Troubleshoot Output Drops on Catalyst 9000 Switches

Contents

Introduction

Prerequisites

- Requirements
- Components Used

Background Information

What are Output Drops

- Types of Congestion
- Congestion with Low Throughput
- Validate Buffer Congestion

Modify Buffers to Resolve Output Drops

- SoftMax Multiplier
- Per-Queue Buffer Modification
- Alternative Methods to Manage Congestion

Analyze Output Drops with Wireshark

- View the I/O Rate
- View the I/O Rate in Milliseconds

Introduction

This document describes how to troubleshoot output drops on the Catalyst 9000 series platforms.

Prerequisites

Requirements

To troubleshoot Quality of Service (QoS) on the Catalyst 9000 series platforms you must understand:

- Standard QoS Concepts
- Modular QoS Command Line Interface (CLI)

Components Used

The information in this document is based on the this hardware and software version, but the methodology and majority of commands can be applied to other Catalyst 9000 series switches on other code:

- Cisco Catalyst 9300
- Cisco IOS® XE 16.12.3

Note: Consult the appropriate configuration guide for the commands that are used in order to enable these features on other Cisco platforms.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

For an in-depth explanation of QoS on the Catalyst 9000 series platforms, which includes default QoS configurations, queue structure and buffer explanations, see the Catalyst 9000 QoS and Queueing White Paper Review the recommended release guide to ensure you are on the latest recommended software for your platform. These recommendations ensure your software is supported, and help avoid known bugs in older codes. Recommended Releases for Catalysts

What are Output Drops

Knowledge of buffer allocation can help you understand how buffer congestion results in output drops. Congestion occurs when the destination interface has a number of packets that exceed its output rate. These packets must be stored in the buffer until they can be transmitted. Consider that these switches have at most 36 MB of buffers per ASIC, which is then shared among all the ports on the ASIC. While an egress interface can be able to empty that buffer at line rate, any scenario that causes packets to be buffered at a greater rate can cause congestion. Congestion can occur even if that traffic burst only lasts a fraction of a second, and can cause latency in the traffic, or output drops if that buffer were to fill completely.

Note: The output drop counter displayed in show interface is presented in bytes by default. In release 16.9.3 and later these counters are packets by default.

Types of Congestion

As shown in Image 1, there are two types of congestion.

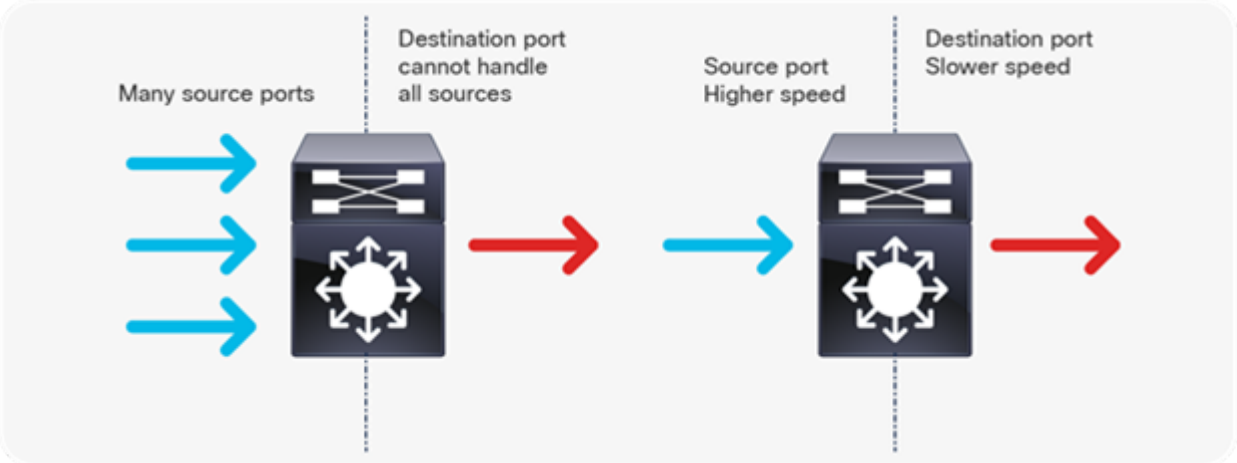


Image 1. Types of Congestion

The two types of congestion shown in Image 1 are:

- Many to one: When multiple source ports send traffic toward a single destination at the same time, the destination port can get congested with the amount of traffic it has received from multiple sources.

- Speed mismatch: When a port with higher speed transmits to a port with lower speed (for example 10 Gbps to 1 Gbps), packets must take time to drain out of the egress port, which can result in delay and/or packet drops.

Congestion with Low Throughput

Traffic bursts can cause output drops even when the interface output rate is significantly lower than the maximum interface capacity. By default, the output rates in the **show interface** command are averaged over five minutes, which is not adequate to capture any short-lived bursts. It is best to average them over 30 seconds, though even in this scenario a burst of traffic for milliseconds could result in output drops that do not cause the 30 second average rate to increase. This document can be used to troubleshoot any other type of congestion you see on your Catalyst 9000 series switch.

Validate Buffer Congestion

There are two commands used to validate buffer congestion. The first command is **show platform hardware fed switch active qos queue config interface <interface>**. This command allows you to see the current buffer allocation on the port, as shown in in Image 2.

```
9300#
```

show platform hardware fed switch active qos queue config interface gigabitEthernet 1/0/48

```
Asic:0 Core:0 DATA Port:47 GPN:48 LinkSpeed:0x1
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 376 - 383
  DrainFast:Disabled PortSoftStart:2 - 1800
   DTS
```

Hardmax

Softmax

	PortSMin		GlblSMin		PortStEnd					
0	1	6	200	7	800	19	475	0	0	3	2400
1	1	5	0	8	1200	19	712	8	300	3	2400
2	1	5	0	6	0	0	0	0	0	3	2400
3	1	5	0	6	0	0	0	0	0	3	2400
4	1	5	0	6	0	0	0	0	0	3	2400
5	1	5	0	6	0	0	0	0	0	3	2400
6	1	5	0	6	0	0	0	0	0	3	2400
7	1	5	0	6	0	0	0	0	0	3	2400

Image 2. Queue Buffer Allocation

You specifically want to look at the Hardmax and Softmax column which shows the number of buffers the queues have available. For information on what these buffers are and how they are allocated by default, see the Catalyst 9000 QoS and Queueing White Paper.

The second command is **show platform hardware fed switch active qos queue stats interface <interface>**. This command allows you to see per-queue statistics on an interface, which includes how many bytes were enqueued into the buffers, and how many bytes were dropped due to lack of available buffers.

<#root>

```
9300#
```

show platform hardware fed switch active qos queue stats interface Gig 1/0/1

```
DATA Port:0 Enqueue Counters
-------------------------------------------------------------------------------------------
Q Buffers          Enqueue-TH0          Enqueue-TH1          Enqueue-TH2             Qpolicer
  (Count)              (Bytes)              (Bytes)              (Bytes)              (Bytes)
- ------- -------------------- -------------------- -------------------- --------------------
0       0                    0                    0
```

```
384251797

                    0
1       0                   0                   0

488393930284

                    0
2       0                   0                   0                   0                   0
3       0                   0                   0                   0                   0
4       0                   0                   0                   0                   0
5       0                   0                   0                   0                   0
6       0                   0                   0                   0                   0
7       0                   0                   0                   0                   0

DATA Port:0 Drop Counters
------------------------------------------------------------------------------------------------------
Q             Drop-TH0             Drop-TH1             Drop-TH2             SBufDrop             QebD
               (Bytes)              (Bytes)              (Bytes)              (Bytes)              (Byt
- -------------------- -------------------- -------------------- -------------------- ------------------
0                     0                    0                    0                    0
1                     0                    0

192308101

                    0                   0                   0
2                     0                    0                    0                    0
3                     0                    0                    0                    0
4                     0                    0                    0                    0
5                     0                    0                    0                    0
6                     0                    0                    0                    0
7                     0                    0                    0                    0
```

Image 3. Queue Buffer Statistics with Drops

As shown in Image 3, Queue 0 and Queue 1 both have bytes enqueued, but it is Queue 1 that experiences drops in the Drop-TH2 column. This information indicates that Queue 0 traffic has not been impacted by this congestion, and that the cause of the congestion is specifically Queue 1 traffic.

Modify Buffers to Resolve Output Drops

SoftMax Multiplier

To increase the number of buffers each queue can request from the shared pool, increase the SoftMax threshold with the configuration qos queue-softmax-multiplier <100 – 1200>. The highest value is 1200, and increases by a multiple of 12, the ability of a single port queue to absorb micro-bursts. This command increases the port queue thresholds so that the port queue can consume additional buffer units from the shared pool. As shown in Image 4, configuration and the increased buffer allocation.

```
<#root>

9300(config)#

qos queue-softmax-multiplier 1200



9300#

show platform hardware fed switch active qos queue config interface


gigabitEthernet


1/0/48


Asic:0 Core:0 DATA Port:47 GPN:48 LinkSpeed:0x1
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 376 - 383
  DrainFast:Disabled PortSoftStart:3 - 14400
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd
  ----- --------  --------  --------  --------  ---------
 0   1  6   200   9  9600   2   600   0     0   1 15000
 1   1  5     0  10 14400   2   900   1   450   1 15000
 2   1  5     0   6     0   0     0   0     0   1 15000
 3   1  5     0   6     0   0     0   0     0   1 15000
 4   1  5     0   6     0   0     0   0     0   1 15000
 5   1  5     0   6     0   0     0   0     0   1 15000
 6   1  5     0   6     0   0     0   0     0   1 15000
 7   1  5     0   6     0   0     0   0     0   1 15000
```

Image 4. Queue Config with SoftMax Multiplier of 1200

This a common configuration used as a quick method to resolve output drops. In Image 4, this configuration applies to all non-priority queues across all interfaces. The buffer allocation itself assumes that the micro-bursts does not happen, on all ports, on the switch at the same time. If micro-bursts happen in random moments, the shared buffer can dedicate additional buffer units to absorb them.

Per-Queue Buffer Modification

Per-Queue buffer modification can be leveraged for scenarios where you cannot use the SoftMax multiplier, or in scenarios where you attempt to fine tune the buffers to fit a traffic profile. To modify the queue buffer allocation, the switch on a per interface basis, you must use policy-maps. In most circumstances, you modify the current policy-map of an interface and change the buffers on a per class basis.

In this example, interface GigabitEthernet1/0/48 has experienced output drops. As shown in Image 5, egress policy-map that is applied to this interface.

```
policy-map MYPOL
 class Voice
  priority level 1 percent 20
 class Video
  priority level 2 percent 10
 class Control
  bandwidth percent 10
 class Data
  bandwidth percent 5
 class class-default
```

Image 5. Example Policy Map

This policy-map has 5 class-maps, which results in 5 total egress queues on the interface. Each class has a default number of buffers allocated to it based on its priority level.

Image 6 displays the current buffer allocations.

<#root>

9300#

show platform hardware fed switch active qos queue config interface gigabitEthernet 1/0/48

```
Asic:0 Core:0 DATA Port:47 GPN:48 LinkSpeed:0x1
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 376 - 383
  DrainFast:Disabled PortSoftStart:3 - 600
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd
  ----- --------  --------  --------  --------  ---------
 0   1  7   100   9   100   0     0   0     0   3   800
 1   1  7   100  10   400  19   237   0     0   3   800
 2   1  5     0  10   400  19   237   8   100   3   800
 3   1  5     0  10   400  19   237   8   100   3   800
 4   1  5     0  10   400  19   237   8   100   3   800
 5   1  5     0   6     0   0     0   0     0   3   800
 6   1  5     0   6     0   0     0   0     0   3   800
 7   1  5     0   6     0   0     0   0     0   3   800
```

Image 6. Queue Buffer Config with the Example Policy

Since this interface has experienced output drops, look at the queueing statistics of the interface to see where the congestion is.

<#root>

9300#

show platform hardware fed switch active qos queue stats interface gigabitEthernet 1/0/48

```
DATA Port:0 Enqueue Counters
-------------------------------------------------------------------------------------------
Q Buffers          Enqueue-TH0          Enqueue-TH1          Enqueue-TH2             Qpolicer
  (Count)              (Bytes)              (Bytes)              (Bytes)              (Bytes)
- ------- -------------------- -------------------- -------------------- --------------------
0       0                    0                    0               489094                    0
1       0                    0                    0              4846845                    0
2       0                    0                    0             89498498                    0
3       0                    0                    0

21297827045


                   0
4       0                    0                    0             74983184                    0
5       0                    0                    0                    0                    0
6       0                    0                    0                    0                    0
7       0                    0                    0                    0                    0

DATA Port:0 Drop Counters
-------------------------------------------------------------------------------------------
Q             Drop-TH0             Drop-TH1             Drop-TH2             SBufDrop             QebD
                (Bytes)              (Bytes)              (Bytes)              (Bytes)             (Byt
- -------------------- -------------------- -------------------- -------------------- ----------
0                    0                    0                    0                    0
1                    0                    0                    0                    0
2                    0                    0                    0                    0
3                    0                    0

3854484


                   0                    0                    0
4                    0                    0                    0                    0
5                    0                    0                    0                    0
6                    0                    0                    0                    0
7                    0                    0                    0                    0
```

Image 7. Queue Buffer Statistics with Drops with an Example Policy

Image 7 shows that Queue 3 has more traffic enqueued than any other queue, and it is also the only that has experienced output drops. Since the queue number starts at 0, Queue 3 maps to the fourth class-map, class Data.

To alleviate the drops on this queue, allocate more buffers to Queue 3. To change this buffer allocation, use the queue-buffers ratio <0-100> configuration in the policy-map. If configured on each class in the policy, it must add up to 100. If you only configure a single class with this command, the system attempts to evenly subtract buffers from the other queues.

In Image 8, the Data class has been configured with queue-buffers ratio 40.

<#root>

```
policy-map MYPOL
```

```
class Voice
  priority level 1 percent 20
 class Video
  priority level 2 percent 10
 class Control
  bandwidth percent 10
 class Data
  bandwidth percent 5


queue-buffers ratio 40
```

Image 8. Example Policy Map with Modified Queue Buffers

In Image 9, you can see the Data class now has 40% of the interface buffers, 800 buffers total.

<#root>

9300#

show platform hardware fed switch active qos queue config interface gigabitEthernet 1/0/48

```
Asic:0 Core:0 DATA Port:47 GPN:48 LinkSpeed:0x1
AFD:Disabled FlatAFD:Disabled QoSMap:0 HW Queues: 376 - 383
  DrainFast:Disabled PortSoftStart:3 - 1200
   DTS  Hardmax  Softmax   PortSMin  GlblSMin  PortStEnd
  ----- --------  --------  --------  --------  ---------
 0   1  7    75   9    75   0     0   0     0   3  1600
 1   1  7    75  10   300  19   178   0     0   3  1600
 2   1  5     0  10   300  19   178   8    75   3  1600
 3   1  5     0   7
```

800

```
  19   475   8   200   3  1600
 4   1  5     0  10   300  19   178   8    75   3  1600
 5   1  5     0   6     0   0     0   0     0   3  1600
 6   1  5     0   6     0   0     0   0     0   3  1600
 7   1  5     0   6     0   0     0   0     0   3  1600
```

Image 9. Queue Buffer Config with the Updated Example Policy

This also causes the other queues to have less Softmax buffers. It is important to make these buffer changes in small increments to ensure the changes do not result in output drops on the other queues.

With that change made, check the queue stats and see if drops still increment on this or any other queue. If the drops continue, modify the queue-buffer configuration further until the output drops are resolved.

Alternative Methods to Manage Congestion

QoS is primarily a method to prioritize traffic, and it is not a solution for every output drop scenario. There are some scenarios where a modification of the queue buffers is not enough to resolve all the output drops. In those scenarios, you can manage congestion in several other ways:

- Reduce the oversubscription ratio.

This includes methods that increase your egress bandwidth such as port-channels or Equal Cost Multipath (ECMP), but can also require more involved configurations such as traffic engineering.

- Use a queueing scheduler to prioritize traffic.

While a queue scheduler does not stop congestion, it does protect your important traffic from impact by the congestion

- Use congestion management algorithms such as Weighted Random Early Discard (WRED) or Weighted Tail Drop (WTD) to drop some of the traffic earlier.

- Police the traffic on ingress to reduce the traffic on egress.

Analyze Output Drops with Wireshark

Wireshark is a useful tool to identify bursts of traffic that cause buffer congestion and drops. If you SPAN an interface in the egress direction while it experiences drops, Wireshark can graph the output rate to see when and what traffic triggered the drops. This is especially useful when identifying output drops in low throughput scenarios.

View the I/O Rate

Once you open your SPAN capture with Wireshark, select **Statistics**, then **I/O Graph**, as demonstrated in Image 10.

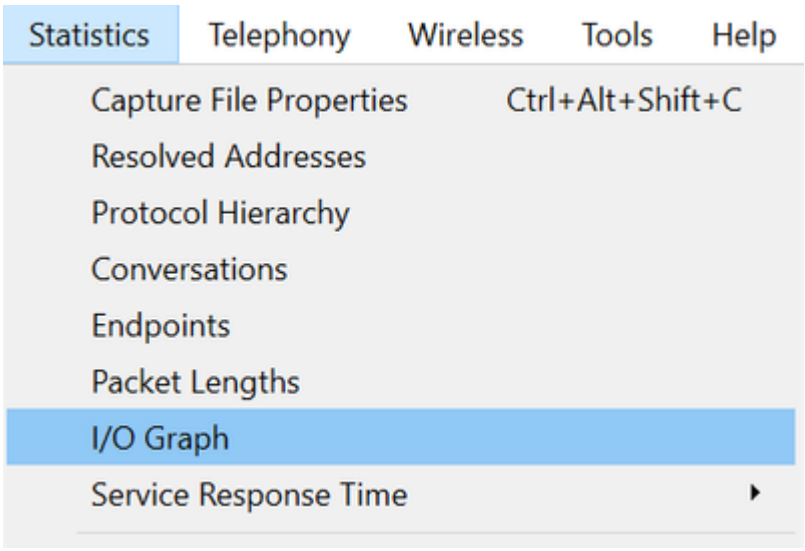


Image 10. Select the I/O Graph

Once that is selected, Wireshark generates a graph of the traffic in bits per second. Image 11 shows an example graph for an interface while it experienced output drops.

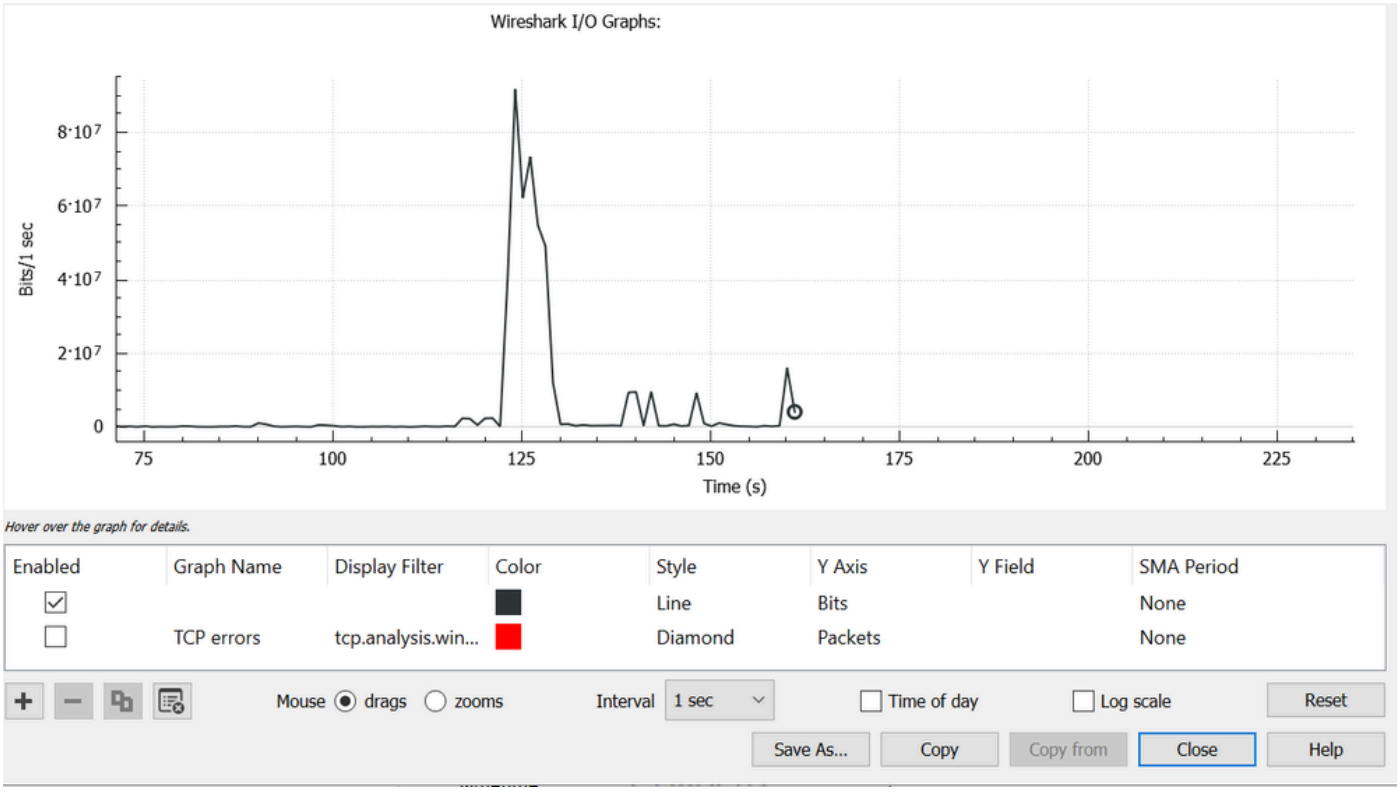


Image 11. I/O Graph Bits/millisecond

The Image 11 graph indicates that the interface had a maximum throughput that barely exceeds 80Mbps. The default graph view is not granular enough to identify small bursts of traffic that cause packet drops. It is an average of the traffic rate on a per-second basis. To understand how this rate could cause buffer congestion, consider the throughput on a millisecond scale.

A Gigabit interface can forward 1,000,000,000 bits per second. Once converted to milliseconds, this equals 1,000,000 (or 10^6) bits per millisecond.

When the interface rate goes beyond that forwarding speed of the interface, the switches must buffer these packets, which results in congestion and output drops.

View the I/O Rate in Milliseconds

Wireshark allows the user to graph the I/O Rate as bits per millisecond. To do this, reduce the Interval from 1 sec to 1 ms, then click **Reset** to properly view the graph. This step is shown in Image 12.

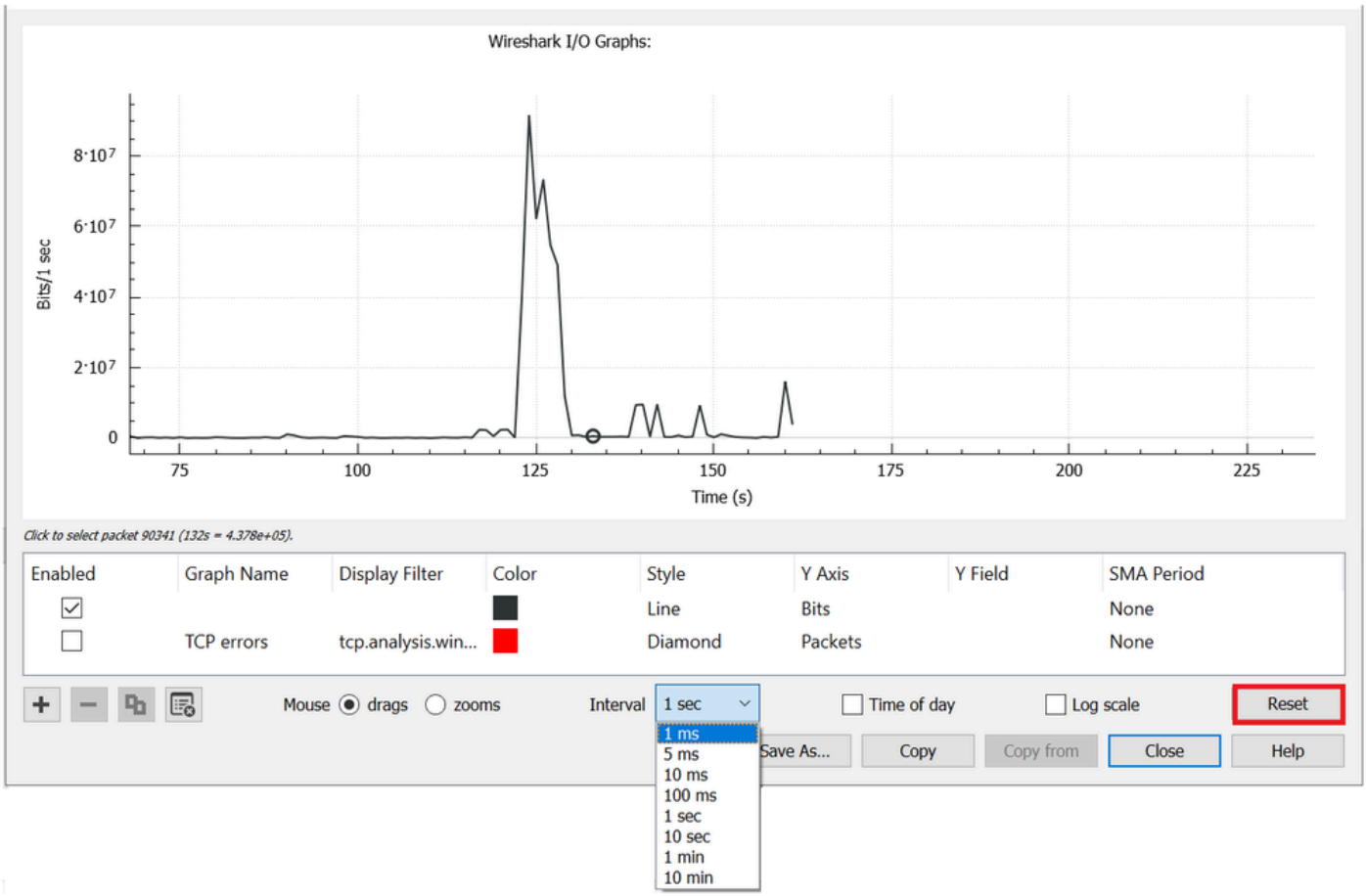


Image 12. Reduce the Interval to 1ms and Reset the Graph

The updated graph more accurately displays the true I/O rate of the interface. When the rate meets or exceeds 10^6 bits per millisecond, the switch experiences congestion or output drops. Image 13 shows the updated I/O graph for an interface that experienced output drops.

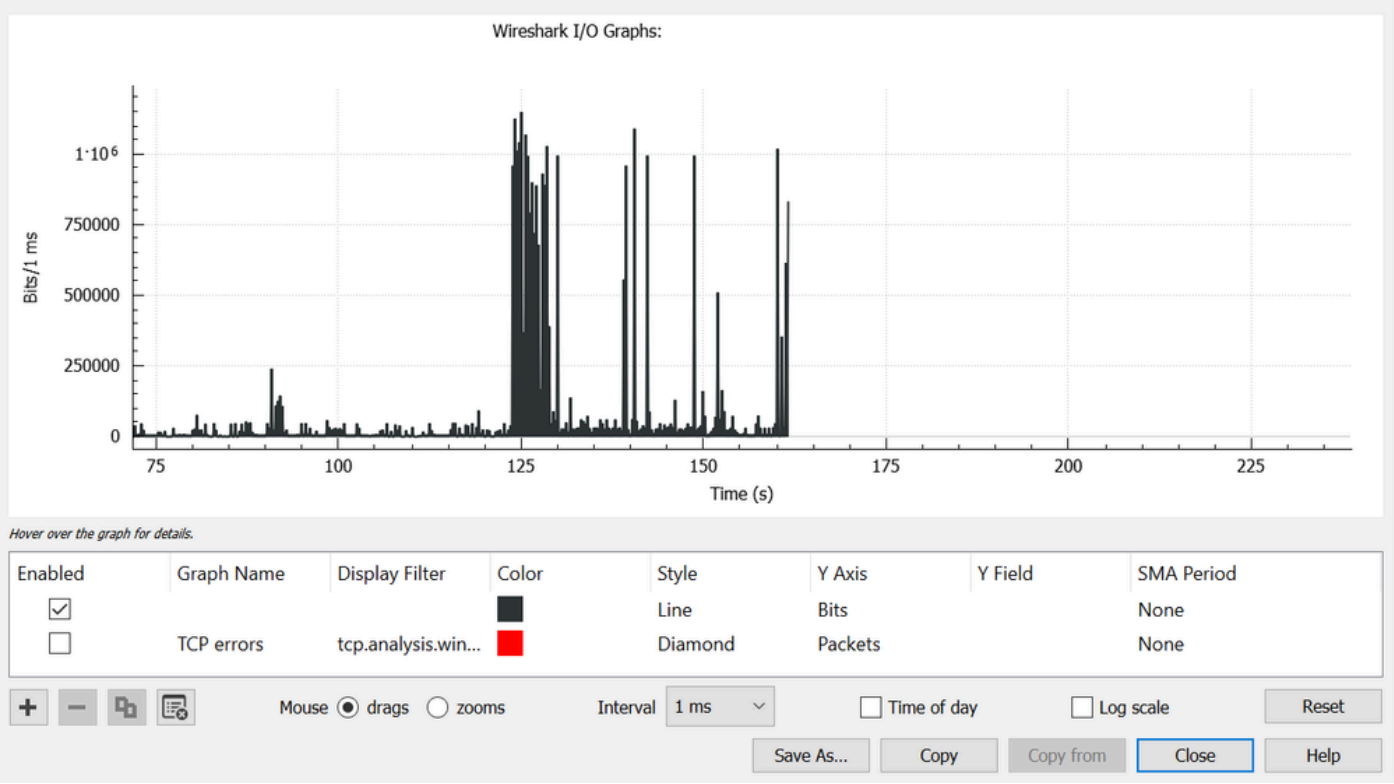


Image 13. I/O Graph Bits/millisecond

Image 13 shows that there are multiple traffic peaks that meet or exceed the 10^6 threshold. Traffic would be subject to buffering and be dropped if it exceeds our egress buffer size.

Note: If the SPAN destination is connected by a 1 Gbps interface, the I/O rate in Wireshark cannot exceed that 10^6 bits per millisecond rate, no matter what the source interface rate is. The SPAN destination interface instead buffers or drops those packets. It is common to see the I/O graph plateau at that maximum throughput or present an average traffic rate that appears to go higher.
