Technical White Paper–2023

Using NetFlow for Monitoring and Troubleshooting with VMware Aria Operations for Networks

vmware®

Table of contents

Overview

Customers need visibility into the applications that run over their networks and an understanding of how network traffic flows in their environment. The VMware Aria Operations™ for Networks solution provides that information.

Data center management that once mainly revolved around compute, storage, network and security has become more application-centric, with compute, storage and the like becoming enablers for DevOps. Accordingly, application site reliability engineers (SREs)/DevOps engineers need to monitor, manage and troubleshoot the applications running on an underlying infrastructure that might not be visible or accessible to them.

Application performance monitoring products can provide visibility into an application's performance metrics, but they can't help monitor or troubleshoot any issues that might arise out of the underlying infrastructure. So it's imperative to get visibility into the performance of the application and its tiers, its communication with its peers, and latency and other metrics between them.

Getting the details of the underlying infrastructure and network becomes crucial to ensure the application always runs without issues. For example, if load increases on an application, the app owner/network engineer needs to be alerted if any networking component reaches its threshold limit.

This information can be used to successfully keep the lights on as well as for optimizations or cost reductions. For example, based on the observations, the placement of workloads can be optimized to ensure the top talking application/tiers are placed with minimum hops between them.

Flows and associated metrics can be a treasure trove of information for those who know how to use it. This white paper reviews how VMware Aria Operations for Networks can be leveraged for some use cases and scenarios.

Typical customer example

Let's consider an application with three different services (database, analytics and SQL) connected to three different L2s (Segment-DB, Segment-Analytics, and Segment-SQL). Whenever a request comes from the application, these services communicate with each other and provide the desired result.

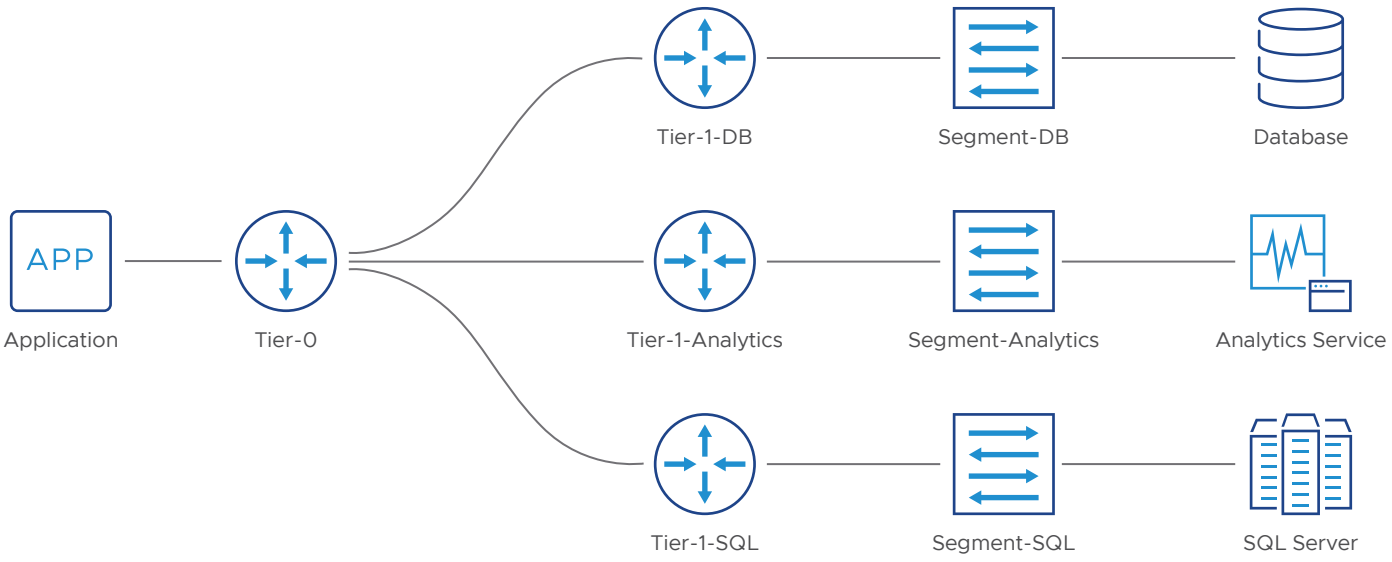

Figure 1: Typical customer topology diagram of where NetFlow is collected.

In a typical flow:

• The user gives a query to the application based on certain parameters to fetch data.

• The parameters are sent to SQL, which queries the database for the results.

• The results are then sent to the analytics service to analyze the data and present it to the user.

For this example, the application user has reported issues with retrieving the analyzed data and that the application sometimes gives the results slowly.

Now, the application owner needs to analyze why the application's performance is impacted and find the services responsible for it. The app owner will need information on the services talking to each other frequently as well as the total traffic, the traffic rate, latency, packet loss, and the like. And the network admin will need to check for any bandwidth issues on the routers when the services communicate with each other.

VMware Aria Operations for Networks leverages IPFIX/NetFlow to capture and display all network traffic in the data center. The flow/traffic is attached with details of the source, destination, port and protocol. The source/destination can be workloads in the form of VMs.

The information obtained via the IPFIX/NetFlow records is minimal. VMware Aria Operations for Networks enriches the flow with information about the data center, host, L2s, and the like. Security information, such as the security groups and firewall, also gets attached to the flow.

NSX network info

Port	L2 network	Router	...
db-port1	Segment-DB	Tier-1-DB	
analytics-port1	Segment-Analytics	Tier-1-Analytics	
sql-port1	Segment-SQL	Tier-1-SQL	

ESX host info

Host ID	Host name	Host IP	...
Host1	db-server	192.168.10.0/24, 2001:fd02::12/64	
Host2	analytics-server	192.168.11.0/24, 2001:fd02::22/64	
Host3	sql-server	192.168.12.0/24, 2001:fd02::2/64	

VM info

VM ID	VM name	VM IP	VM host	VM L2 network	VM default gateway
VM1	db-VM1	192.168.10.21/32, 2001:fd02::12/64	Host1	Segment-DB	Tier-1-DB
VM2	analytics-VM1	192.168.11.16/32, 2001:fd02::22/64	Host2	Segment-Analytics	Tier-1-Analytics
VM3	sql-VM1	192.168.12.23/32, 2001:fd02::2/64	Host3	Segment-SQL	Tier-1-SQL

NSX security policy info

Policy ID	Policy name	...
SP1	sqlVMs-dbVMs	
SP2	sqlVMs-analyticsVMs	
SP3	dbVMs-sqlVMs	

Flow info

Flow ID	Source IP	Source VM	Dest. IP	Dest. VM	Source port	Dest. port	Policy ID	Size	Source L2	Dest. L2	...
flow1	192.168.12.32	sql-VM1	192.168.10.21	db-VM1	43654	443	SP1	2048	Segment-SQL	Segment-DB	
flow2	192.168.12.32	sql-VM1	192.168.11.16	analytics-VM1	43214	443	SP2	1286	Segment-SQL	Segment-Analytics	

Figure 2: Flow enrichment in VMware Aria Operations for Networks.

In addition to the flow information, various metrics provide details about packet drops, memory usage, CPU usage, and the like for various networking entities, such as routers and interfaces, which will help the network admin. So the NetFlow information in VMware Aria Operations for Networks gives a holistic view of the application's networking and security constructs.

Let's see how data from VMware Aria Operations for Networks can be leveraged to solve the application's issues.

Most of the traffic would be to/from the database. To find the network traffic originating from the database, either of the following queries can be used:

```
flow where source L2 Network = 'Segment-DB' and Destination L2 Network != 'Segment-DB' group by Destination L2 Network order by sum(Total traffic)
```

```
flow where source L2 Network = 'Segment-DB' and flow type = 'Routed' group by Destination L2 Network order by sum(Total traffic)
```

A similar query can used to determine the traffic to the database:

```
flow where Destination L2 Network = 'Segment-DB' and Source L2 Network != 'Segment-DB' group by Source L2 Network order by sum(Total traffic)
```

```
flow where Destination L2 Network = 'Segment-DB' and flow type = 'Routed' group by Source L2 Network order by sum(Total traffic)
```

Use the following query to get traffic happening within an L2 network:

```
flow where L2 Network = 'Segment-DB' and flow type = 'Switched' order by sum(Total traffic)
```

```
flow where source L2 Network = 'Segment-DB' and Destination L2 Network = 'Segment-DB' order by sum(Total traffic)
```

To get all of this information in a single view, use the following query:

```
flow where L2 Network = 'Segment-DB' group by Source L2 Network,Destination L2 Network order by sum(Total traffic)
```

Table 1 shows the traffic patterns observed from running these queries and analyzing the data. It can be seen that the traffic between Segment-DB and Segment-SQL is very high. This traffic does increase during peak times.

Table 1: Total traffic between pairs of L2 networks

Source L2 network	Destination L2 network	Total traffic
Segment-DB	Segment-Analytics	514GB
Segment-DB	**Segment-SQL**	**10.2TB**
Segment-DB	Segment-DB	112GB
Segment-SQL	**Segment-DB**	**2.8TB**
Segment-Analytics	Segment-DB	231GB

So now that the application owner is aware of the app's traffic patterns, and with the help of a network engineer, the data can be used to determine if there are any packet drops on interfaces of tier routers (Tier-1-DB and Tier-1-SQL).

The corresponding edge nodes (CPU usage, memory usage, etc.) also need to be checked for limit breaches. Figure 3 shows that the memory/CPU usage of the edge nodes is reaching 100 percent. As a result, the application is facing packet drops and the performance is impacted.

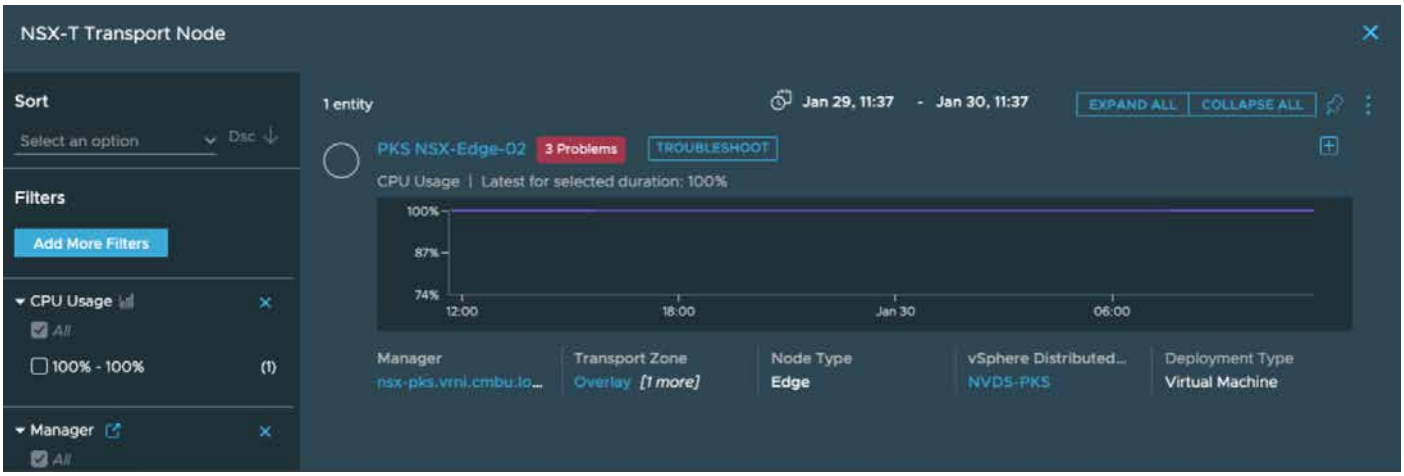

Figure 3: Example of 100 percent CPU usage on edge nodes.

With insight into the topology, the workloads under Segment-DB and Segment-SQL were identified as contributing to the maximum traffic. The application owner can optimize the topology by placing them under the same tier-1 router under different L2s. This ensures that the edge node will not be overloaded due to the traffic between these services. The modified application will look as seen in Figure 4.

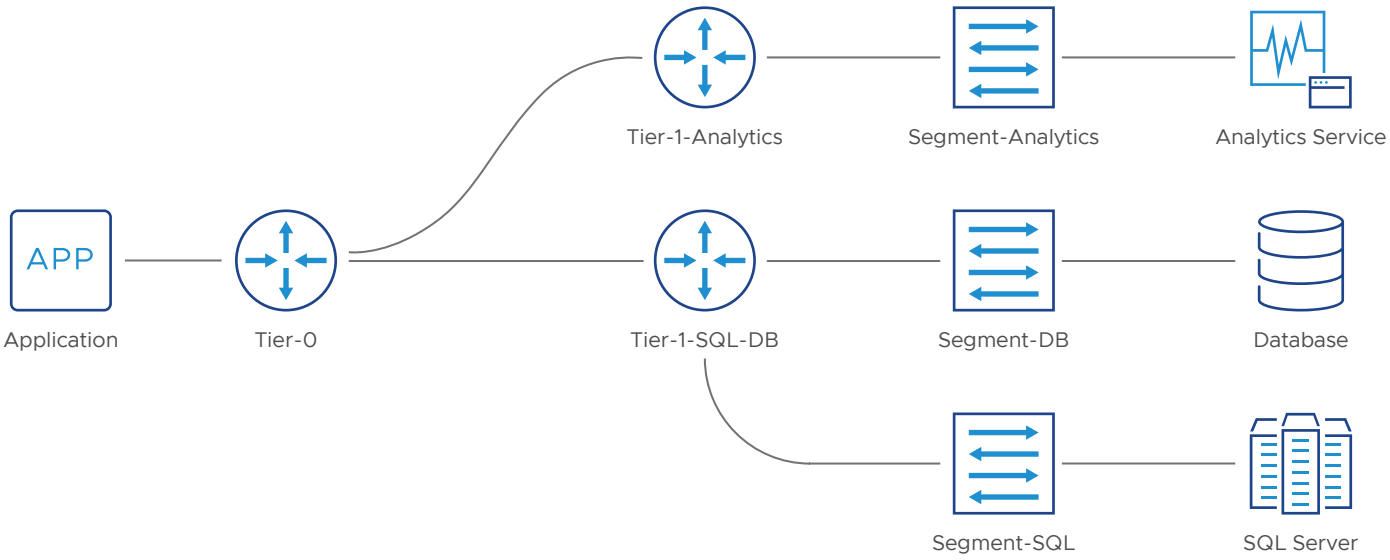

Figure 4: Modified customer topology diagram.

The following queries can also be useful.

If interested in determining the flow across all the L2 network, use the following query:

```
flow group by Source L2 Network,Destination L2 Network order by sum(Total traffic)
```

Other metrics—such as TCP RTT, number of sessions, TCP retransmission count, and the like—can also be determined by changing the order by clause.

Use the following query if interested in knowing the traffic leaving the Tier-1-SQL-DB router:

```
flow where source L2 Network in (Segment-DB, Segment-SQL) and Destination L2 Network not
in (Segment-DB, Segment-SQL) order by total traffic
```

The following query can also be used to identify all traffic leaving a tier-1 router:

```
flow where source vm in (vm where Default Gateway Router = 'Tier-1-SQL-DB') and destination
vm not in (vm where Default Gateway Router = 'Tier-1-SQL-DB')
```

A similar query can be used to determine the incoming traffic to the tier-1 router. The way to identify the traffic from a tier-0 router is explained later in this white paper.

The insight into the data from VMware Aria Operations for Networks helped optimize the topology and identify the root cause for the application's slowness.

Getting the flow/traffic information between the application/tiers with its networking constructs can help solve various problems, as shown in this section.

Customers deploy their applications using VMware vCenter® for workloads and VMware NSX® for networking. VMware Aria Operations for Networks processes the DFW IPFIX flow records from NSX and enriches them with information about the application and its networking. The NetFlow information provides valuable insights into the system. Security planning and micro-segmentation are some of the obvious use cases of the NetFlow traffic information. This white paper covers the not-so-obvious scenarios that center around four major Day 2 operations themes: insights, monitoring, alerts and troubleshooting.

Flow insights using L2/application

This section shows what different insights can be gathered from flow information in VMware Aria Operations for Networks.

First, determine the top applications by traffic in VMware Aria Operations for Networks:

```
flow group by application order by sum(Total traffic)
```

If the traffic flows from a different L2 (i.e., VLAN and NSX L2) and you want to determine the top talking applications over the NSX L2s in the system, use the following query:

```
flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network
where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net
in (NSX-T L2 Network where name is set)) group by application order by sum(Total traffic)
```

This shows the top talking applications with both the source L2 and destination L2 belonging to NSX. If the NSX L2 has packet drops, then the application on the L2s will be impacted.

Figure 5: Example of the top talking applications on NSX L2 networks.

In addition to the top talking applications, you can also get the top talking application pairs:

```
flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network
where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net
in (NSX-T L2 Network where name is set)) group by source application, destination application
order by sum(Total traffic)
```

If there are only NSX L2s, then the query can be simplified:

```
flow group by source application, destination application order by sum(Total traffic)
```

This result will help you understand which are the top talking application pairs over the NSX L2. The top pairs are important as most of the traffic happens between these applications.

You need to ensure the workloads in these applications have the load distributed equally and that the underlying networking components aren't reaching threshold limits.

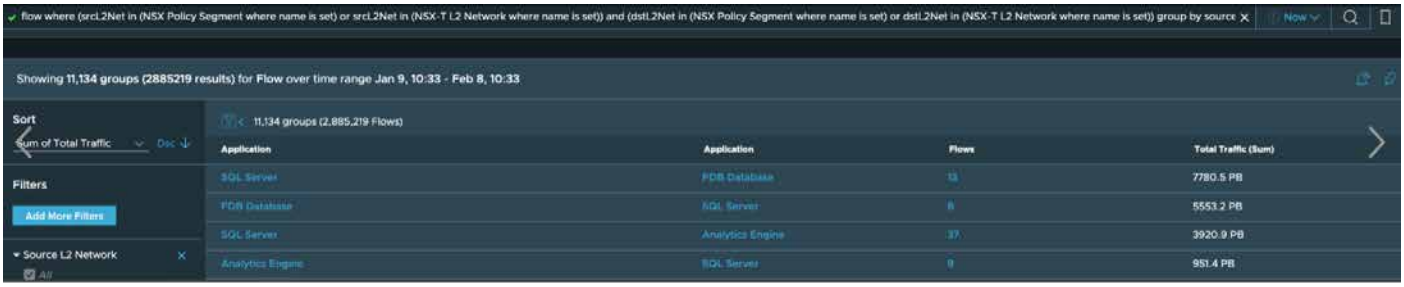

Figure 6: Example of total traffic between applications.

Clicking on the flow count will give details about the top flows causing this traffic. The flow will have details about the L2 networks and hosts involved.

If the application constructs do not exist in the system, you can get the top talking L2s. The following query helps get the top L2s contributing to the flow:

```
flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where name is set)) group by source L2 Network, destination L2 Network order by sum(Total traffic)
```

The query can be simplified if there are only NSX L2s:

```
flow group by source L2 Network, destination L2 Network order by sum(Total traffic)
```

If the top L2s are under different edge routers and there's heavy traffic between them, you can check if the bandwidth and network config of the tier routers/transport node can handle the traffic across the L2s.

Figure 7: Example of the top talking L2 networks by total traffic.

You can also get the TCP RTT and TCP retransmission ratio metrics with high value for L2/application pairs:

```
flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where name is set)) group by source L2 Network, destination L2 Network order by avg(Average TCP RTT)
```

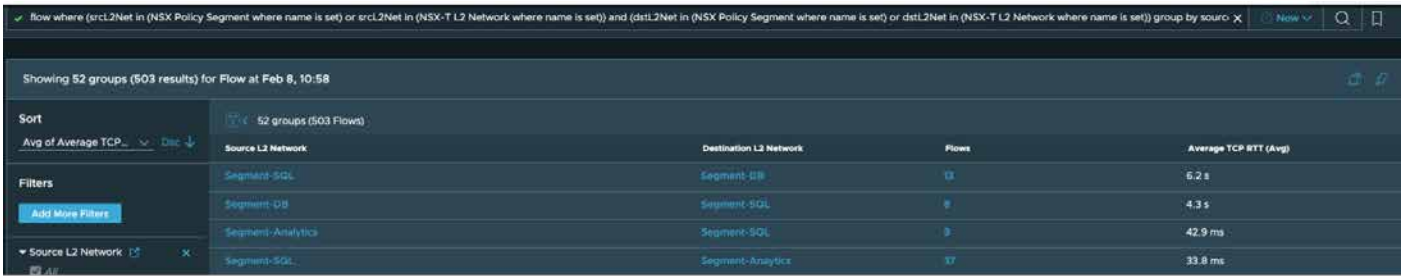

Figure 8: Example of a pair of L2 networks with a high average TCP RTT.

Figure 8 shows a list of NSX L2 pairs with TCP RTT values in descending order.

A similar query can be used for the TCP retransmission ratio:

```
flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where name is set)) group by source L2 Network, destination L2 Network order by avg(TCP Retransmission Ratio)
```

Figure 9: Example of a pair of L2 networks with a high TCP retransmission ratio.

If you want to get the metric values of the TCP RTT and the TCP retransmission of the application/L2 having the highest traffic between them, use the following query:

```
avg(Average TCP RTT), avg(TCP Retransmission Ratio) of flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where name is set)) group by source L2 Network, destination L2 Network order by sum(Total traffic)
```

Figure 10: Example of a pair of L2 networks with the total traffic, the TCP retransmission ratio, and the average TCP RTT.

Other metrics can be projected on the flow, such as session count, packets, byte rate, lost packets, source bytes, and destination bytes. The metrics are available under the metric section in the filters:

```
avg(session count),total(packets), avg(byte rate) of flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where name is set)) group by source L2 Network, destination L2 Network order by sum(Total traffic)
```

Figure 11: Example of a pair of L2 networks with total traffic, sessions, number of flow packets, and traffic rate.

You can also determine the services any application/L2 is talking to based on the port. This helps identify the services being accessed by the application or the VMs on the L2 network:

```
flow where source application = 'On-prem' and destination application != 'On-prem' group by port order by sum(Total traffic)
```

```
flow where source L2 Network = 'L2-Net' and destination L2 Network != 'L2-Net' group by port order by sum(Total traffic)
```

You can also get the top talking VMs and hosts in the top talking application:

```
flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where name is set)) and source application = 'App1' and destination application = 'App2' group by source vm, destination vm order by sum(Total traffic)
```

```
flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where name is set)) and source application = 'App1' and destination application = 'App2' group by source host, destination host order by sum(Total traffic)
```

Another essential insight is the highest TCP RTT between application and L2 networks:

```
flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where name is set)) group by source L2 Network, destination L2 Network order by max (Average TCP RTT)
```

```
flow where (srcL2Net in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set)) and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where name is set)) group by source Application, destination Application order by max (Average TCP RTT)
```

The "where" condition in all these queries can be removed if there is only an NSX L2 in the system.

Flows insights using L3/routers

Routers are critical components of every application. Different services communicate with each other via the routers. In case of an issue between the services communicating across the L3, it becomes crucial to determine the traffic across the routers.

To identify the amount of traffic across all VMs with a specific default gateway router, use the following query:

```
sum(bytes) of flow where vm in (vm where Default Gateway Router = 'name')
```

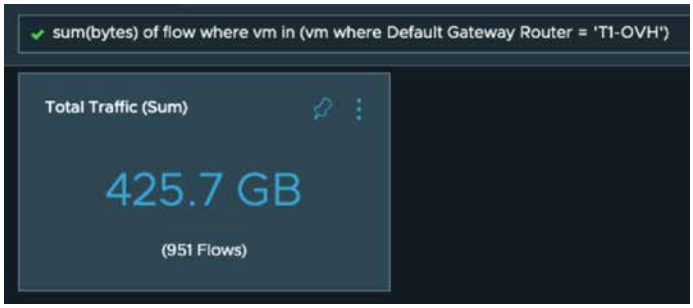

Figure 12: Example of traffic via a tier router.

To find out the individual traffic going from VMs with a specific default gateway router, use the following query:

```
flow where source vm in (vm where Default Gateway Router = 'name') order by Total Traffic
```

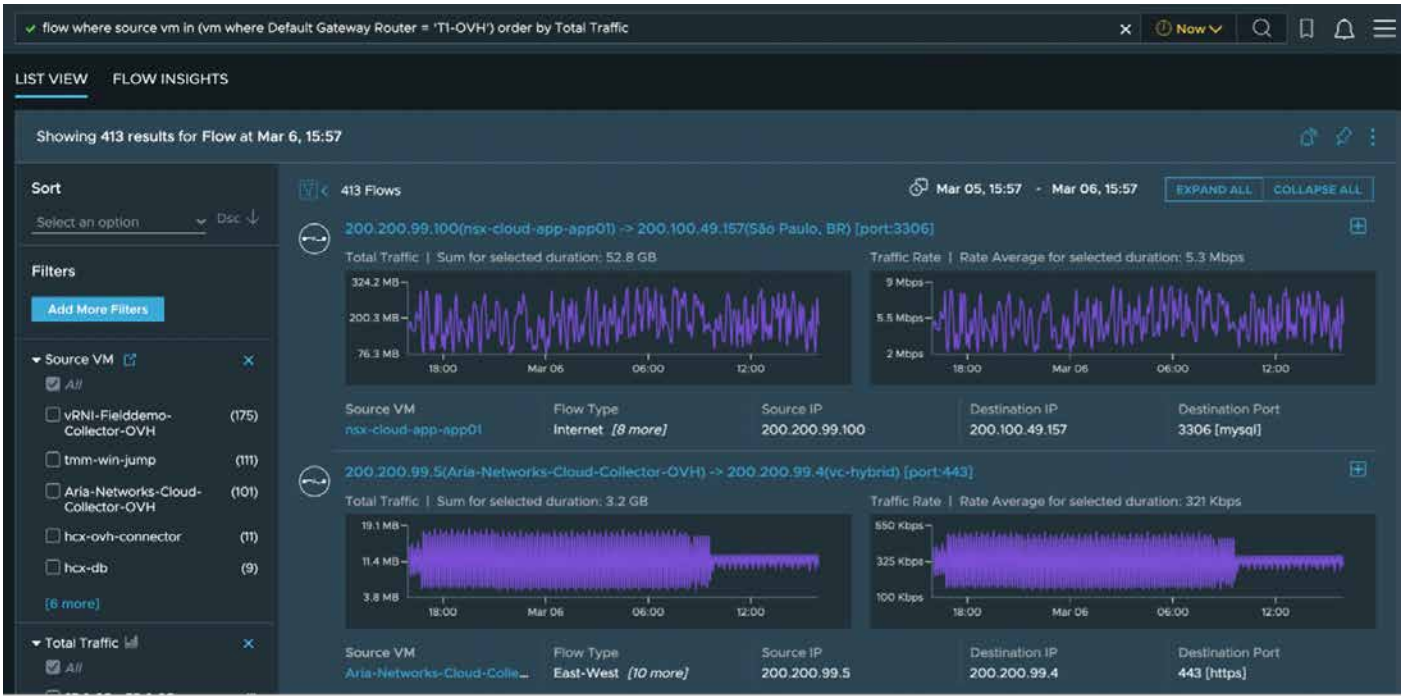

Figure 13: Example of flows with high traffic via a tier router.

Figure 14: Example of total traffic via a tier router to specific ports.

Figure 14 shows which services(ports) are being accessed by VMs with a particular default gateway. You can prefix the query with sum(bytes) to get the sum of the total traffic .

The following queries will help in providing insights for traffic across tier-1 routers.

Outgoing traffic from tier-1:

```
flow where source vm in (vm where Default Gateway Router = 'name') and destination vm not
in (vm where default gateway Router = 'name')
```

Incoming traffic to tier-1:

```
flow where destination vm in (vm where Default Gateway Router = 'name') and source vm not
in (vm where default gateway Router = 'name')
```

Traffic within tier-1:

```
flow where source vm in (vm where Default Gateway Router = 'name') and destination vm in
(vm where default gateway Router = 'name')
```

Incoming and outgoing traffic from a tier-1 router:

```
flow where (source vm in (vm where Default Gateway Router = 'name') and destination vm not
in (vm where default gateway Router = 'name')) or (destination vm in (vm where Default Gateway
Router = 'name') and source vm not in (vm where default gateway Router = 'name'))
```

Based on the insights from these queries, you can identify the elephant flows consuming the maximum bandwidth of the tier-1 router. If multiple services run under the same tier router, you can identify the VMs attached to the elephant flows and determine the service they belong to or are trying to access.

If there's an issue with the tier router, you can get further insights and narrow down the scope of the problem based on the L2 network connected to the tier router by using these queries:

```
flow where source vm in (vm where Default Gateway Router = 'name') group by source L2 Network
order by sum(Total traffic)
```

```
flow where vm in (vm where Default Gateway Router = 'name') group by L2 Network order
by max(Packet Loss)
```

In these queries, the group by can be performed on source, destination L2 pairs, and the VM can either be in source or destination of the flow.

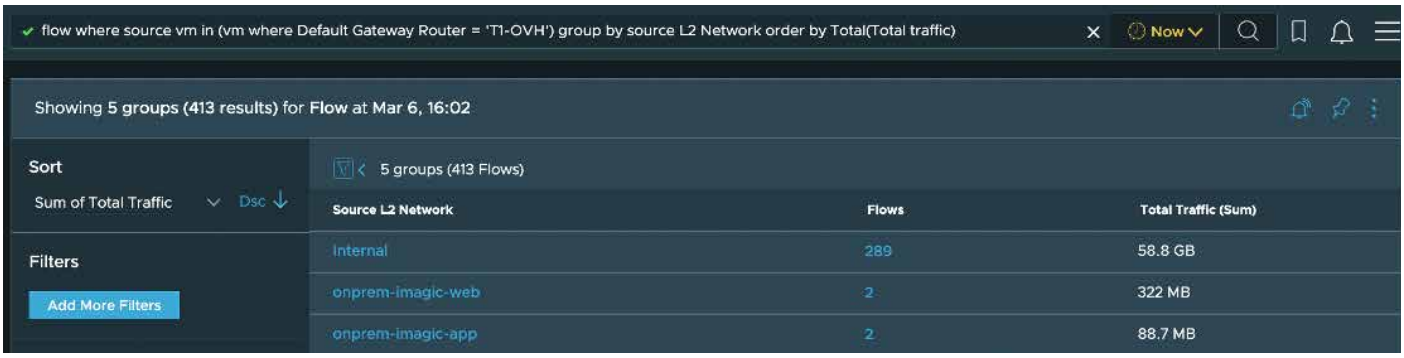

Figure 15: Example of total traffic from L2 networks connected to a tier router.

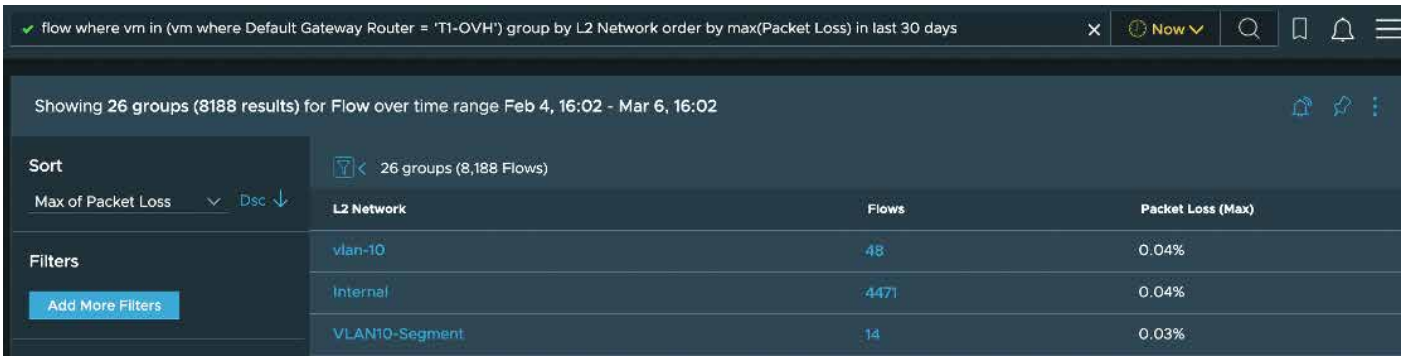

Figure 16: Example of packet loss on L2 networks connected to a tier router.

The tier-1 routers in NSX are connected to tier-0 routers.

The following queries will help in providing insights for traffic across tier-0 routers.

Outgoing traffic from tier-0:

```
flow where source vm in (vms where defaultVRF in (vrf where Default Gateway Routers in (vrf where Default Gateway Routers = 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF = 'name') and destination vm not in (vms where defaultVRF in (vrf where Default Gateway Routers in (vrf where Default Gateway Routers = 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF = 'name')
```

Incoming traffic to tier-0:

```
flow where Destination vm in (vms where defaultVRF in (vrf where Default Gateway Routers in (vrf where Default Gateway Routers = 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF ='name') and Source Vm not in (vms where defaultVRF in ( vrf where Default Gateway Routers in (vrf where Default Gateway Routers = 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF = 'name')
```

Traffic within tier-0:

```
flow where source vm in (vms where defaultVRF in (vrf where Default Gateway Routers in (vrf where Default Gateway Routers = 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF = 'name') and destination vm in (vms where defaultVRF in (vrf where Default Gateway Routers in (vrf where Default Gateway Routers = 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF = 'name')
```

Incoming and outgoing traffic from a tier-0 router:

```
flow where (Source Vm in (vms where defaultVRF in (vrf where Default Gateway Routers in (vrf where Default Gateway Routers= 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF = 'name') and destination vm not in (vms where defaultVRF in ( vrf where Default Gateway Routers in (vrf where Default Gateway Routers= 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF = 'name')) or (Destination Vm in (vms where defaultVRF in (vrf where Default Gateway Routers in ( vrf where Default Gateway Routers = 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF ='name') and Source Vm not in (vms where defaultVRF in ( vrf where Default Gateway Routers in ( vrf where Default Gateway Routers = 'name')) or defaultVRF in (vrf where Default Gateway Routers = 'name') or defaultVRF = 'name'))
```

The use case of flows from the tier-0 router is similar to the one described for the tier-1 router. If two applications are deployed under different tier-0 routers and there's heavy traffic between them, you can determine the traffic happening across the tier-0 router and top flows along with their metrics.

Note: To have accurate flow information from the tier routers, ensure that the default gateway property of the VMs is set correctly.

Flow insights using series() query

The series() search in VMware Aria Operations for Networks is a useful search that helps determine issues in the system, as illustrated in this section.

If a user experiences large spikes of traffic over an area of the network, VMware Aria Operations for Networks can help you determine the primary contributing factors to the surge in traffic. By using the series() search capabilities, you can aggregate metric results into a single metric or aggregate them in a grouped format.

For the examples in this section, the following search query was used for all flows in the test environment.

```
series(sum(bytes rate)) of flow
```

Using series(sum(bytes rate)) of flow as the search resulted in a single metric chart, as shown in Figure 17. This chart provides the sum of the bytes rate of each and every flow in the search result.

Figure 17: Example of the sum of the traffic rate for all flows.

Using the mouse over the chart, you can see the time stamp (19:00) associated with the 1.2Gbps spike in the environment.

To narrow down the results to a specific service causing the spike, you can take the same search and group it by port. This will take all flows and group them using the destination port of the flow. What you want for this result is a metric chart that mirrors the spike pattern seen in the first search.

After reviewing the page, you can see that port 2233 is the primary service attributed to the spike at 19:00:

```
series(sum(bytes rate)) of flow group by port
```

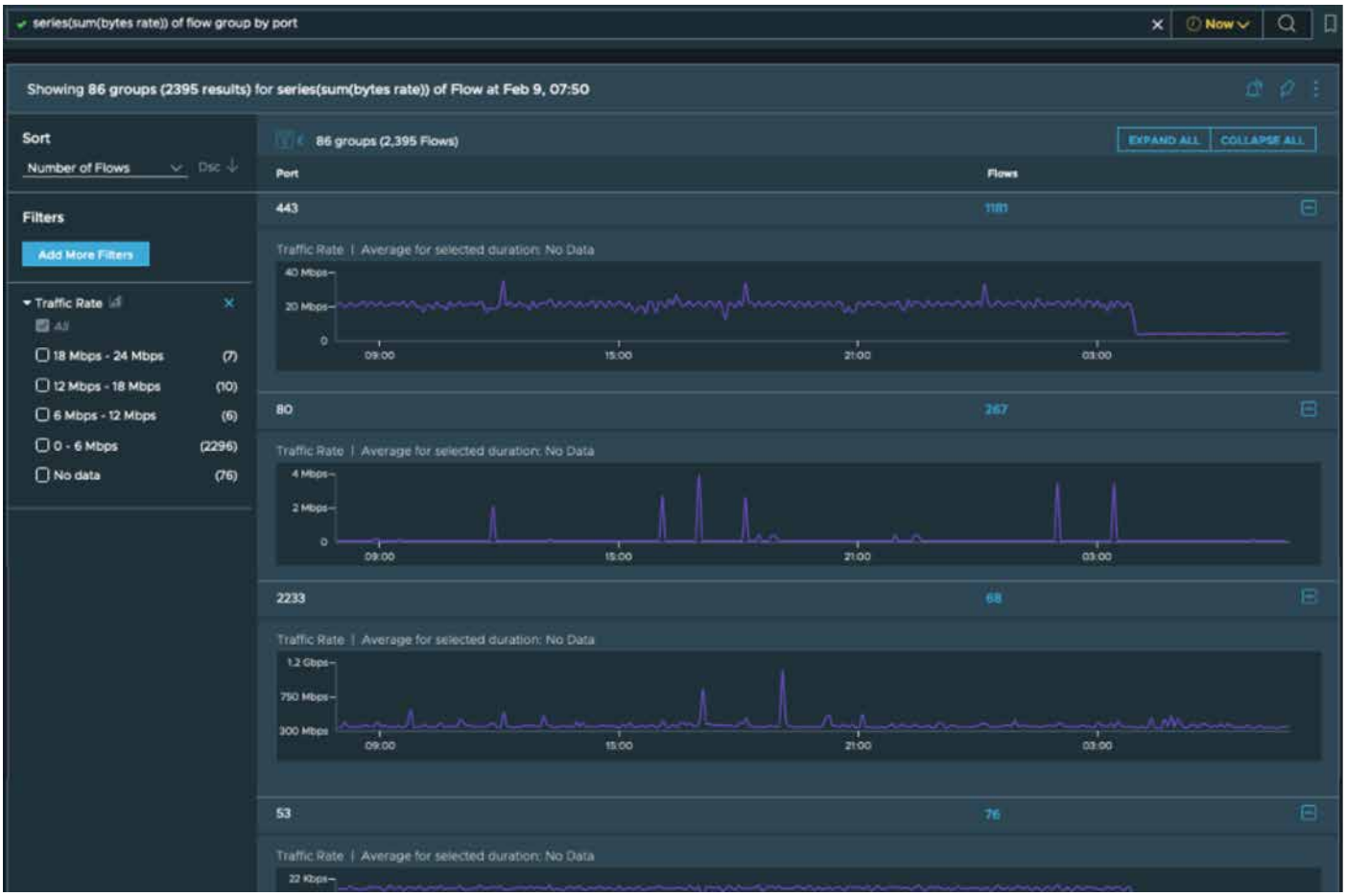

Figure 18: Example of the sum of the traffic rate on ports across all flows.

As a next step, you can modify the search to only include port 2233, and then group by the source or destination network to narrow down a specific VLAN that might be impacted:

```
series(sum(bytes rate)) of flow where port = 'number' group by source/destination l2 network
```

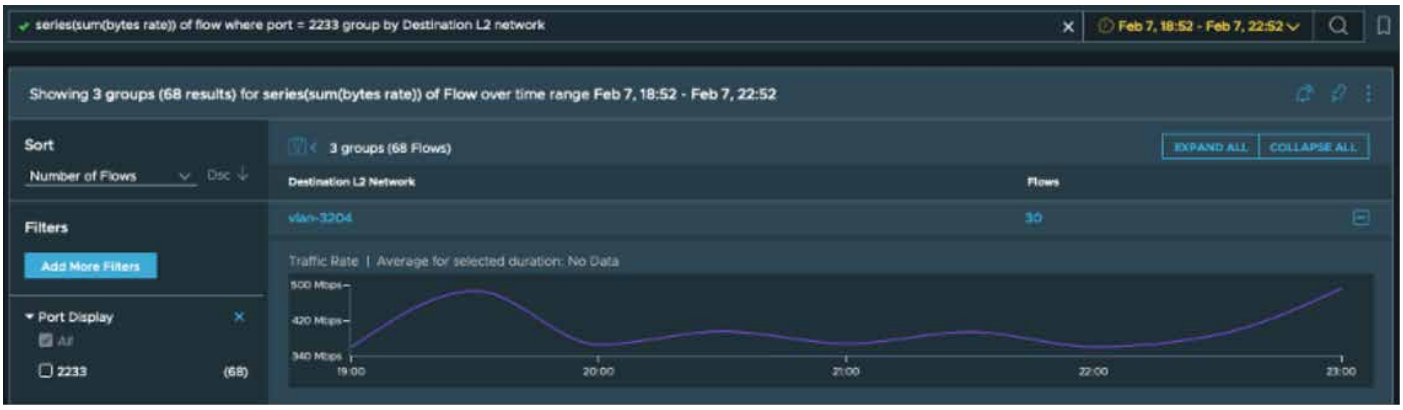

Figure 19: Example of the traffic rate for a particular port grouped by the destination L2 network.

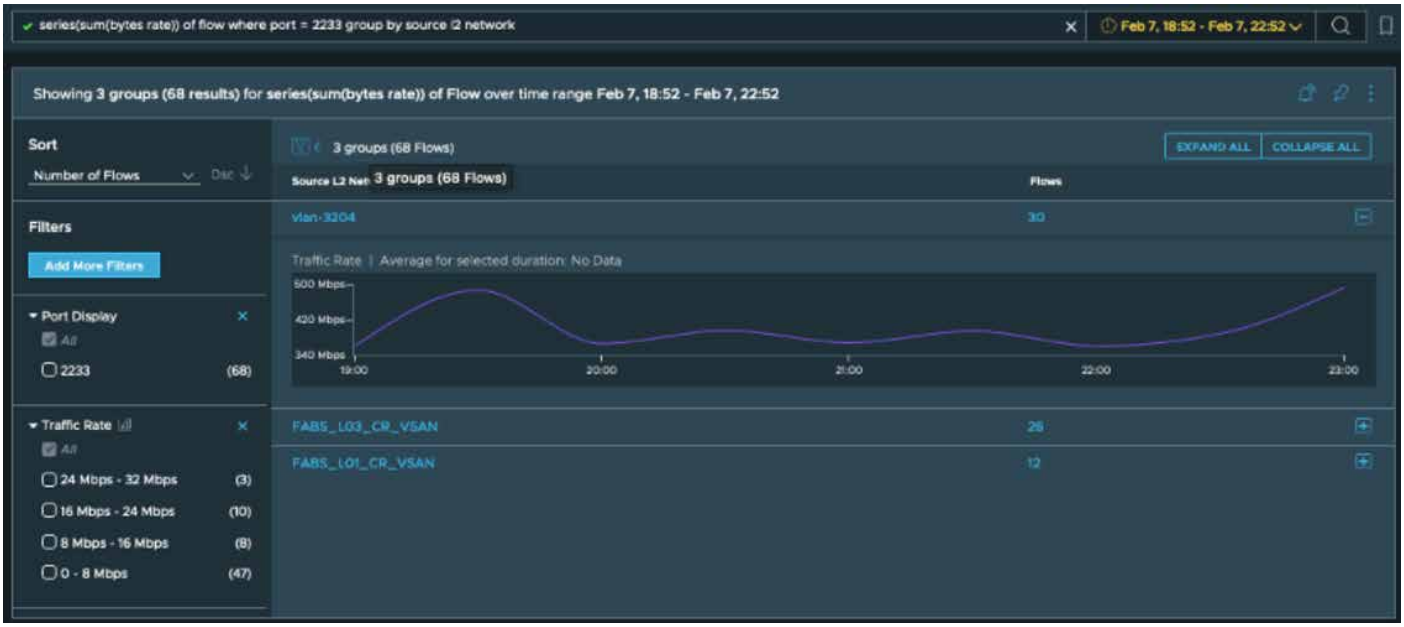

Figure 20: Example of the traffic rate for a particular port grouped by the source L2 network.

After identifying VLAN 3204 as the primary network attributing to this spike, you can dig further in by looking at the source or destination IPs. First, modify the search to also include the network, then group by the source or destination IPs:

```
series(sum(bytes rate)) of flow where port = 'number' and l2 network = 'network' group
by source/destination ip address
```

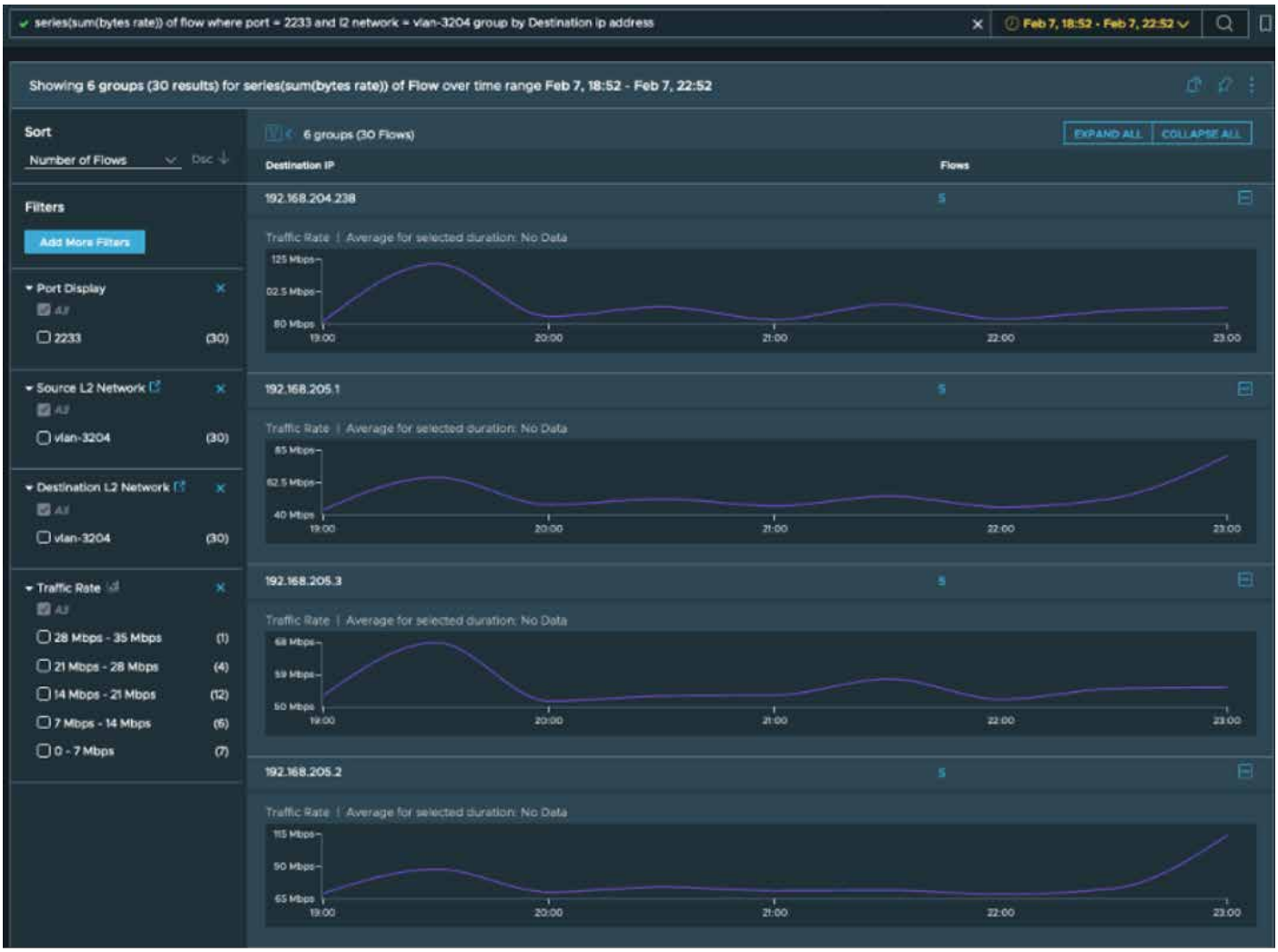

Figure 21: Example of the traffic rate on destination IPs for a specific L2 network and port.

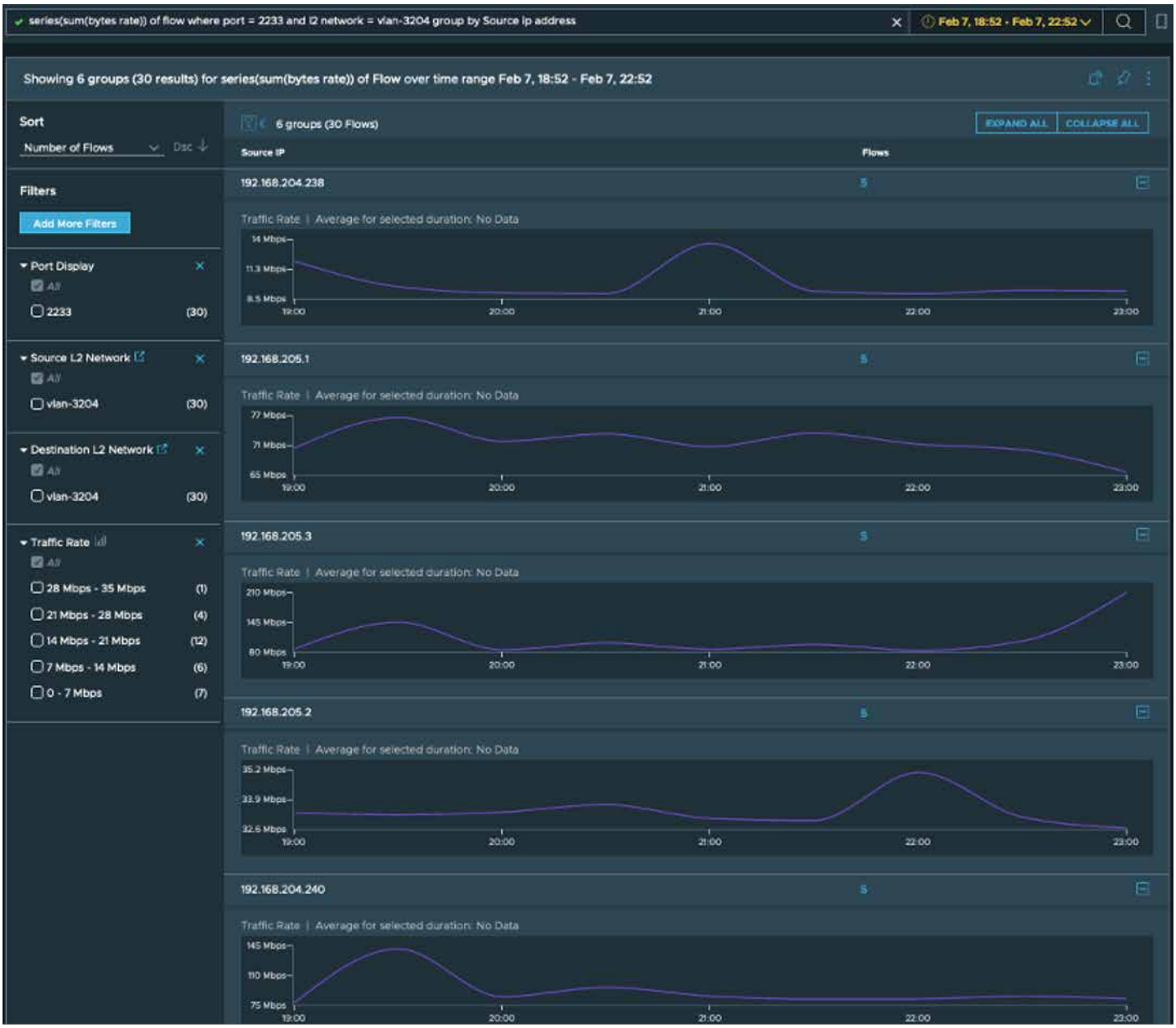

Figure 22: Example of the traffic rate on source IPs for a specific L2 network and port.

Using a series() query, you can determine the VMs, VLAN and port contributing to the surge in traffic.

To get the series data of the TCP RTT and the TCP retransmission, use the following query:

```
series(avg(Average TCP RTT)), series(avg(TCP Retransmission Ratio)) of flow where (srcL2Net
in (NSX Policy Segment where name is set) or srcL2Net in (NSX-T L2 Network where name is set))
and (dstL2Net in (NSX Policy Segment where name is set) or dstL2Net in (NSX-T L2 Network where
name is set)) group by source L2 Network, destination L2 Network order by sum(Total traffic)
```

Figure 23: Example of using a series() query to get TCP RTT data and the TCP retransmission ratio.

Monitoring using dashboards and widgets

Based on the insights from the previous section, if the network admin/app owner wants to view the critical flow data at a glance, the queries can be pinned on a dashboard, and the search queries can be saved. The dashboard will give the results of the queries with a single click.

The pinboard can be shared with other users to give them insights.

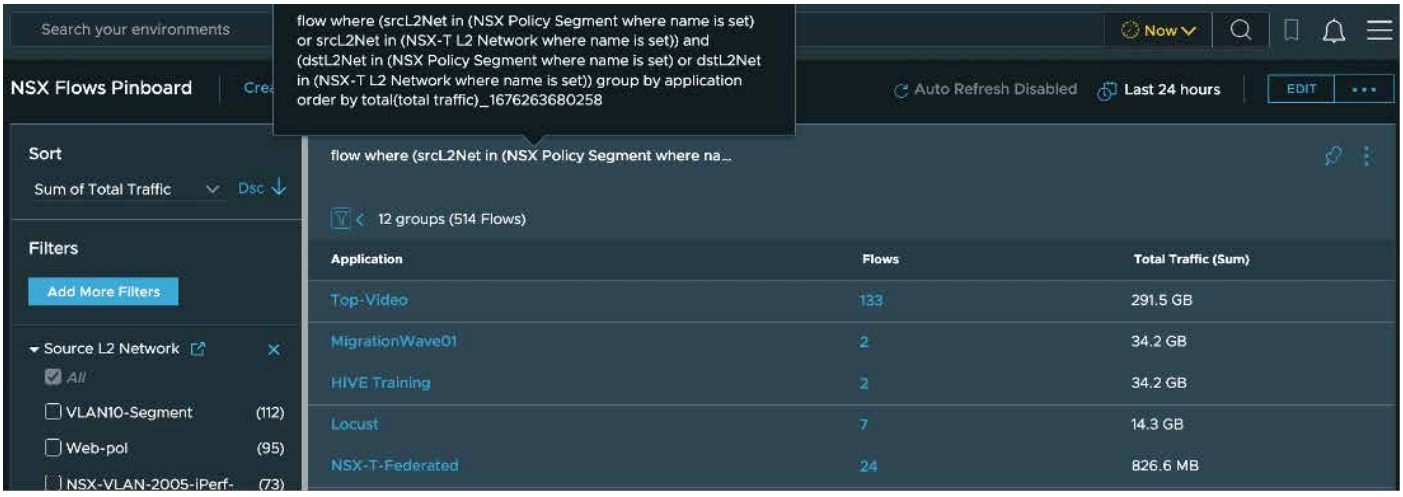

Figure 24: Example of a top talking application pinboard.

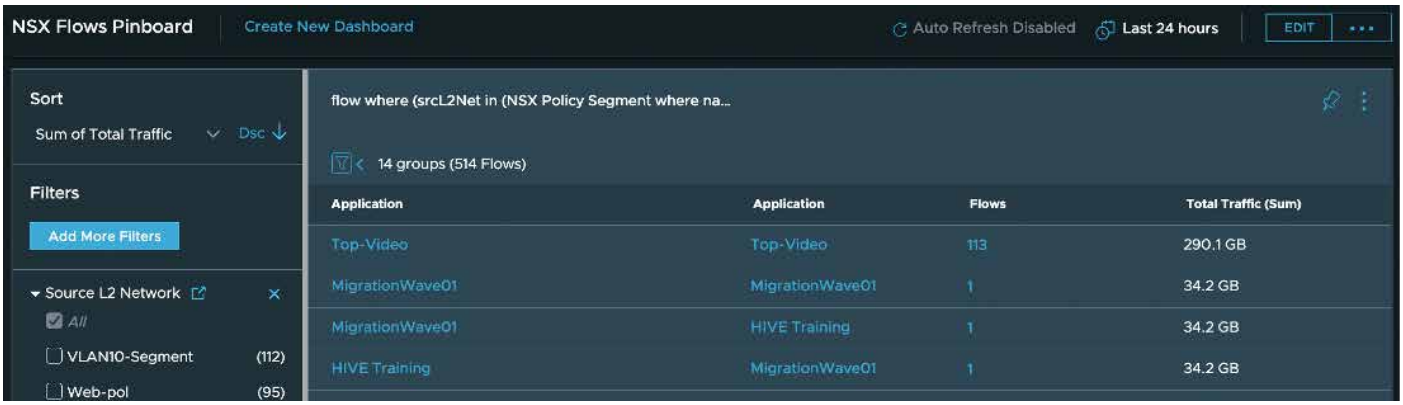

Figure 25: Example of a top talking application pairs pinboard.

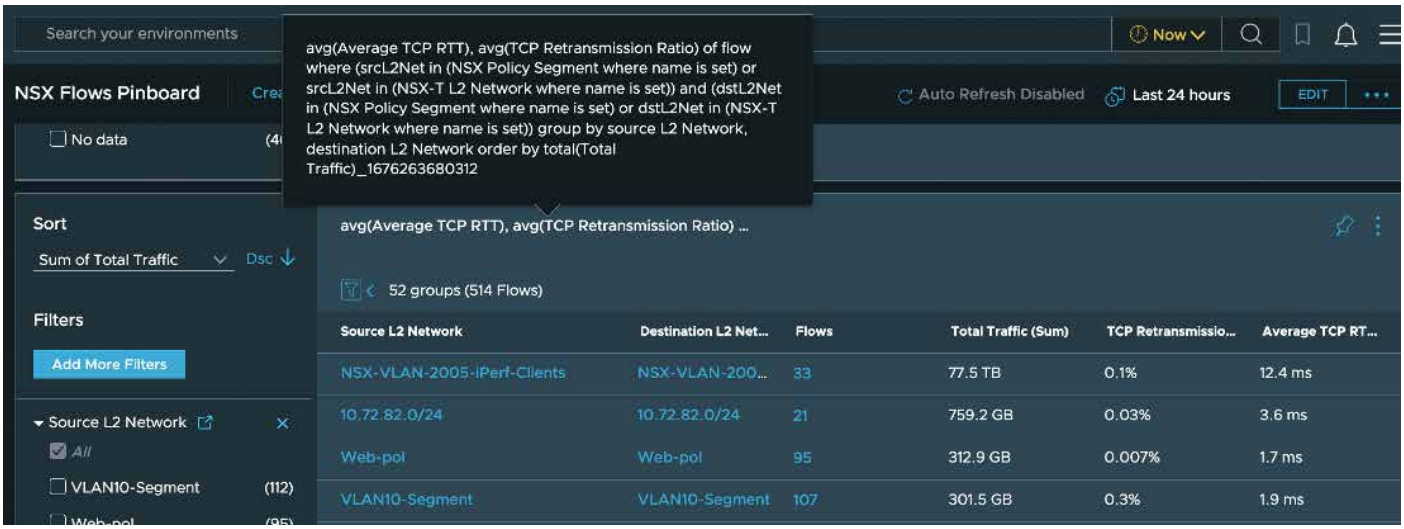

Figure 26: Example of a pinboard.

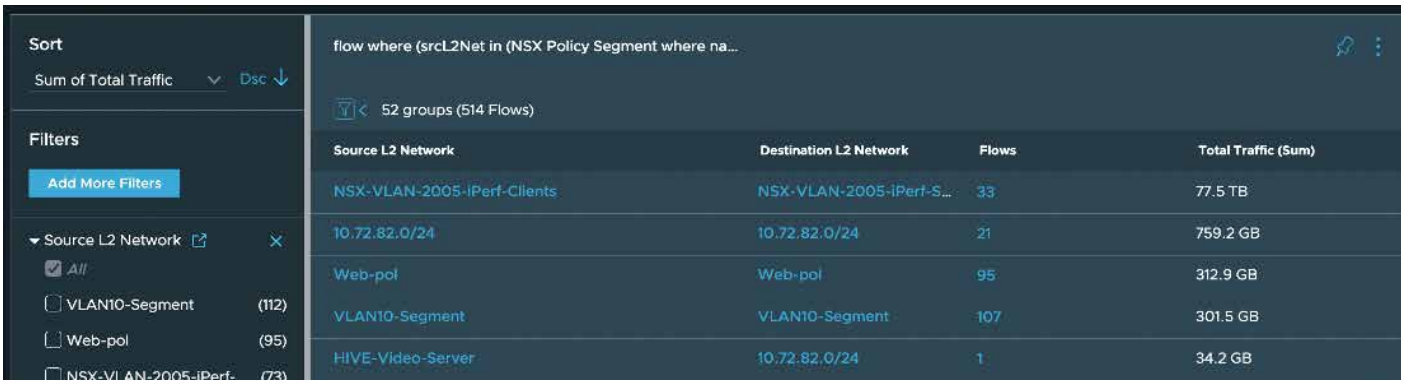

Figure 27: Example of a pinboard.

Threshold alerts

From the queries in the previous sections, you are now aware of the top talking applications/VMs and their networking constructs. Using VMware Aria Operations for Networks, you can configure threshold alerts for them. These threshold alerts will notify the user when breaching certain limits and that the application performance might be impacted.

For example, a threshold alert can be configured to notify the app owner when the VMs in one of the top talking applications faces packet drops. If the threshold is breached, the app owner can then investigate the packet drop.

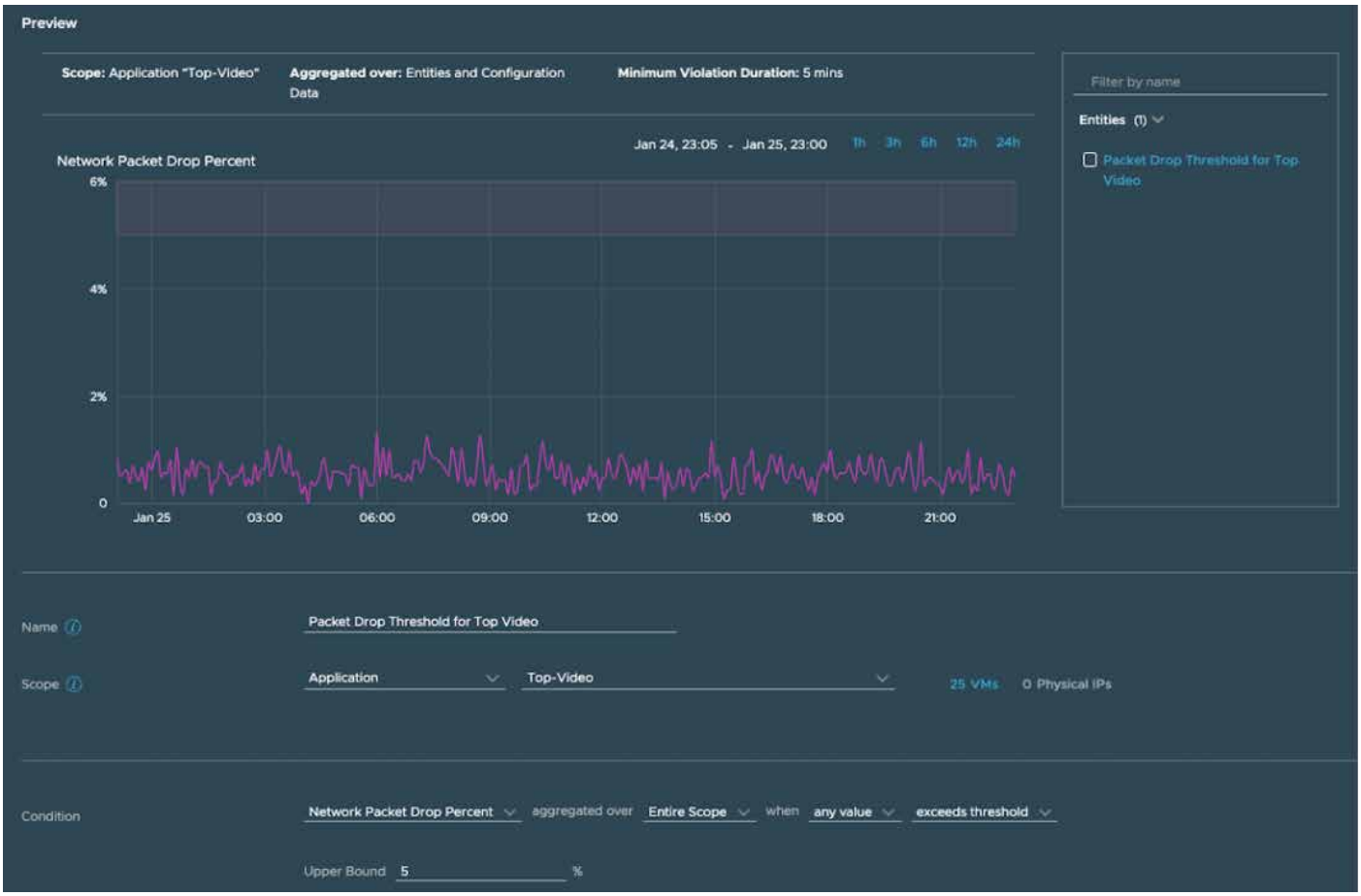

Figure 28: Example of configuring threshold alerts in VMware Aria Operations for Networks.

You can also configure a threshold alert to notify if the CPU usage of any VM in the application crosses a limit. This alert will help the user take timely action so that application performance is not affected.

Figure 29: Example of threshold alerts for CPU usage.

Threshold alerts can be configured for various other metrics, such as the TCP retransmission ratio, session, and total traffic.

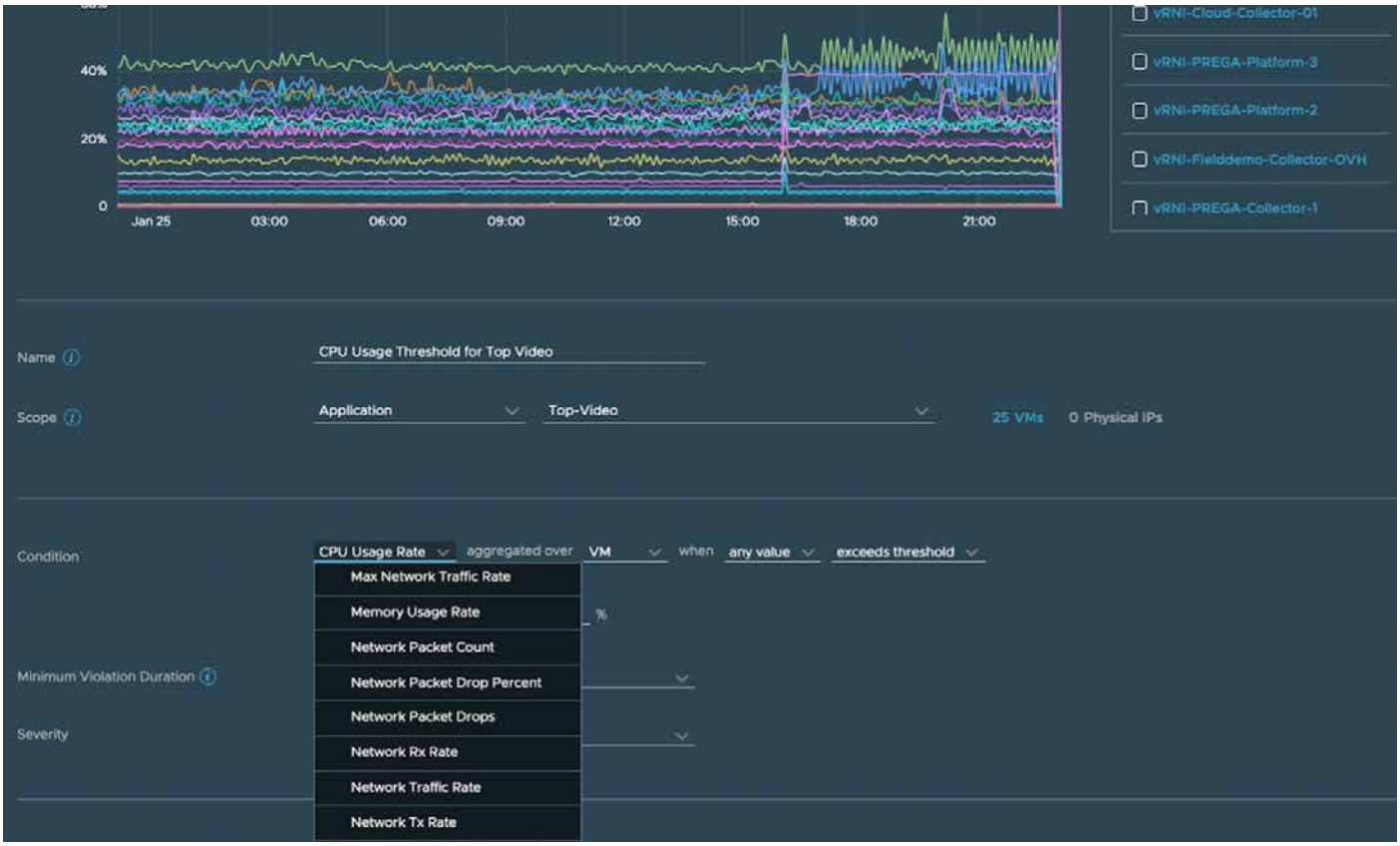

Figure 30: Example of various metrics for configuring threshold alerts.

The different components/tiers of an application can be associated with different L2s. The VMs on each L2 might be responsible for running a service of an application. For example, consider a web service running on VMs connected to L2 network Web-L2 and a database service running on VMs connected to L2 network DB-L2. In this scenario, it's important to ensure the VMs on each of these L2s are monitored for packet drops, memory usage, and the like. If the VMs of any L2 network face issues, the application performance will be impacted.

Threshold alerts can also be configured for VMs with a specific default gateway router.

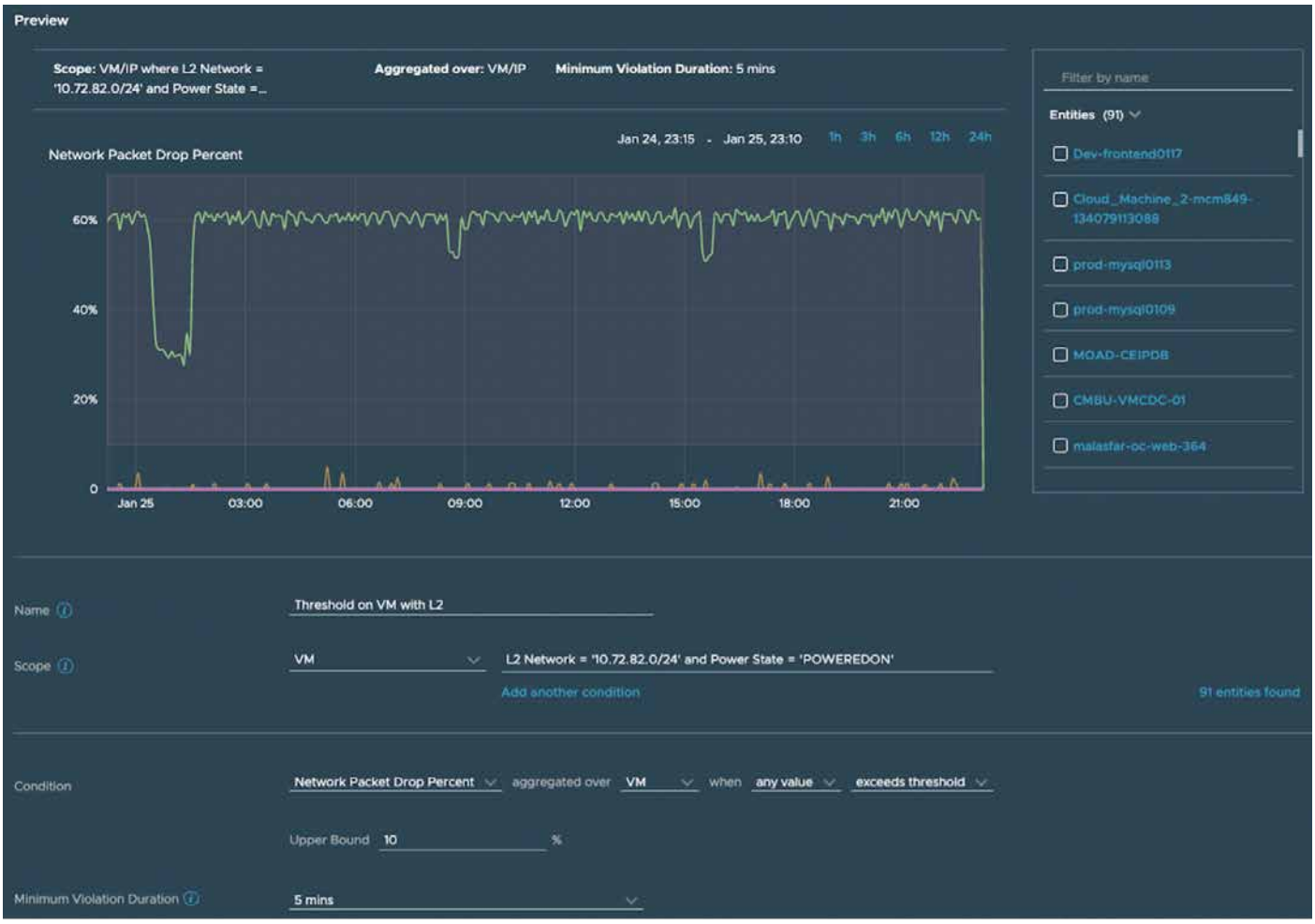

Figure 31: Example of a VM breaching a threshold.

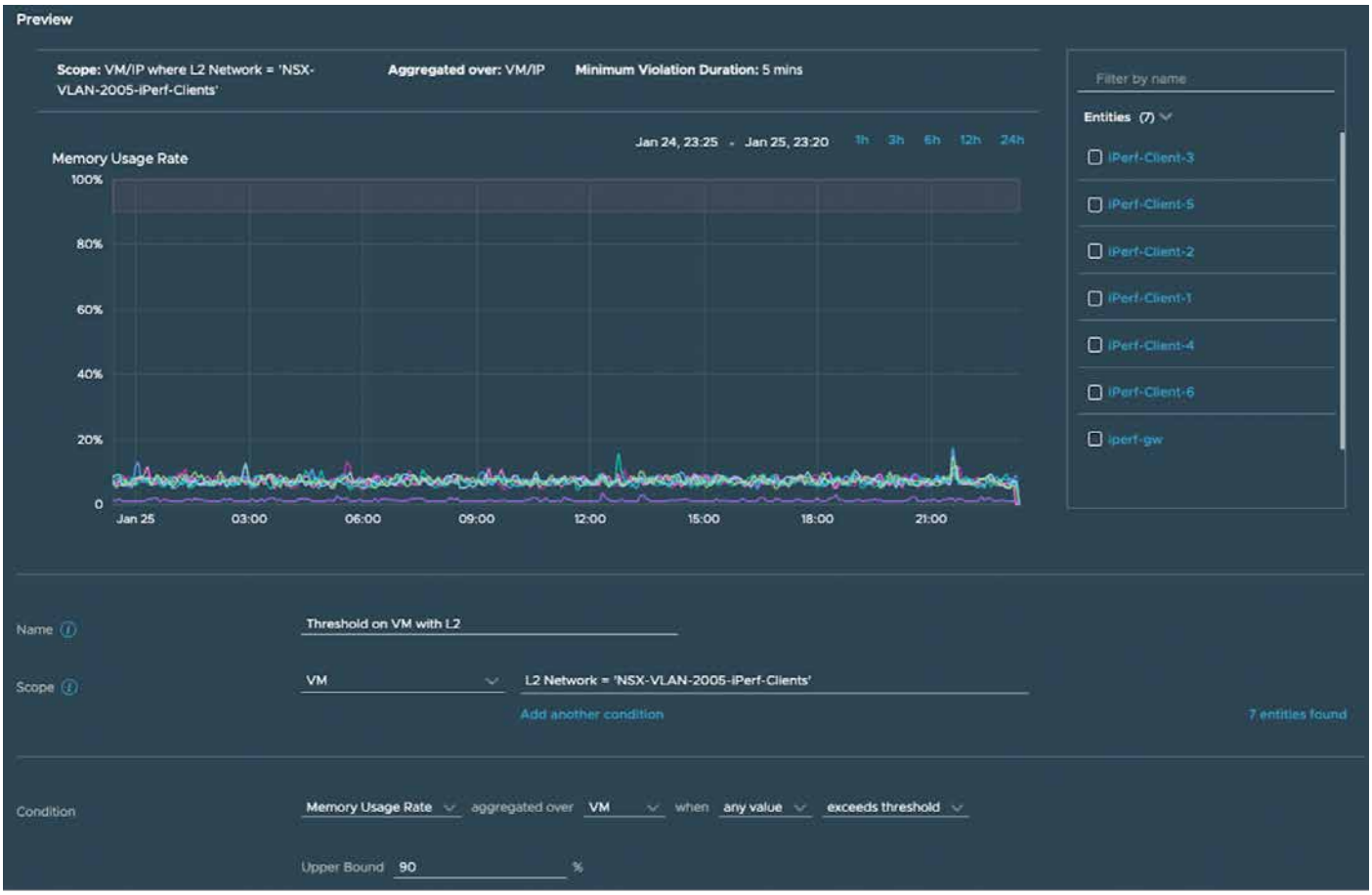

Figure 32: Example of a threshold alert on the memory usage rate of a VM.

A threshold alert can be configured on different parameters (TCP retransmission ratio, total traffic, session count, etc.) over flows with different conditions. For example, you can be notified if the TCP RTT of any flow between applications/L2 networks/VMs crosses a certain threshold.

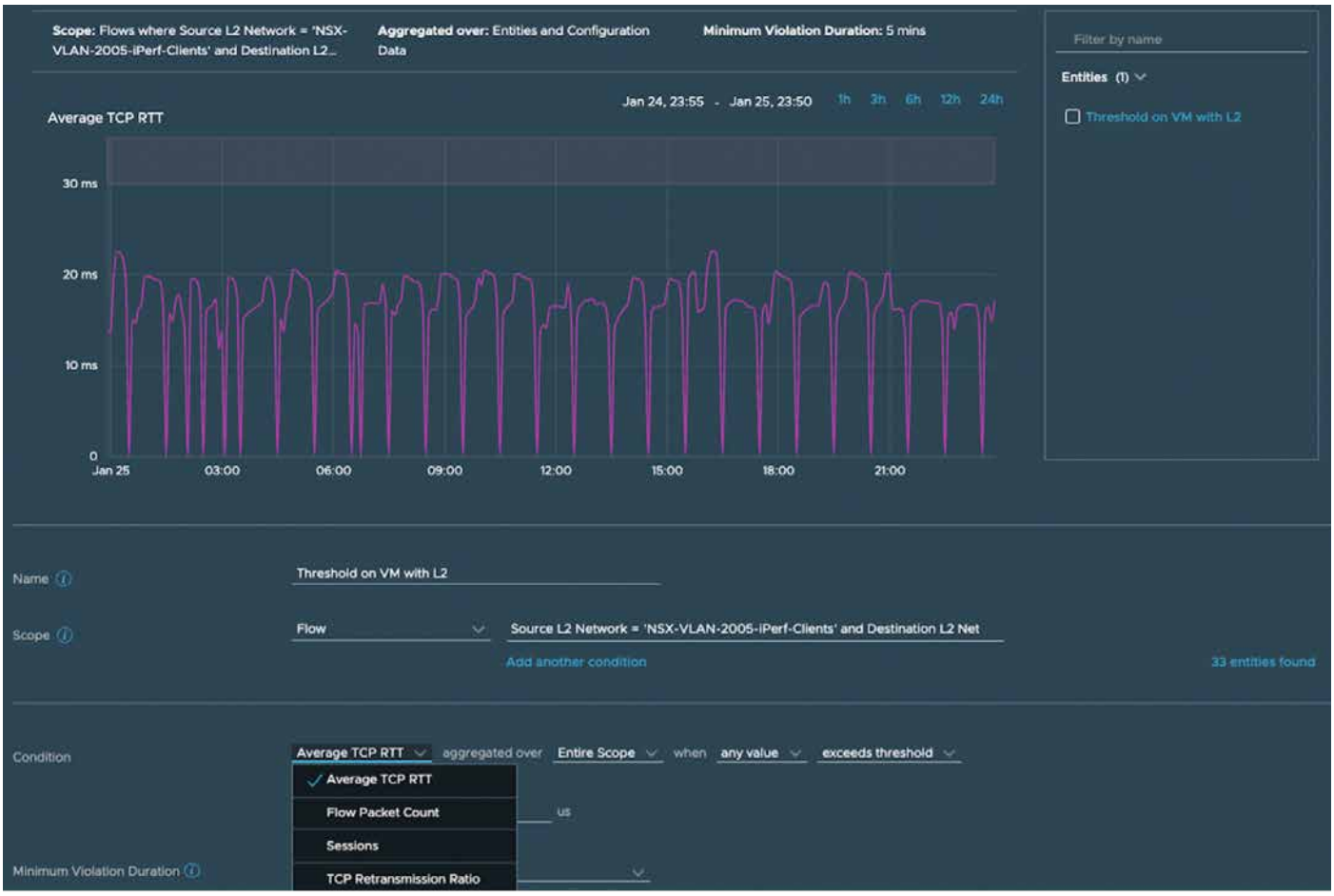

Figure 33: Example of a threshold alert for different metrics in flows.

Conclusion

The examples throughout this white paper show how you can use NetFlow information from VMware Aria Operations for Networks to run your applications better. By running the many useful queries in this white paper, you can answer questions about how your applications are running on your infrastructure and use flows to optimize your applications.

About the authors

Devraj Baheti is a senior member of technical staff for VMware Aria Operations for Networks in the Modern Applications and Management Business Group at VMware. He works on VMware Aria Operations for Networks integrations with VMware NSX and VMware Tanzu® Kubernetes Grid™ Integrated Edition. He has a passion for solving customer use cases and making them successful.

Trey Tyler has been assisting and educating customers to leverage VMware products to their fullest since 2014, starting with VMware vCloud® Air™, NSX, and now VMware Aria Operations for Networks. He enjoys seeing his customers' visions come to fruition.

Acknowledgements

The authors would like to thank Amol Vaikar for contributing to this white paper.

The authors would also like to thank Pravin Goyal, Ganesh Wagle, Sehjung Hah, and Matt Just for their guidance and valuable feedback on this white paper.

Item No: 1963900aq-tech-wp-netflow-mntr-trblst-a4 5/23