LAN Interfaces Configuration Guide Addendum

Contents

Copyright Statement

About This Guide

This addendum describes LAN interfaces functionality that was updated on the AT&T Vyatta vRouter (referred to as a virtual router, vRouter, or router in the guide).

Features	vRouter Release	Rev of LAN Interfaces Configuration Guide 5600 Addendum	Date
TCP MSS Clamping	1801	1	Dec 2018

TCP MSS

The TCP protocol has a concept of *Maximum Segment Size* (*MSS)*. MSS is simply the amount of useful data in a packet, or maximum transmission unit (MTU) minus the IP and TCP headers.

A maximum transmission unit (MTU) is the largest packet that can be sent on a network. The standard Ethernet MTU is 1500 bytes.

However, higher level protocols may want to create packets larger than an interfaces MTU.

To get around this issue, IPv4 allows fragmentation which divides a datagram into pieces

Fragmentation is not supported by all devices and application, and therefore, it is best avoided.

The best way to avoid fragmentation is to adjust the TCP maximum segment size (MSS). MSS is the amount of useful data in a packet (payload), or maximum transmission unit (MTU) minus the IP and TCP headers
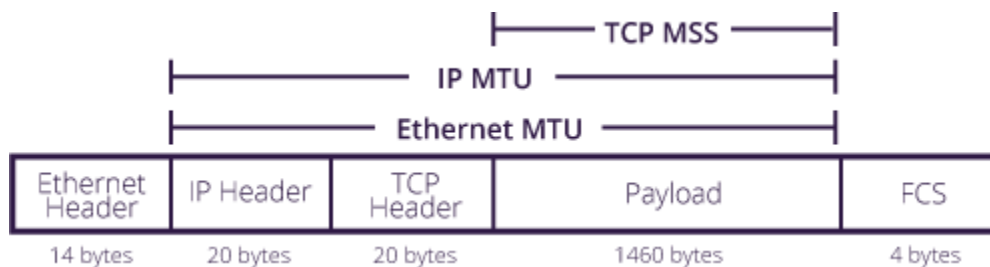
Note: IPv4 fragmentation issues have become more prevalent, because IP4 tunnels are more widely deployed and the tunnel encapsulation adds more "overhead" to the size of a packet.

Advantages of TCP MSS

TCP MSS has a number of advantages over the Path MTU Discovery (PMTUD) method, which relied on transmitting multiple ICMP "Fragmentation Needed" (Type 3, Code 4) messages. Notably, many network devices block *all* ICMP messages.

Characteristics of TCP MSS

- MSS is a parameter in the TCP options field, which comes between the TCP header and the payload (data) of the packet.
- It specifies the largest amount of data that a device can receive in a single TCP segment.
- It does **not** include the IP header and TCP header.
- TCP MSS is an announcement, not a negotiation.
- TCP MSS option is sent during the three-way handshake by both sides.
- TCP MSS can be different in each direction.

TCP MSS Clamping

TCP MSS clamping is a process where network devices rewrite the MSS value in TCP handshakes.

The example below shows a 24-byte GRE header added to a packet. In this case, TCP MSS should be clamped to 1436.

IP	TCP	Data
20 bytes	20 bytes	1460 bytes

GRE encapsulation

IP	GRE	IP	TCP	Data
20 bytes	24 bytes	20 bytes	20 bytes	1460 bytes
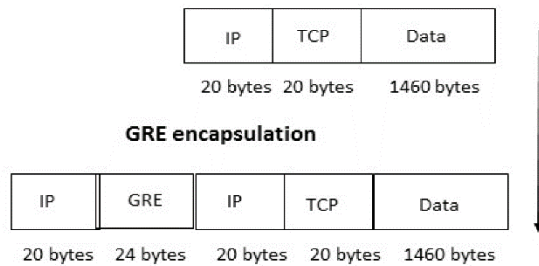
TCP MSS Clamping on vRouter

When configured on vRouter, clamping is performed on TCP MSS SYN and SYN-ACK packets entering and leaving the interface.

Note: The TCP MSS option is not an obligatory part of the TCP packets. The vRouter can adjust the TCP MSS option value only *if* the MSS option is present in the packet However, most TCP devices *do include* the MSS option.

Clamping can be applied as follows:

- Can be configured per interface
- Can be configured independently for IPv4 and IPv6
- Supports the following options::

Options	Example	Command	Recommended Usage
MTU Maximum MSS Based on the interface MTU (MTU less default IP and TCP header sizes).	If the interface MTU was 1500, MSS is clamped to 1460.	`ip tcp-mss mtu`	If there is no additional encapsulation on a packet's path, use this option.
MTU Minus Based on the interface MTU (MTU less default IP and TCP header	If the MTU was 1500, MSS is clamped to 1436.	`ip tcp-mss mtu-minus 24`	If PPPoE is used, use this option. PPPoE requires an additional 8 bytes

Options	Example	Command	Recommended Usage
sizes, and with an additional value subtracted.			and truncates the Ethernet MTU to 1492 (1500 –8). PPPoE is often used with an Asymmetric digital subscriber line (ADSL).
Specific Value The user configures a specific value.	MSS clamped to 1436	`ip tcp-mss limit 1436`	

TCP MSS Clamping on vRouter, Supported Interfaces

TCP MSS clamping can be configured on the following types of interfaces:

- Dataplane (with and without vif)
- Bridge
- L2tpeth (with and without vif)
- Openvpn
- Tunnel
- Vti
- Bonding (with and without vif)
- S2s-fp (Only supports clamping to aspecific value.). \

Encapsulation Overhead

The amount of protocol overhead varies based on the encapsulation type.

- GRE adds 24 bytes (20 byte IPv4, 4 byte GRE)
- 6in4 encapsulation adds 20 bytes
- 4in6 encapsulation adds 20 bytes
- MPLS adds 4 bytes for each label in the stack
- 802.1Q adds 4 bytes (Q-in-Q would add 8 bytes)
- VXLAN adds 50 bytes
- PPPoE adds 8 bytes

Commands for configuring TCP MSS Clamping

Specify the following CLI (nder existing "`interface <iftype> <ifname>`"):

Command	Meaning
`ip tcp-mss mtu`	For IPv4, sets TCP Maximum MSS to interface MTU less default IP and TCP header sizes
`ipv6 tcp-mss mtu`	For IPv6, sets TCP Maximum MSS to interface MTU less default IP and TCP header sizes
`ip tcp-mss mtu-minus` *value*	For IPv4, sets TCP MSS to interface MTU less default IP and TCP header sizes, and with an additional value subtracted
`ipv6 tcp-mss mtu-minus` *value*	For IPv6, sets TCP MSS to interface MTU less default IP and TCP header sizes, and with an additional value subtracted.
`ip tcp-mss limit` *value*	For IPv4, sets TCP MSS to a specific value.
`ipv6 tcp-mss limit` *value*	For Ipv6, sets TCP MSS to a specific value.

Values range from 1 to 65535.

Note: For a given interface and protocol (IPv4 or IPv6), only one type of MSS clamping can be configured.