-reasm-fails?       yang:counter32
    +--ro in-reasm-OKs?         yang:counter64
    +--ro in-unknown-protos?     yang:counter32
    +--ro in-discards?          yang:counter32
    +--ro in-delivers?          yang:counter64
    +--ro out-requests?         yang:counter64
    +--ro out-no-routes?        yang:counter32
    +--ro out-forwards?         yang:counter64
    +--ro out-frag-requires?     yang:counter64
    +--ro out-frag-fails?       yang:counter32
    +--ro out-frag-OKs?         yang:counter64
    +--ro out-frag-creates?     yang:counter64
    +--ro out-mcast-pkts?       yang:counter64
    +--ro out-mcast-octets?     yang:counter64
    +--ro out-bcast-pkts?       yang:counter64
    +--ro out-discards?         yang:counter32
    +--ro out-transmits?        yang:counter64
    +--ro out-octets?           yang:counter64
```


3 Relationship to IP-MIB

The IP-MIB defines two ICMP statistic tables: icmpStatsTable and icmpMsgStatsTable. The icmpStatsTable contains generic system-wide ICMP counters for each IP version. The icmpMsgStatsTable contains system-wide per-version, per-message type ICMP counters. The YANG IP statistics module aggregates all ICMP statistics into one container for each IP version. This module does not individually track all possible ICMP type values. It only tracks the most common ICMP types for each IP version and aggregate all other ICMP types into a single counter.

The IP-MIB defines two IP statistics tables: ipSystemStatsTable and ipIfStatsTable). The ipSystemStatsTable contain objects to count the number of packets, fragments, datagrams and octets that a given system has processed for each IP version. The ipIfStatsTable contain objects to count the number of packets, fragments, datagrams and octets that a given interface has received or transmitted for each IP version. The data nodes defined in the "ip-if-stats" group corresponds to the counters defined in ipIfStatsTable. The system wide table ipSystemStatsTable is not defined in this data model.

The following tables list the YANG data nodes with corresponding objects in the IP-MIB.

YANG data node in	IP-MIB Object
/ip-stats:system-stats	
icmpv4/in-msgs	icmpStatsInMsgs
icmpv4/in-errors	icmpStatsInErrors
icmpv4/in-echo-requests	
icmpv4/in-echo-replies	
icmpv4/in-dest-unreachs	
icmpv4/in-param-problems	
icmpv4/in-time-exceeds	
icmpv4/in-others	
icmpv4/out-msgs	icmpStatsOutMsgs
icmpv4/out-errors	icmpStatsOutErrors
icmpv4/out-echo-requests	
icmpv4/out-echo-replies	
icmpv4/out-dest-unreachs	
icmpv4/out-param-problems	
icmpv4/out-time-exceeds	
icmpv4/out-others	

Table 1: YANG ICMPv4 Stats Nodes to IP-MIB Objects

YANG data node in	IP-MIB Object
/ip-stats:system-stats	
icmpv6/in-msgs	icmpStatsInMsgs
icmpv6/in-errors	icmpStatsInErrors
icmpv6/in-echo-requests	
icmpv6/in-echo-replies	
icmpv6/in-dest-unreachs	
icmpv6/in-param-problems	
icmpv6/in-time-exceeds	
icmpv6/in-redirects	
icmpv6/in-pkt-too-bigs	
icmpv6/in-router-solicits	
icmpv6/in-router-adverts	
icmpv6/in-neighbor-solicits	
icmpv6/in-neighbor-adverts	
icmpv6/in-MLDv2-queries	
icmpv6/in-MLDv2-reports	
icmpv6/in-others	
icmpv6/out-msgs	icmpStatsInMsgs
icmpv6/out-errors	icmpStatsInErrors
icmpv6/out-echo-requests	
icmpv6/out-echo-replies	
icmpv6/out-dest-unreachs	
icmpv6/out-param-problems	
icmpv6/out-time-exceeds	
icmpv6/out-redirects	
icmpv6/out-pkt-too-bigs	
icmpv6/out-router-solicits	
icmpv6/out-router-adverts	
icmpv6/out-neighbor-solicits	
icmpv6/out-neighbor-adverts	
icmpv6/out-MLDv2-queries	
icmpv6/out-MLDv2-reports	

Table 2: YANG ICMPv6 Stats Nodes to IP-MIB Objects

YANG data node in /if:interface-state/if:interface	IP-MIB Object
{ipv4,ipv6}/stats/in-receives	ipIfStatsHCInReceives
{ipv4,ipv6}/stats/in-octets	ipIfStatsHCInOctets
{ipv4,ipv6}/stats/in-hdr-errors	ipIfStatsInHdrErrors
{ipv4,ipv6}/stats/in-mcast-pkts	ipIfStatsHCInMcastPkts
{ipv4,ipv6}/stats/in-mcast-octets	ipIfStatsHCInMcastOctets
{ipv4,ipv6}/stats/in-bcast-pkts	ipIfStatsHCInBcastPkts
{ipv4,ipv6}/stats/in-truncates	ipIfStatsInTruncatedPkts
{ipv4,ipv6}/stats/in-dest-addr-errors	ipIfStatsInAddrErrors
{ipv4,ipv6}/stats/in-scr-addr-errors	
{ipv4,ipv6}/stats/in-forwards	ipIfStatsHCInForwDatagrams
{ipv4,ipv6}/stats/in-no-routes	ipIfStatsInNoRoutes
{ipv4,ipv6}/stats/in-reasm-requires	ipIfStatsReasmReqds
{ipv4,ipv6}/stats/in-reasm-fails	ipIfStatsReasmFails
{ipv4,ipv6}/stats/in-reasm-OKs	ipIfStatsReasmOKs
{ipv4,ipv6}/stats/in-unknown-protos	ipIfStatsInUnknownProtos
{ipv4,ipv6}/stats/in-discards	ipIfStatsInDiscards
{ipv4,ipv6}/stats/in-delivers	ipIfStatsHCInDelivers
{ipv4,ipv6}/stats/out-requests	ipIfStatsHCOutRequests
{ipv4,ipv6}/stats/out-no-routes	
{ipv4,ipv6}/stats/out-forwards	ipIfStatsHCOutForwDatagrams
{ipv4,ipv6}/stats/out-frag-requires	ipIfStatsOutFragReqds
{ipv4,ipv6}/stats/out-frag-fails	ipIfStatsOutFragFails
{ipv4,ipv6}/stats/out-frag-OKs	ipIfStatsOutFragOKs
{ipv4,ipv6}/stats/out-frag-creates	ipIfStatsOutFragCreates
{ipv4,ipv6}/stats/out-mcast-pkts	ipIfStatsHCOutMcastPkts
{ipv4,ipv6}/stats/out-mcast-octets	ipIfStatsHCOutMcastOctets
{ipv4,ipv6}/stats/out-bcast-pkts	ipIfStatsHCOutBcastPkts
{ipv4,ipv6}/stats/out-discards	ipIfStatsOutDiscards
{ipv4,ipv6}/stats/out-transmits	ipIfStatsHCOutTransmits
{ipv4,ipv6}/stats/out-octets	ipIfStatsHCOutOctets

Table 3: YANG IP Stats Nodes to IP-MIB Objects

4 IP Statistics YANG Module

This module imports typedefs from [RFC6991] and [RFC7223], and it references [RFC0791], [RFC0826], [RFC2460], [RFC4861], [RFC4862], [RFC4941], and [RFC7217].

```
<CODE BEGINS> file "ietf-ip-stats@2015-04-27.yang"

module ietf-ip-stats {

    namespace "urn:ietf:params:xml:ns:yang:ietf-ip-stats";
    prefix ip-stats;

    import ietf-ip {
        prefix ip;
    }
    import ietf-interfaces {
        prefix if;
    }
    import ietf-yang-types {
        prefix yang;
    }

    organization
        "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

    contact
        "WG Web:  <http://tools.ietf.org/wg/netmod/>
        WG List:  <mailto:netmod@ietf.org>

        WG Chair: Thomas Nadeau
                  <mailto:tnadeau@lucidvision.com>

        WG Chair: Juergen Schoenwaelder
                  <mailto:j.schoenwaelder@jacobs-university.de>

        Editor:   Steve Baillargeon
                  <mailto:steve.baillargeon@ericsson.com>";

    description
        "This YANG module contains a collection of IP and ICMP statistics
        for monitoring IPv4 and IPv6 implementations.

        Copyright (c) 2015 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 (<http://trustee.ietf.org/license-info>).

This version of this YANG module is part of TBD; see the RFC itself for full legal notices.";

```
revision 2015-04-27 {
  description
    "Initial revision.";
  reference
    "TBD";
}

/*
 * Groupings
 */

grouping ip-if-stats {
  description
    "A group of IP layer statistics for IP traffic
    received and transmitted by an interface.";

  container stats {
    config false;
    description
      "A collection of IP layer statistics for IP traffic
      received and transmitted by an interface.";

    leaf in-receives {
      type yang:counter64;
      units packets;
      description
        "The number of input IP packets received on the
        interface, including those received in error.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        'discontinuity-time'.";
      reference
        "RFC 4293: The IP MIB - ipIfStatsHCInReceives";
    }

    leaf in-octets {
      type yang:counter64;
```



```
    units octets;
    description
      "The number of octets received in IP packets,
       including those received in error. Octets from packets
       counted in in-receives MUST be counted.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 4293: The IP MIB - ipIfStatsHCInOctets";
  }

  leaf in-hdr-errors {
    type yang:counter32;
    units packets;
    description
      "The number of input IP packets discarded due to errors
       in their IP headers, including version number mismatch,
       other format errors, hop count exceeded, errors
       discovered in processing their IP options, etc.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 4293: The IP MIB - ipIfStatsInHdrErrors";
  }

  leaf in-mcast-pkts {
    type yang:counter64;
    units packets;
    description
      "The number of input IP multicast packets received on
       the interface.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 4293: The IP MIB - ipIfStatsHCInMcastPkts";
  }

  leaf in-mcast-octets {
    type yang:counter64;
```



```
    units octets;
    description
      "The number of octets received in IP multicast
       packets. Octets from packets counted in
       /ipv4/in-mcast-pkts MUST be counted.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 4293: The IP MIB - ipIfStatsHCInMcastOctets";
  }

  leaf in-bcast-pkts {
    type yang:counter64;
    units packets;
    description
      "The number of input IP broadcast packets received on
       the interface.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 4293: The IP MIB - ipIfStatsHCInBcastPkts";
  }

  leaf in-truncates {
    type yang:counter32;
    units packets;
    description
      "The number of input IP packets discarded because the
       packet didn't carry enough data.

       Discontinuities in the value of this counter can occur
       at re-initialization of the management system, and at
       other times as indicated by the value of
       'discontinuity-time'.";
    reference
      "RFC 4293: The IP MIB - ipIfStatsInTruncatedPkts";
  }

  leaf in-dest-addr-errors {
    type yang:counter32;
    units packets;
    description
```


"The number of input IP packets discarded because the IP address in their IP header's destination field was not a valid address to be received at this system. For systems that are not IP routers and therefore do not forward packets,

 this counter includes packets that are discarded because the destination address was not a local address.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

reference

[RFC 4293](#): The IP MIB - ipIfStatsInAddrErrors";

}

leaf in-scr-addr-errors {

 type yang:counter32;

 units packets;

 description

 "The number of input IP packets discarded because the IP address in their IP header's source field was not a valid address to be received at this system. This count includes source filtering on multicast packets.

 Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

 reference

[RFC 4293](#): The IP MIB - ipIfStatsInAddrErrors";

}

leaf in-forwards {

 type yang:counter64;

 units packets;

 description

 "The number of input IP packets for which this system was not their final IP destination and for which this system attempted to find a route to forward them to that final destination.

 Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

 reference

[RFC 4293](#): The IP MIB - ipIfStatsHCInForwDatagrams";

}

Baillargeon

Expires October 29, 2015

[Page 15]

```
leaf in-no-routes {
  type yang:counter32;
  units packets;
  description
    "The number of input IP packets for which this system was
    not their final IP destination and for which this system
    attempted to find a route to forward them to that final
    destination.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 4293: The IP MIB - ipIfStatsInNoRoutes";
}

leaf in-reasm-requires {
  type yang:counter64;
  units fragments;
  description
    "The number of input IP fragments received that needed
    to be reassembled at this interface.

    When tracking interface statistics, the counter of the
    interface to which these fragments were addressed is
    incremented. This interface might not be the same as the
    input interface for some of the fragments.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 4293: The IP MIB - ipIfStatsReasmReqds";
}

leaf in-reasm-fails {
  type yang:counter32;
  units failures;
  description
    "The number of failures detected by the IP re-assembly
    algorithm (for whatever reason: timed out, errors, etc.).
    Note that this is not necessarily a count of discarded IP
    fragments since some algorithms can lose track of the
    number of fragments by combining them as they are
    received."
```


When tracking interface statistics, the counter of the interface to which these fragments were addressed is incremented. This interface might not be the same as the input interface for some of the fragments.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 4293](#): The IP MIB - ipIfStatsReasmFails";

}

leaf in-reasm-OKs {

type yang:counter64;

units datagrams;

description

"The number of input IP datagrams successfully reassembled.

When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 4293](#): The IP MIB - ipIfStatsReasmOKs";

}

leaf in-unknown-protos {

type yang:counter32;

units datagrams;

description

"The number of locally-addressed input IP datagrams received successfully but discarded because of an unknown or unsupported protocol.

When tracking interface statistics, the counter of the interface to which these datagrams were addressed is incremented. This interface might not be the same as the input interface for some of the datagrams.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at


```
        other times as indicated by the value of
        'discontinuity-time'.";
    reference
        "RFC 4293: The IP MIB - ipIfStatsInUnknownProtos";
}

leaf in-discards {
    type yang:counter32;
    units datagrams;
    description
        "The number of input IP datagrams for which no problems
        were encountered to prevent their continued processing,
        but were discarded (e.g., for lack of buffer space).
        Note that this counter does not include any datagrams
        discarded while awaiting re-assembly.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        'discontinuity-time'.";
    reference
        "RFC 4293: The IP MIB - ipIfStatsInDiscards";
}

leaf in-delivers {
    type yang:counter64;
    units datagrams;
    description
        "The number of input IP datagrams successfully
        delivered to IP user protocols (including ICMP).

        When tracking interface statistics, the counter of the
        interface to which these datagrams were addressed is
        incremented. This interface might not be the same as the
        input interface for some of the datagrams.

        Discontinuities in the value of this counter can occur
        at re-initialization of the management system, and at
        other times as indicated by the value of
        'discontinuity-time'.";
    reference
        "RFC 4293: The IP MIB - ipIfStatsHCInDelivers";
}

leaf out-requests {
    type yang:counter64;
    units datagrams;
    description
```


"The number of output IP datagrams that local IP user protocols (including ICMP) supplied to IP in requests for transmission. Note that this counter does not include any datagrams counted out-forwards.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 4293](#): The IP MIB - ipIfStatsHCOutRequests";

}

leaf out-no-routes {
 type yang:counter32;
 units datagrams;
 description

"The number of locally generated output IP datagrams discarded because no route could be found to transmit them to their destination.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 4293](#): The IP MIB";

}

leaf out-forwards {
 type yang:counter64;
 units packets;
 description

"The number of output IP packets for which this system was not their final IP destination and for which it was successful in finding a path to their final destination.

When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully forwarded packet.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 4293](#): The IP MIB - ipIfStatsHCOutForwDatagrams";

}


```
leaf out-frag-requires {
  type yang:counter64;
  units datagrams;
  description
    "The number of output IP datagrams that would require
    fragmentation in order to be transmitted.

    When tracking interface statistics, the counter of the
    outgoing interface is incremented for a successfully
    fragmented datagram.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 4293: The IP MIB - ipIfStatsOutFragReqs";
}

leaf out-frag-fails {
  type yang:counter32;
  units datagrams;
  description
    "The number of output IP datagrams that have been
    discarded because they needed to be fragmented but
    could not be. This includes IPv4 packets that have
    the DF bit set or IPv6 packets that are being
    forwarded and exceed the outgoing link MTU.

    When tracking interface statistics, the counter of the
    outgoing interface is incremented for a successfully
    fragmented datagram.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 4293: The IP MIB - ipIfStatsOutFragFails";
}

leaf out-frag-OKs {
  type yang:counter64;
  units datagrams;
  description
    "The number of output IP datagrams that have been
    successfully fragmented."
```


When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 4293](#): The IP MIB - ipIfStatsOutFragOKs";

}

leaf out-frag-creates {

type yang:counter64;

units fragments;

description

"The number of output IP fragments that have been generated as a result of IP fragmentation.

When tracking interface statistics, the counter of the outgoing interface is incremented for a successfully fragmented datagram.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 4293](#): The IP MIB - ipIfStatsOutFragCreates";

}

leaf out-mcast-pkts {

type yang:counter64;

units packets;

description

"The number of output IP multicast packets transmitted on the interface.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of 'discontinuity-time'.";

reference

"[RFC 4293](#): The IP MIB - ipIfStatsHCOutMcastPkts";

}

leaf out-mcast-octets {

type yang:counter64;


```
units octets;
description
  "The number of octets transmitted in IP
  multicast packets. Octets from packets
  counted in /ipv4/out-mcast-pkts MUST be counted.

  Discontinuities in the value of this counter can occur
  at re-initialization of the management system, and at
  other times as indicated by the value of
  'discontinuity-time'.";
reference
  "RFC 4293: The IP MIB - ipIfStatsHCOctets";
}

leaf out-bcast-pkts {
  type yang:counter64;
  units packets;
  description
    "The number of output IP broadcast packets transmitted
    on the interface.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 4293: The IP MIB - ipIfStatsHCOctets";
}

leaf out-discards {
  type yang:counter32;
  units packets;
  description
    "The number of output IP packets for which no problem was
    encountered to prevent their transmission to their
    destination, but were discarded (e.g., for lack of
    buffer space). Note that this counter would include
    datagrams counted in out-forwards if any such
    packets met this (discretionary) discard criterion.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 4293: The IP MIB - ipIfStatsOutDiscards";
}
```



```
leaf out-transmits {
  type yang:counter64;
  units packets;
  description
    "The number of output IP packets that this system
    supplied to the lower layers for transmission.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 4293: The IP MIB - ipIfStatsHCOutTransmits";
}

leaf out-octets {
  type yang:counter64;
  units octets;
  description
    "The number of octets in output IP packets delivered
    to the lower layers for transmission. Octets from packets
    counted in out-transmits MUST be counted.

    Discontinuities in the value of this counter can occur
    at re-initialization of the management system, and at
    other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 4293: The IP MIB - ipIfStatsHCOutOctets";
}
}

/*
 * Data nodes for ICMP statistics for IP-capable system
 */

container system-stats {
  config false;
  description
    "A collection of system wide statistics";

  container icmpv4 {
    presence
      "Present if IPv4 is enabled on this system";
    description
      "A collection of ICMPv4 layer statistics processed by
      this system.";
```



```
leaf in-msgs {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv4 messages received by this system.
     Note that this counter includes all those counted by
     /icmpv4/in-errors.";
  reference
    "RFC 4293: The IP MIB - icmpStatsInMsgs";
}

leaf in-errors {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv4 messages that this system received
     but determined as having ICMP-specific errors (bad ICMP
     checksums, bad length, etc).";
  reference
    "RFC 4293: The IP MIB - icmpStatsInErrors";
}

leaf in-echo-requests {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv4 Echo Request messages
     (type 8) received by this system.";
  reference
    "RFC 4293: The IP MIB";
}

leaf in-echo-replies {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv4 Echo Reply messages
     (type 0) received by this system.";
  reference
    "RFC 4293: The IP MIB";
}

leaf in-dest-unreaches {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv4 Destination Unreachable messages
     (type 3) received by this system.";
```



```
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-param-problems {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv4 Parameter Problem messages
        (type 12) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-time-exceeds {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv4 Time Exceeded messages
        (type 11) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-others {
    type yang:counter32;
    units messages;
    description
      "The number of any other ICMPv4 message types
        received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-msgs {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv4 messages this system attempted to
        send. Note that this counter includes all those counted
        by /icmpv4/out-errors.";
    reference
      "RFC 4293: The IP MIB - icmpStatsOutMsgs";
  }

  leaf out-errors {
    type yang:counter32;
    units messages;
```



```
    description
      "The number of ICMPv4 messages this system did not send
      due to problems discovered within ICMP, such as a lack of
      buffers. This value should not include errors discovered
      outside the ICMP layer, such as the inability of IP to
      route the resultant datagram.";
    reference
      "RFC 4293: The IP MIB - icmpStatsOutErrors";
  }

  leaf out-echo-requests {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv4 Echo Request messages
      (type 8) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-echo-replies {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv4 Echo Reply messages
      (type 0) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-dest-unreachs {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv4 Destination Unreachable messages
      (type 3) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-param-problems {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv4 Parameter Problem messages
      (type 12) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }
```



```
}

leaf out-time-exceeds {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv4 Time Exceeded messages
    (type 11) this system attempted to send.";
  reference
    "RFC 4293: The IP MIB";
}

leaf out-others {
  type yang:counter32;
  units messages;
  description
    "The number of any other ICMPv4 message types this
    system attempted to send.";
  reference
    "RFC 4293: The IP MIB";
}
}

container icmpv6 {
  presence
    "Present if IPv6 is enabled on this system";
  description
    "A collection of ICMPv6 layer statistics for ICMPv6 traffic
    processed by this system.";

  leaf in-msgs {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 messages received by this system.
      Note that this counter includes all those counted by
      icmpv6/in-errors.";
    reference
      "RFC 4293: The IP MIB - icmpStatsInMsgs";
  }

  leaf in-errors {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 messages that this system received
      but determined as having ICMP-specific errors (bad ICMP
      checksums, bad length, etc).";
```



```
    reference
      "RFC 4293: The IP MIB - icmpStatsInErrors";
  }

  leaf in-echo-requests {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Echo Request messages
        (type 128) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-echo-replies {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Echo Reply messages
        (type 129) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-dest-unreaches {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Destination Unreachable messages
        (type 1) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-param-problems {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Parameter Problem messages
        (type 4) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-time-exceeds {
    type yang:counter32;
    units messages;
    description
```



```
    "The number of ICMPv6 Time Exceeded messages
      (type 3) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-redirects {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Redirect messages
        (type 137) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-pkt-too-bigs {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Packet Too Big messages
        (type 2) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-router-solicits {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv4 Router Solicit messages
        (type 133) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-router-adverts {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv4 Router Advertisement messages
        (type 134) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-neighbor-solicits {
    type yang:counter32;
```



```
    units messages;
    description
      "The number of ICMPv6 Neighbor Solicit messages
        (type 135) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-neighbor-adverts {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Neighbor Advertisement messages
        (type 136) received by this system.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf in-MLDv2-queries {
    type yang:counter32;
    units messages;
    description
      "The number of Multicast Listener Discovery version 2
        Query messages(type 130 with length equal or greater
        than 28 bytes) received by this system.";
    reference
      "RFC 3810: MLDv2 for IPv6";
  }

  leaf in-MLDv2-reports {
    type yang:counter32;
    units messages;
    description
      "The number of Multicast Listener Discovery version 2
        Report messages(type 143) received by this system.";
    reference
      "RFC 3810: MLDv2 for IPv6";
  }

  leaf in-others {
    type yang:counter32;
    units messages;
    description
      "The number of any other ICMPv6 message types received
        by this system.";
    reference
      "RFC 4293: The IP MIB";
  }
```



```
leaf out-msgs {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv6 messages this system attempted to
      send. Note that this counter includes all those counted
      by /icmpv6/out-errors.";
  reference
    "RFC 4293: The IP MIB - icmpStatsInMsgs";
}

leaf out-errors {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv6 messages that this system received
      but determined as having ICMP-specific errors (bad ICMP
      checksums, bad length, etc).";
  reference
    "RFC 4293: The IP MIB - icmpStatsInErrors";
}

leaf out-echo-requests {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv6 Echo Request messages
      (type 128) this system attempted to send.";
  reference
    "RFC 4293: The IP MIB";
}

leaf out-echo-replies {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv6 Echo Reply messages
      (type 129) this system attempted to send.";
  reference
    "RFC 4293: The IP MIB";
}

leaf out-dest-unreachs {
  type yang:counter32;
  units messages;
  description
    "The number of ICMPv6 Destination Unreachable messages
      (type 1) this system attempted to send.";
```



```
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-param-problems {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Parameter Problem messages
        (type 4) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-time-exceeds {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Time Exceeded messages
        (type 3) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-redirects {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Redirect messages
        (type 137) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-pkt-too-bigs {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Packet Too Big messages
        (type 2) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-router-solicits {
    type yang:counter32;
    units messages;
    description
```



```
    "The number of ICMPv6 Router Solicit messages
      (type 133) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-router-adverts {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Router Advertisement messages
        (type 134) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-neighbor-solicits {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Neighbor Solicit messages
        (type 135) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-neighbor-adverts {
    type yang:counter32;
    units messages;
    description
      "The number of ICMPv6 Neighbor Advertisement messages
        (type 136) this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }

  leaf out-MLDv2-queries {
    type yang:counter32;
    units messages;
    description
      "The number of Multicast Listener Discovery version 2
        Query messages(type 130 with length equal or greater
        than 28 bytes) this system attempted to send.";
    reference
      "RFC 3810: MLDv2 for IPv6";
  }

  leaf out-MLDv2-reports {
```



```
    type yang:counter32;
    units messages;
    description
      "The number of Multicast Listener Discovery version 2
       Report messages(type 143) this system attempted to send.";
    reference
      "RFC 3810: MLDv2 for IPv6";
  }

  leaf out-others {
    type yang:counter32;
    units messages;
    description
      "The number of any other ICMPv6 message types
       this system attempted to send.";
    reference
      "RFC 4293: The IP MIB";
  }
}
/*
 * Data nodes for IP statistics for IPv4-capable interface
 */

augment "/if:interfaces-state/if:interface/ip:ipv4" {
  description
    "Data nodes for statistics about IPv4 traffic on an
     interface when IPv4 is enabled on that interface.

     The interface can be any IPv4-capable interface including
     an Ethernet interface, a link aggregate (LAG) interface,
     a L3 IP VLAN interface and a loopback interface.";

  uses ip-if-stats;
}

/*
 * Data nodes for IP statistics for IPv6-capable interface
 */

augment "/if:interfaces-state/if:interface/ip:ipv6" {
  description
    "Data nodes for statistics about IPv6 traffic on an
     interface when IPv6 is enabled on that interface.

     The interface can be any IPv6-capable interface including
     an Ethernet interface, a link aggregate (LAG) interface,
     a L3 IP VLAN interface and a loopback interface.";
```



```
    uses ip-if-stats;
  }
}
```

<CODE ENDS>

5 Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [[RFC6241](#)]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [[RFC6242](#)]. The NETCONF access control model [[RFC6536](#)] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

There are a number of data nodes defined in the YANG module which are readable IP statistics (read-only counters) with config false statement. Even though these data nodes are generally not considered sensitive or vulnerable, it may be desirable to limit the number of interfaces with IP statistic collection in some network environments. These are the subtrees and data nodes and their sensitivity/vulnerability:

{ipv4|ipv6}/stats: These leafs are used to collect IP statistics about a specific interface. Enabling or collecting IP statistics on all IP capable interfaces on a system may have a negative impact on its performance or data storage capacity. The IP statistics may also contain sensitive operational information about the usage of the network.

6 IANA Considerations

This document registers a URI in the "IETF XML Registry" [[RFC3688](#)]. Following the format in [RFC 3688](#), the following registration has been made.

URI: urn:ietf:params:xml:ns:yang:ietf-ip-stats

Registrant Contact: The NETMOD WG of the IETF.

XML: N/A; the requested URI is an XML namespace.

This document registers a YANG module in the "YANG Module Names" registry [[RFC6020](#)].

Name: ietf-ip-stats
Namespace: urn:ietf:params:xml:ns:yang:ietf-ip-stats
Prefix: ip-stats
Reference: RFC XXXX

[7](#) References

[7.1](#) Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, [RFC 791](#), September 1981.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), January 2004.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), September 2007.
- [RFC3810] Vida, R., Costa, L., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", [RFC 3810](#), June 2004.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", [RFC 6241](#), June 2011.
- [RFC6991] Schoenwaelder, J., "Common YANG Data Types", [RFC 6991](#), July 2013.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 7223](#), May 2014.
- [RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", [RFC 7277](#), June 2014.
- [XML] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation, November 2008,

<<http://www.w3.org/TR/xml/>>.

7.2 Informative References

- [RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", [RFC 7224](#), May 2014.
- [RFC4293] Routhier, S., "Management Information Base for the Protocol (IP)", [RFC 4293](#), April 2006.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), June 2011.

Authors' Addresses

Steve Baillargeon
349 Terry Fox Drive
Kanata, Ontario, K2K 2V6
Canada

EMail: steve.baillargeon@ericsson.com

