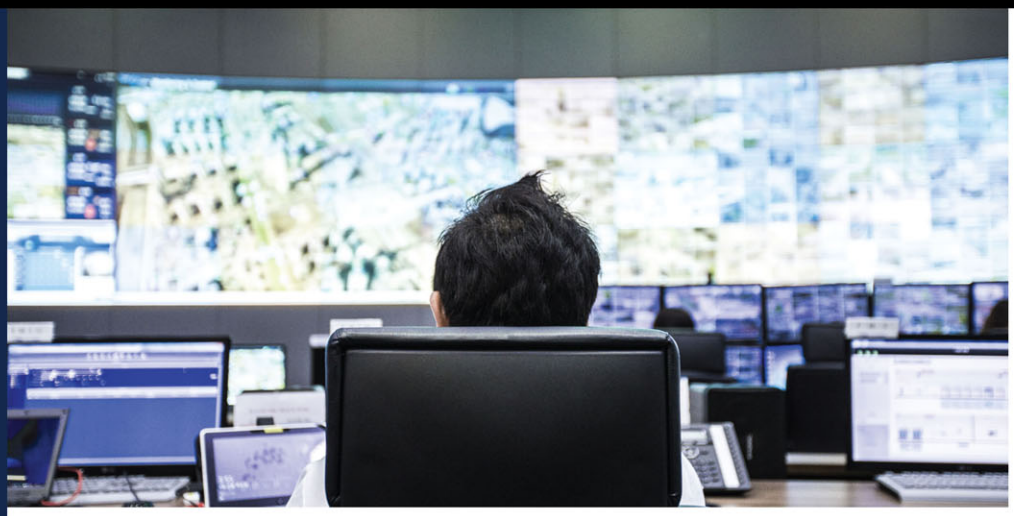

Detecting, Troubleshooting, and Preventing Congestion in Storage Networks

PARESH GUPTA, CCIE® NO. 36645
EDWARD MAZUREK, CCIE® NO. 6448

ciscopress.com

Detecting, Troubleshooting, and Preventing Congestion in Storage Networks

Paresh Gupta
Edward Mazurek

Cisco Press
Hoboken, New Jersey

The topology is interactive. You can hover over a network link or a switch in the topology to see more details about it. By double-clicking on an entity, you can limit the scope of the table to that entity.

You can click on the trend line on the left of the interface to see the metric charts. As Figure 3-11 shows, you get a pop-up chart of congestion metrics for that interface. You can hover over the chart to see the value at that time. You can click on the legend below the graph to hide or show the chart for that metric.

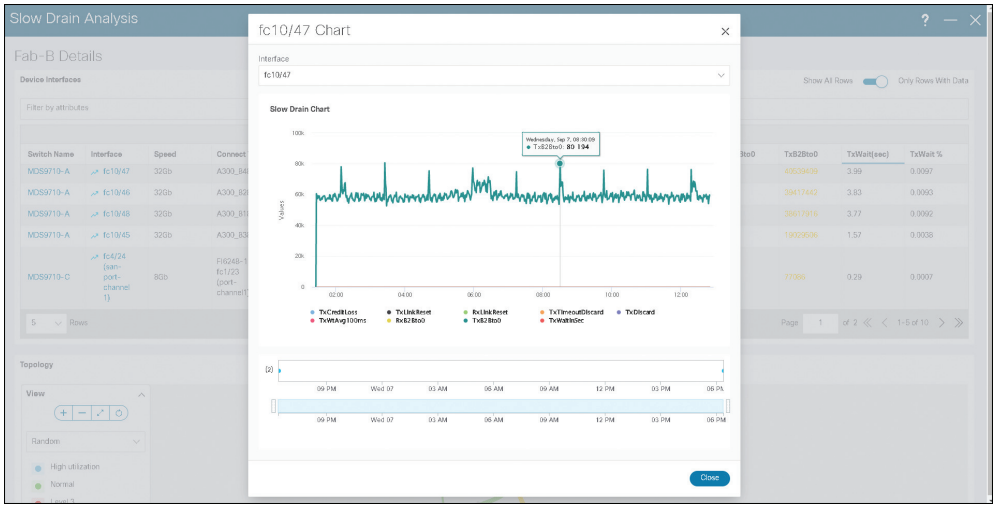

Figure 3-11 *Trends and Seasonality of Congestion in NDFC Congestion Analysis*

Note Congestion analysis in NDFC is continuously evolving. New features are being added to it in every release. It is possible that by the time you read this chapter, the latest release of NDFC will look different from the screenshots in this chapter. Please refer to the Cisco NDFC SAN release notes for up-to-date information.

The MDS Traffic Monitoring (MTM) App

The MDS Traffic Monitoring (MTM) app is a simple application for monitoring a large number of MDS switches. It was developed by Paresh Gupta, coauthor of this book, and it is available on GitHub (https://github.com/paregupt/mds_traffic_monitor) for free under the MIT license.

MTM Architecture

MTM polls the MDS switches every 30 seconds (by default) for metrics such as inventory, health metrics like CPU and memory, port states, port speeds, port utilization, error counters, and transceiver health metrics such as temperature and power.

MTM has a collector (written in Python) that pulls metrics from Cisco MDS switches using NX-API. The metrics are normalized and correlated before being written to InfluxDB (a time-series database). Finally, Grafana provides the visualization and use cases.

The MTM collector needs a read-only account on the MDS switches, and NX-API must be enabled on them using the NX-OS command **feature nxapi**.

By default, all the metrics are stored in the database forever.

MTM Use Cases

MTM is a lightweight app. It is not designed to take the place of fully functional monitoring platforms such as Cisco NDFC. Instead, it is designed for users who like customizing monitoring platforms based on the problems they are trying to solve. Many ready-made use cases are already available in MTM. Users can customize it further or integrate it with other similar apps, like UCS Traffic Monitoring (UTM), which monitors traffic in the Cisco UCS servers. See Chapter 9, "Congestion Management in Cisco UCS Servers," for more details on the UTM app.

MTM groups switches logically. For example, you can monitor all the switches at a site or all the switches in a fabric on a single dashboard. Then, you can drill down at the switch level or investigate a switchport.

At the time of this writing, MTM is used for monitoring many production environments. Chapter 2 uses MTM, and Chapter 6 demonstrates DIRL using MTM. The illustrations later in this chapter, in the section "Average and Peak Utilization," are taken from MTM running in another production environment.

The following sections explain a handful of MTM use cases.

Finding Slow-Drain Devices

As Figure 3-12 shows, MTM shows the top 10 ports with the highest TxWait values across all the monitored MDS switches. If these ports are edge-type ports, their connected devices are the source of slow drain. Typically, you should start investigating the port at the top of this list.

Finding Highly Utilized Links

As Figure 3-13 shows, MTM shows the top 10 most utilized ports across all the monitored MDS switches. As previously explained, ports that show high utilization are likely to be the cause of congestion due to overutilization. In Figure 3-13, the port at the top of the list is running at 98.3% capacity, and it might very well be causing congestion.

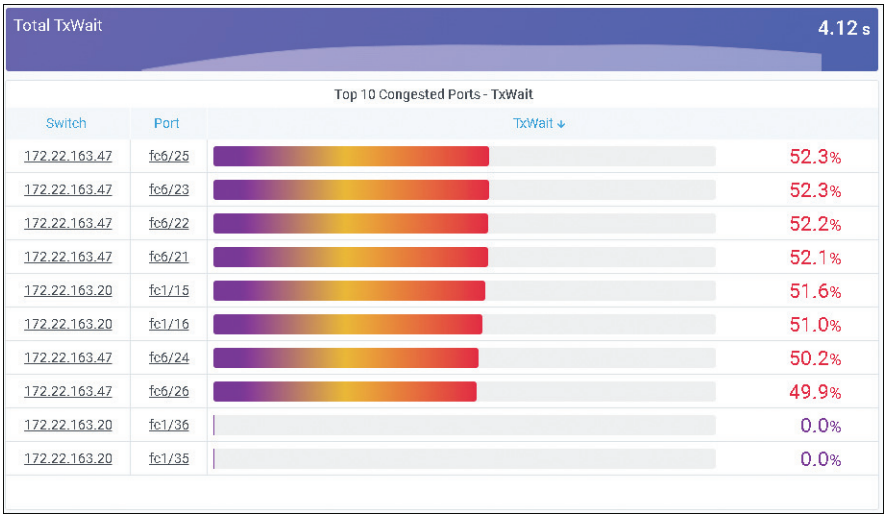

Figure 3-12 *Top 10 Congested Ports Due to TxWait in the MTM App*

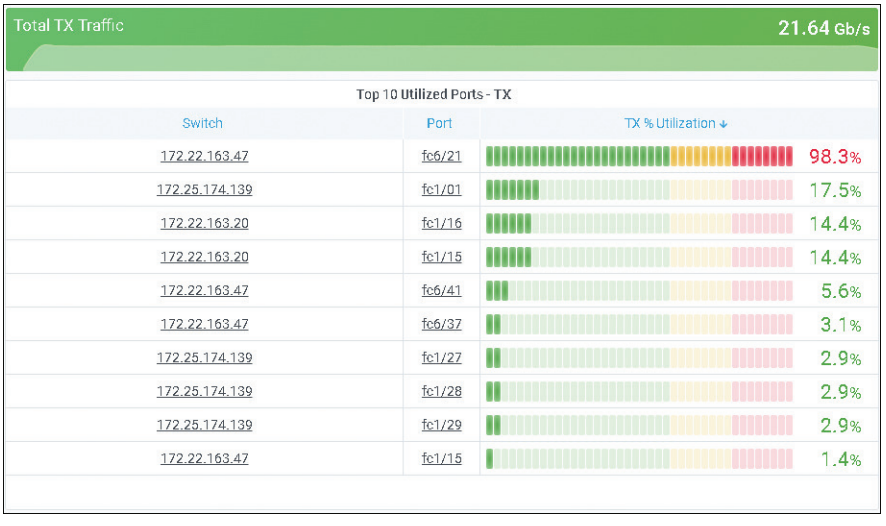

Figure 3-13 *Top 10 Utilized Ports in the MTM App*

Finding Ports with the Most Errors

MTM shows the top 10 ports with the most errors across all the monitored MDS switches. Figure 3-14 shows the top 10 ports with FEC-corrected blocks. As Chapter 2 explains, FEC-corrected blocks indicate bit errors, which may lead to congestion or worsen existing congestion.

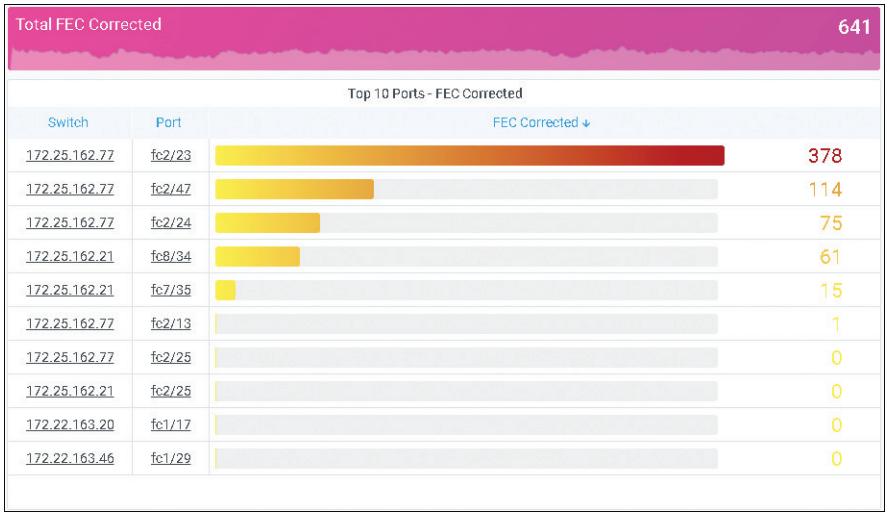

Figure 3-14 *Top 10 Ports with the Highest Numbers of FEC Corrected Blocks*

MTM monitors the following metrics for MDS switches and shows the top 10 ports:

- CRC-corrupted frames
- FEC-corrected blocks
- FEC-uncorrected blocks
- Tx and Rx B2B credit transitions to zero
- Timeout discards
- Invalid transmission words (ITWs)
- Link failures
- Signal loss
- Sync loss
- Link Reset (LR) received and transmitted
- Link Reset Response (LRR) received and transmitted

Trending and Seasonality

MTM stores all the metrics forever to provide long-term trending and seasonality. By using the port utilization trends, you can predict when a port may reach more than 90% utilization and plan an upgrade accordingly. As Figure 3-15 shows, you can correlate multiple metrics on a switchport side by side.

Figure 3-15 *Side-by-Side Trends and Seasonality of Multiple Metrics*

End-to-End Correlation

For troubleshooting congestion end to end in a fabric, you can monitor the ports on multiple switches in the data path. For example, Chapter 6 uses the MTM app for monitoring an edge switchport side by side with an ISL port on an upstream switch. In the same demonstrations, metrics from the application layer on the connected hosts are also shown side by side, although this feature is not available in the MTM app by default.

Metric Export Mechanisms

This section explains various mechanisms and recommendations for exporting metrics from network devices to remote monitoring platforms.

Parsing the Command-Line Output over SSH

Remote monitoring platforms, applications, or scripts can open a SSH session to a device, run the commands, collect their outputs, and parse the outputs in order to capture relevant metrics. Although this approach is widely used in production environments, its users often complain that the parsing logic breaks when the command output changes, and trying to catch up with the ongoing changes takes significant effort. Another drawback is the longer polling interval because opening SSH sessions and collecting the command output involves delay.

Simple Network Management Protocol (SNMP)

SNMP is the most common mechanism for exporting metrics from network devices. Typically, the remote monitoring platforms poll the SNMP OID at every polling interval. Some OIDs, such as ifHCInOctets and ifHCOutOctets, are standardized and thus work on most devices. For polling device-specific metrics, refer to the documentation to find their OIDs.

Table 3-4 shows commonly used SNMP MIBs and their OIDs for polling metrics from the ports on Cisco MDS switches.

Table 3-4 *Commonly Used MIBs for Monitoring the Ports on Cisco MDS Switches*

Serial No.	MIB Name	OID	Description
1	ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6	Number of bytes received by the interface. This can be used to calculate Rx datarate in bits per second and percentage.
2	ifHCOutOctets	1.3.6.1.2.1.31.1.1.1.10	Number of bytes transmitted by the interface. This can be used to calculate Tx datarate in bits per second and percentage.
3	fcIfTxWaitCount	1.3.6.1.4.1.9.9.289.1.2.1.1.15	TxWait, the number of 2.5 µs when a switch port could not transmit due to zero remaining-Tx-B2B-credits.
4	fcHCIfBBCreditTransistionFromZero	1.3.6.1.4.1.9.9.289.1.2.1.1.40	Tx B2B credit transitions to zero.
5	fcIfBBCreditTransistionToZero	1.3.6.1.4.1.9.9.289.1.2.1.1.39	Rx B2B credit transitions to zero.