Networking Fundamentals

Part of the SolarWinds IT Management Educational Series

VOLUME 3

NetFlow Basics and Deployment Strategies

This paper examines NetFlow technology and implementation considerations. As part of the Network Fundamentals series, it is intended to provide an introduction to traffic flow analysis and guidelines for implementation.

solarwinds

Table of Contents

Document Revised: Oct 1, 2010

The Need for Flow Analysis

In general, Network Management Systems (NMS) perform a couple of basic functions: they tell you when something has failed and they tell you when something is overloaded. Most of the features of an NMS revolve around providing this information in different formats such as reports, alerts, Syslog viewers, trap viewers, et cetera. They answer the questions **what happened** and **when**. But when the issue is performance based, there is very little information to answer the *why* questions, especially with a saturated WAN link, one of the most common performance issues.

As a Network Management Engineer, one will eventually get the call — "Users in site X are complaining the network is slow." Using an NMS, you might find that the link between site X and the headquarters is saturated. You could have received a couple of text messages or pages from the NMS telling you when the link became saturated. The problem is you don't know why the link is saturated. If you could see what type of traffic is using the link, you could see what is causing it. That is what NetFlow, and other related flow analysis protocols do, they capture information **about the nature of the flows**.

Before NetFlow was around, there were some very cumbersome ways to do this, such as placing Y cables on WAN connections allowing a special protocol analyzer appliance to be connected when a link showed unusual behavior. This was expensive, required extra hardware and the results were hard to decipher. A new technology called RMON came about where a protocol analyzing agent was embedded in network equipment. The problem with RMON was that it only applied to LAN connections and bandwidth issues happen much more frequently on the WAN. RMON2 was an improvement as it added fields for network and application layer monitoring as well as support for WAN technologies. Cisco then launched NetFlow and being that Cisco has such a great share of the internetworking equipment market, NetFlow fairly quickly became the defacto standard.

With an NMS and Netflow deployed, you are able to see "what" — a link has became saturated, "when" — the time the problem began and "why" — **the nature of the traffic** on the link. We'll see why I call it the nature of the traffic in the next section.

For the purpose of this paper, I'll focus on NetFlow, but I'll also point out the differences and similarities with other flow monitoring technologies as appropriate.

How does NetFlow Work?

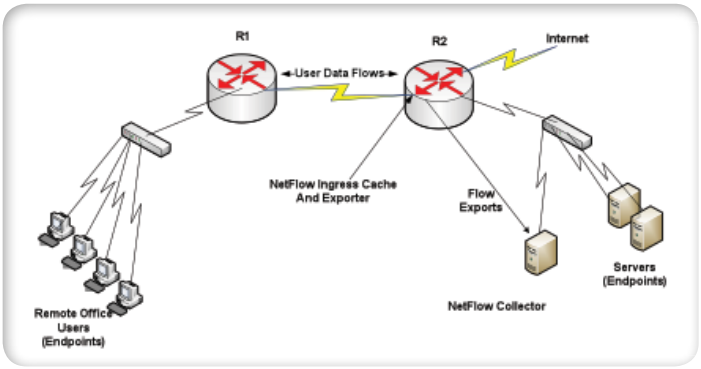
The term NetFlow is often used interchangeably between the three major components of NetFlow technology:

1. The NetFlow cache
2. The NetFlow exporter on the router or switch
3. The NetFlow collector used to analyze the flow information.

The NetFlow cache is the active monitoring of traffic and only exists on a NetFlow enabled device. The NetFlow exporter involves sending completed flow information from the device to a NetFlow collector, such as Orion NetFlow Traffic Analyzer.

Here is what the whole thing looks like graphically in a much simplified environment (see **Figure 1**).

Figure 1.



Here's what is happening step-wise:

- Users in the remote office are accessing the information from the corporate servers and the Internet. In NetFlow terminology, the machines participating in a flow are known as endpoints.
- As the users' data flows into R2's WAN interface, the NetFlow ingress cache is making records about the flows and saving them in R2's memory.
- As flows expire (we'll cover how that happens later) R2's NetFlow exporter sends the expired flow information to the NetFlow collector.
- The NetFlow collector stores and presents information about user flows.

This might seem simple a first glance but there is a lot going on to make all this work. To best understand all the moving pieces we'll examine the NetFlow cache first, and then look at the exporter and collector.

The NetFlow Cache

Everything in NetFlow begins with the flow cache. The cache has a couple of basic jobs to do:

- Interrogate data headers and either mark it as a new flow or add to part of an existing flow.
- Keep track of flow timers and other factors and when a flow is considered *complete*, send it to the exporter (if one exists) and delete the flow information off of the device. This process is known as flow aging.

The NetFlow cache only keeps information on current or non-expired flows. This brings up an excellent question: what constitutes a flow and how does a flow expire? NetFlow v5 is the most common version in use so we'll take a look at the v5 flow format. As data enters the NetFlow enabled interface, the IP header is examined and flows are defined as having values that match the following 7 key fields uniquely:

- Source IP address.
- Destination IP address.
- Source port number.
- Destination port number
- Layer 3 protocol type
- ToS byte value
- The IfIndex number, also called the logical interface number

When a packet enters the NetFlow enabled interface and **all seven of these key fields match** an existing flow, it is not considered a new flow but part of the existing flow. If any part of the seven key fields doesn't exactly match an existing flow, it is then a new flow and a new flow record is created.

Consider the below table (Table 1) of active flows in a cache.

Notice that these are all unique, as none of them duplicate any of the other

Table 1.

Key Field	Flow 1	Flow 2	Flow 3	Flow 4
Source IP	10.10.1.1	10.10.1.1	10.10.2.1	10.10.3.35
Destination IP	10.10.2.55	10.1.23.1	10.10.2.253	10.10.2.23
Source port	21	8080	21	443
Destination Port	21	8080	21	1122
Protocol	17	6	17	6
ToS	184	184	184	184
IfIndex	1	1	1	1

in all seven key fields. Now consider a new IP packet enters the NetFlow interface with the following fields:

- Source IP = 10.10.2.1
- Destination IP = 10.10.2.253
- Source port = 21
- Destination port = 21
- Protocol = 17
- ToS = 184
- IfIndex = 1

Because all seven key fields match Flow 3 exactly, a new flow record is not created and information in this packet header is added to the Flow 3 record. But what information is added to flow 3 if all the fields match? The matching fields above are the **key fields**. NetFlow v5 records also contain **non-key fields**. These non-key fields are stored in the flow record and these counters are updated when packets of an existing flow are detected. The non-key fields include:

- Bytes
- Packets
- Output interface IfIndex
- Flow start and finish time
- Next hop IP
- Network masks
- TCP flags
- Source and destination BGP AS numbers

Note that some of these non-key fields come from the new packet and some are achieved by flow cache calculations, for example the flow start and finish times. This information cannot be derived from a field in the packet but is a function of the cache marking the time the first packet is seen in a flow and the time the flow expired.

Let's take a look at the NetFlow cache on R2's WAN interface using the **show ip cache verbose flow** command (see **Figure 2**).

Figure 2.

Seeing as this is an introduction to NetFlow, I won't go into each field in detail but notice there are four sections to the flow cache display:

- Packet size distribution.
- Cache statistics and status.
- Protocol distribution.
- Active flow records.

The cache output can be a valuable realtime trouble shooting tool. The problem is that the output is not presented in the most readable of formats and the data tends to move quickly, as it is realtime. One thing you don't see in the flow information is any of the payload data, the application information the user or device is sending. NetFlow never looks into the payload, so it can't determine anything past layer 4 information. Therefore the information NetFlow gathers tells us about the *type* or **nature of the traffic** as interpreted from the IP header.

Using the default settings in NetFlow, all IP packets are interrogated and recorded. There are settings available to use a sampling algorithm but this is rarely implemented. S-flow uses sampled flow collection by either statistical random sampling or by timer-based sampling, depending on the configuration applied.

The NetFlow Data Exporter (NDE)

NetFlow devices have a limited amount of memory to store flow information, so at some point the device has to make room for new flows. This is where flow aging and exporting comes into play. The NetFlow-enabled device keeps track of several factors regarding the flows and the status of the cache itself. Here are the factors the device uses to age flows and either delete them or export to a collector then delete. These are listed in order of precedence.

- Cache maximum size is reached.
- A TCP connection has been terminated by a RST (reset) or FIN (finish) flag in the flow.
- An active flow timer or inactive flow timer limit is reached.

The NetFlow cache size is configurable on most Cisco routers up to 524,288 entries. Each entry uses a minimum of 64 bytes of memory. Once the cache maximum size is reached the device exports using rules to lower the cache count as quickly as possible. This process will export flows when none of the other mentioned export triggers have been reached.

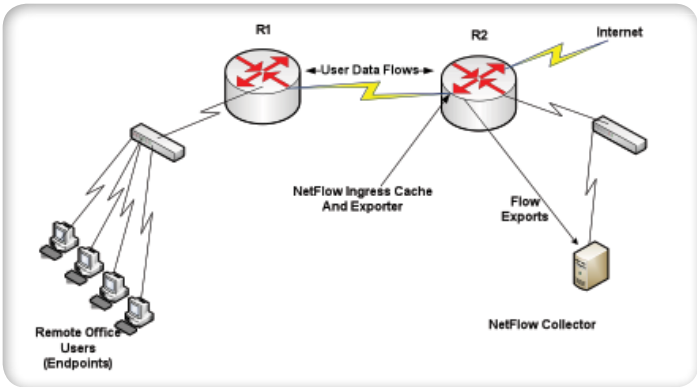
By definition, when an RST or FIN flag are set in a TCP connection the connection is closed. The Netflow device will export any flow when one of these flags is detected within the flow.

The last in order of precedence are the flow timers. The active flow timer tracks the first packet of all flows and exports the flow data, if it is still active after 30 minutes (default setting) of active flow time. The inactive flow timer marks the time the last packet in a flow was added. If a new packet in an existing flow is detected, the timer is reset. If no new packets in a flow are detected within 15 seconds (also default) of the last packet, the flow data is exported. So, the router is examining flow keys for each packet received, updating active flows or starting a new flow record, keeping a start timer on every flow and updating the active timer on each packet in a flow.

Keep in mind that what has been described here is the NetFlow v5 export format.

Let's take a look at a section of the network diagram we saw earlier (see **Figure 3**). Here we'll focus on the NetFlow cache and exporter on R2 as well as the exported flows going to the collector.

Figure 3.



We saw an active NetFlow cache in the previous section. Now, we'll look at an exporter configuration and examine some packets sent from the exporter.

Here is the configuration of our exporter (see **Figure 4**):

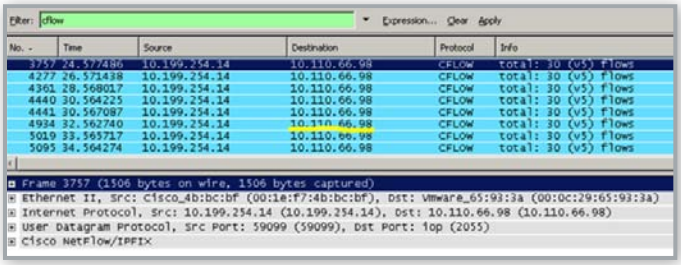
Figure 4.

```
R2#
R2#sho ip flo int
FastEthernet0/0
  ip flow ingress
FastEthernet0/1
  ip flow ingress
  ip flow egress
R2#sho ip flo exp
Flow export v5 is enabled for main cache
  Exporting flows to 10.199.15.51 (2055) 10.110.66.98 (2055)
  Exporting using source IP address 10.199.254.14
  Version 5 flow records
  21997443 flows exported in 733249 udp datagrams
  0 flows failed due to lack of export packet
  6 export packets were sent up to process level
  0 export packets were dropped due to no fib
  0 export packets were dropped due to adjacency issues
  0 export packets were dropped due to fragmentation failures
  0 export packets were dropped due to encapsulation fixup failures
R2#
```

And here is the resulting export as capture by a protocol analyzer at the second IP address of the "Exporting flows to..." line. (see **Figure 5**).

What we are seeing in the packet capture is the raw data as it comes into a NetFlow collector. Note that the destination IP address in the packet capture is the second IP address listed in the "Exporting flows to..." line of

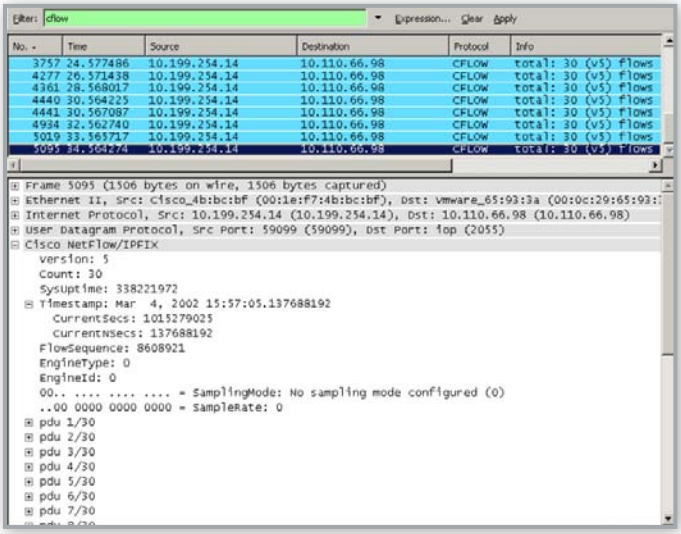
Figure 5.



the telnet session. The source of the packet capture is the interface on the NetFlow router sending flows to our protocol analyzer. The **show ip flow interface** command shows us that the FastEthernet0/0 is configured to collect only inbound (ingress) flows while the FastEthernet0/1 interface is configured to collect inbound and outbound flows. The exporter groups multiple flow records as Protocol Data Units (PDUs) together to reduce the total protocol overhead of exporting. The exporter will place up to 30 PDU's in a single export. This is seen above as each export packet has 30 flows.

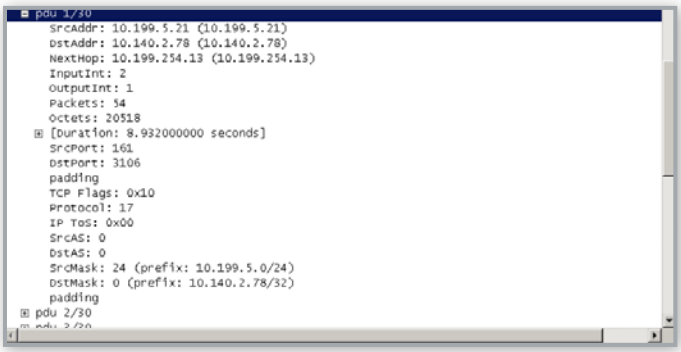
Here is the lower protocol section with the Cisco NetFlow/IPFIX layer expanded. (see **Figure 6**).

Figure 6.



In Figures XX we see the individual flows represented as PDU's and with further expansion below, we can see the information in a single flow record (see **Figure 7**).

Figure 7.



Notice that the seven key fields (Source and destination IP and ports, protocol, ToS, and input interface index number) are present (they define a flow!). The rest are all non-key information.

Remember our goal with NetFlow is to determine who and what is using the bandwidth. Examining the above capture we see that the **who** is the machine at the source IP address, 10.199.5.21, which could be resolved to a machine name with DNS. The **what** is given in the source port line and the protocol line, port 161(SNMP) over protocol 17 (UDP). Given that there are 54 packets in this flow in less than 9 seconds we can deduce that this is most likely automated SNMP requests or some mad typing skills! This is only one out of thirty flow records in this capture containing 29 exports over about a 15 second period. This is a whole lot of information in a short period of time — certainly more than a human could consume and understand in such a short amount of time. This is where the NetFlow Collector comes in. Before we look at the collector, it is worth pointing out that other versions of NetFlow can use other timers and export mechanisms. Here is a brief description of the other NetFlow versions:

- V1 — AKA the router killer. It is still out there on very old equipment.
- V2 to V4 — Development versions only, not released.
- V6 — Created to meet a single customer's needs but no longer supported. (There must be some good stories behind that one!)
- V7 — Catalyst-specific export
- V8 — Allows for the router to preprocess flow information before sending to the collector thus reducing the export traffic. This version is designed for very high throughput Service Provider devices and is not widely used. Unless the NetFlow data is being used for accounting, sampling in preferred.
- V9 — Quickly becoming the new standard. Using flow templates, v9 implements flexibility into the definitions of key and non-key fields in a flow. This means that when a new definition of a flow or a new set of key or non-key fields is needed, it is not necessary to create a new NetFlow version. Flexible NetFlow!
- V10 — IETF standard based flow analysis — IP Flow Information Export or IPFIX. This standard is based on the NetFlow v9 export format.

The NetFlow Collector

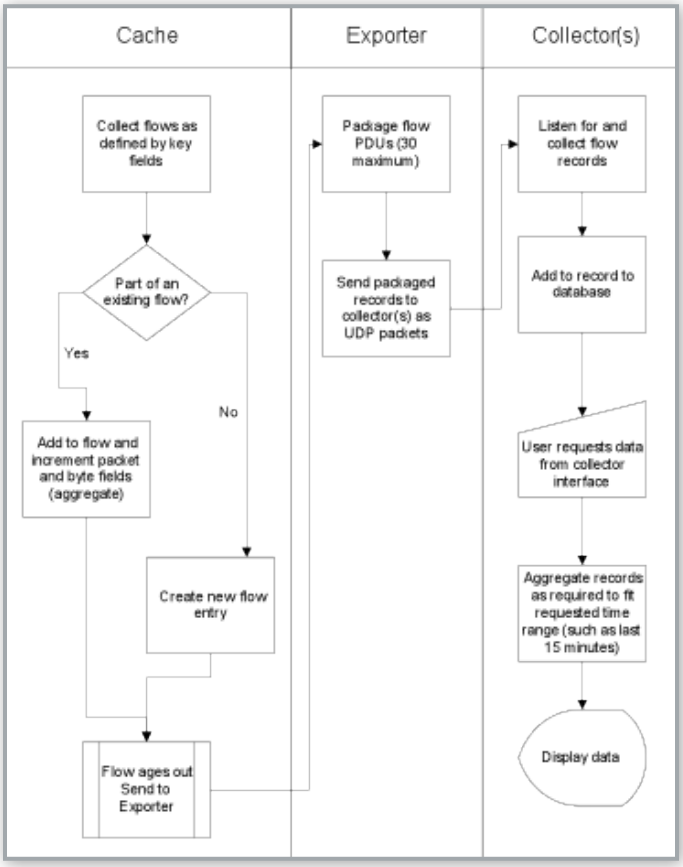
The collector has three main tasks, very similar to an NMS but using different data gathering methods. These are:

- Receive the flows as they are sent from exporters.
- Store and aggregate the data.
- Present the data.

In section 2 we saw a capture of 29 v5 exports, each carrying 30 flow records in about 15 seconds. This equates to a flow rate of 58 flows per seconds (fps), which is considered low. On a mid-sized network we might have 50 or so flow exporters. Assuming the same flow rate from each we now have 2900 fps. Flow rates in large networks are measured in tens of thousands of flows per second. Flow rates vary constantly depending on the number of active user connections detected by the cache.

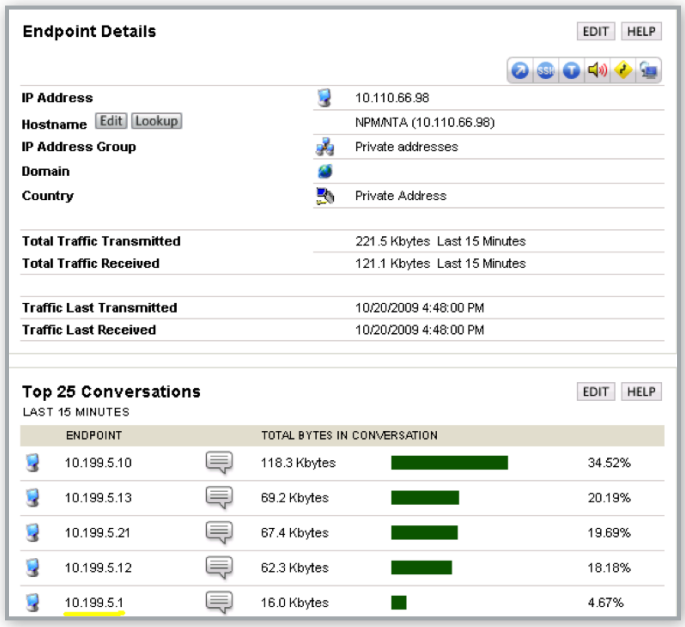
The collector listens on a port (UDP 2055 in our example) for flows from multiple sources. The collector aggregates information received from the exporters, depending on the time frame requested by the user, using key and non-key fields similar to those used in the cache. So, flows can be aggregated twice, once by the cache and once by the collector. Below is a flow chart showing the interactions of the three NetFlow core functions, the cache, exporter and collector. (see **Figure 8**).

Figure 8. Netflow Core Functions



Now we'll take a look at some sample output from my favorite collector, SolarWinds Orion Netflow Traffic Analyzer (NTA). (see **Figure 9**).

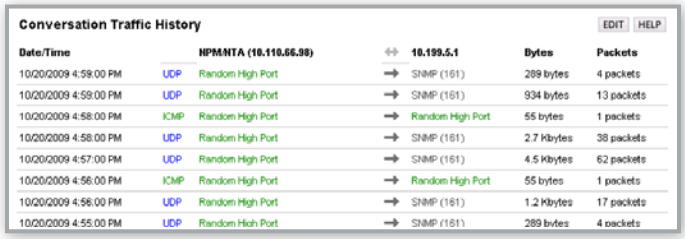
Figure 9.



In the above NTA screen we see two resources, one for the Endpoint Details for the Orion NPM/NTA (10.110.66.98) and one for the Top 25 conversations involving the Orion NPM/NTA endpoint. These are conversations between 10.110.66.98 and other endpoints over the last 15 minutes, as indicated in the figure. Notice the endpoint at 10.199.5.1 is listed in the top 25 conversations. This is the collector's presentation of one of the conversation in the NetFlow cache. Separate flows exported to this collector have been stored and aggregated so we can examine these flows when we need to, rather than only as they happen. This shows us that over the last 15 minutes there has been an aggregated flow conversation containing 16 KB of data and represents 4.67% of the observed traffic.

Below are the individual flow records found by drilling down on the highlighted conversation aggregate from 10.199.5.1 above. (see **Figure 10**).

Figure 10.



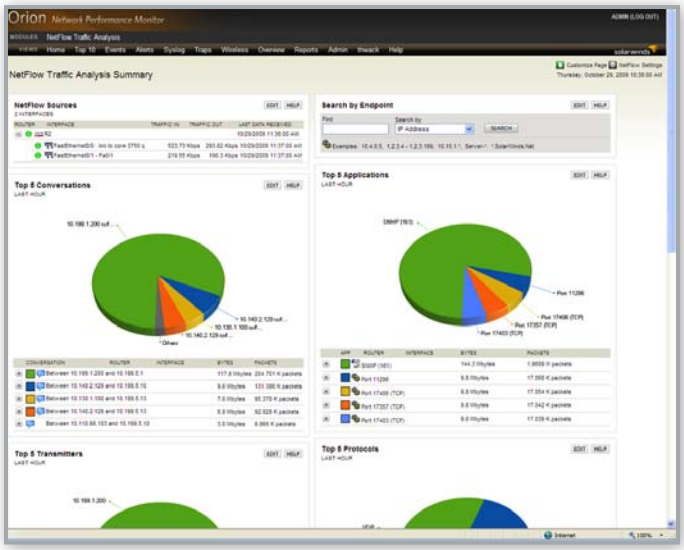
The above screen shots show the flows specifically from the exporter and cache we examined in sections 2 and 3. The collector also has the ability to aggregate flow data to show flow analysis for the whole network, individual nodes and interfaces. Below are examples of each of these.

Summary Level View (see Figure 11)

This shows us a breakdown of the types of traffic seen at the network or summary level. Here there is only one source (exporter) but this level will aggregate data from all sources this collector is listening to. There are more graphs and tables than can be shown on a single screen shot. Other items that can be seen at the summary level include:

- Top N Domains (using DNS and/or NetBIOS resolution of the endpoint IP addresses).
- Top N IP Address Groups
- Top N Receivers
- Top N ToS
- Search by Application
- Collector Status
- NetFlow Events

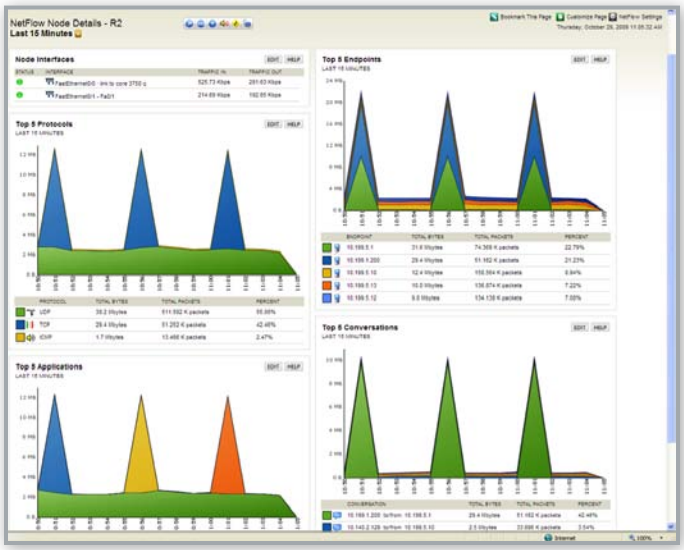
Figure 11. Summary Level View



Node Level View and Interface Level Views (see Figure 12)

Here the same types of resources are available but the data is restricted to all exports from a single node rather than network wide. Similarly, the interface level view shows only flow data from a specific interface.

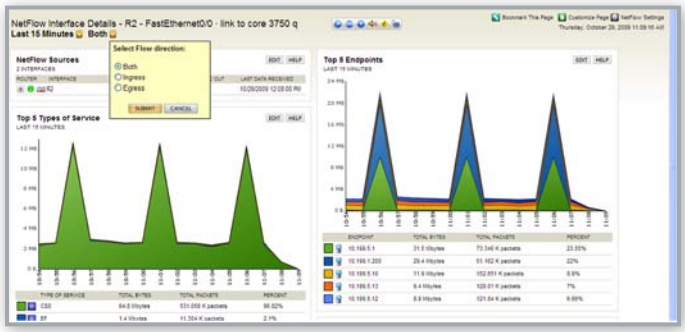
Figure 12. Node Level View and Interface Level Views



Interface View with Flow Direction Options (see Figure 13)

One important difference between node level information and interface level information is that at the interface level we are able to specify the directionality of the data we wish to see, as shown above.

Figure 13. Interface View with Flow Direction Options



Deployment Strategies

NetFlow is a very powerful tool. When it is applied with careful planning, this powerful tool works for you, yielding useful data and requiring a minimal investment. The flip side of that is that if NetFlow is implemented without careful planning it can overwhelm the collector with redundant data and consume valuable resources on the routers and switches. This has the effect of burying the data you need to find under tons of data you probably don't need. In this section, I'll focus on planning deployment and understanding the impact of the choices made in the process.

Deployment Planning and Mapping

The first step in deployment is to get a good understanding of what you want to achieve and then map out what points in the network require flow export to support those goals. It is critical in this phase to examine each goal, determine what questions each goal will answer and prioritize the goals using this information. The result of this process might look something like this: (see **Table 2**).

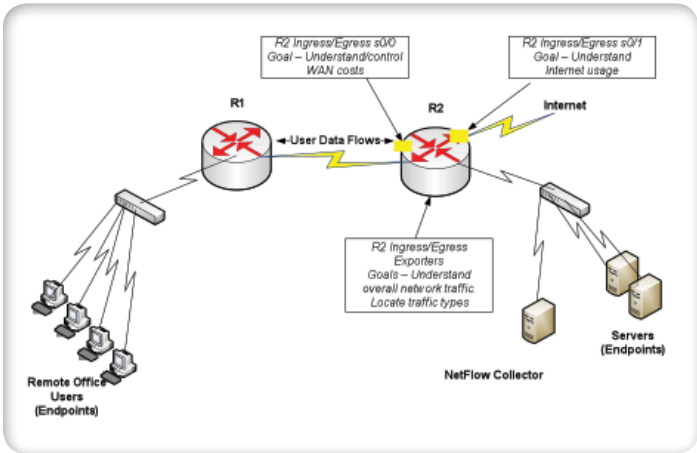
Table 2.

Goal	What question does this answer?	Priority
Understand WAN bandwidth utilization/Control recurring WAN cost.	Why is a circuit experiencing high utilization? How much of the WAN utilization is unnecessary traffic? Is the bandwidth being consumed mostly by a few users? Which applications are consuming the most bandwidth? Are there alternatives to buying more bandwidth?	1
Understand overall network traffic	What is the network being used for? Who uses it the most?	3
Understand Internet usage to control costs, detect attacks and possible legal issues.	Are inappropriate sites being accessed? Who uses the most Internet bandwidth and what are they doing? Are there traffic types that need to be controlled?	2
Locate specific traffic types to detect security issues.	Are security measures working to control access?	3

The goals should speak to business needs as much as possible, not be too technically detailed, and they should not be overly-broad. Goals like "Monitor all traffic on all ports" or "Find the top talkers" are technically possible to achieve but don't speak to a specific business needs. These might also work against other high priority goals such as "Understand overall network traffic" as implementing too many exporters will skew the true picture of the network and burden the collector and the devices exporting flows.

Once the goals are agreed on, the proper placement of exporters to support those goals should be determined. This can best be done using a network map where upon nots can be added to indicate where the goals are being addressed and highlight the locations of exporters. Using our network drawing from section 2 here is what this might look like: (see **Figure 13** on following page).

Figure 13.



This network is very much a simplified version of what you are probably working with, but the basics still apply. I have decided to deploy only two bidirectional exporters — one on the WAN port to R1 and one in the Internet access port. This has the benefit of only requiring configuration changes on R2 so the other network devices will be unaffected by the deployment. Also in this simple network all the data I need to see to achieve my goals can be met with just two exporters. This same idea can be used on a network of any size - it would just require a more in-depth analysis where exporters need to be placed to meet the goals. It would be possible to meet our goals above by exporting from all interfaces on all devices but there are some issues to consider while planning a deployment.

Deployment Planning

NetFlow will add about 4 to 20 percent CPU utilization on the exporting device. The exact amount depends on the active flows in the cache which can only be determined by implementing the exporter. Export of ingress and egress has no significant CPU impact.

NetFlow export traffic will represent about 1.5% to 3% of the total traffic seen by an exporter. If the exporters are added to all interfaces the duplicated export data could become significant, especially if the network aggregates several exporters' traffic into a WAN link. This would be the case if I had decided to collect on all ports in the above network. All the exports from the R1 WAN interface back to the user ports would send to the collector over the single WAN link to R2.

NetFlow uses memory on the exporting device as a function of the flows in the cache.

Implementing flow exporters from many devices requires a lot of time initially and will require occasional troubleshooting and maintenance. This could prove to be very time consuming.

With these points in mind, you should check the system memory and CPU before implementing exporters. Also keep in mind that NetFlow is like counting cars on a highway. You could send five people out to count cars, one each mile for five miles but this could be done by just one person.

All of the above factors should be used to create a deployment plan. A good plan should include a phased implementation with deployment to meet the top priority goals first and add exporters until all goals are met, keeping track on the exporting devices' resources, traffic, and collector resources as you go.

Review

NetFlow consists of:

- The NetFlow cache tracking flows on the device detecting flows
- The NetFlow exporter on the same device
- The NetFlow collector

A v5 flow is defined by seven key fields:

- Source IP address.
- Destination IP addresses.
- Source port number.
- Destination port number
- Layer 3 protocol type
- ToS byte value
- The IfIndex number also called the logical interface number

As the cache receives packets with all seven key fields matching an existing flow record, the cache updates the bytes and packets counts to the existing flow record.

The cache ages flows by tracking:

- TCP flags
- Active and inactive flow timers
- Maximum flow cache size

Once any of the flow aging criteria are met the cache sends the flow information to the exporter and eliminates the record.

The exporter sends flow data to the collector which listens passively for flow data on a specified UDP port.

Exporters can be configured for ingress traffic and/or egress interface traffic.

Collectors collect, store, and present flow data.

Careful consideration of business level goals should be part of the deployment planning. Mass implementation of exporters on all or most interfaces is not recommended.

I hope this has been helpful and I'll see you on thwack!

Related SolarWinds Products

SolarWinds Solutions for NetFlow Monitoring

Now that you've got a solid understanding of NetFlow technology, we wanted to introduce a couple great solutions to help you begin using NetFlow on your network.

Orion NetFlow Traffic Analyzer (Orion NTA)

Orion NetFlow Traffic Analyzer allows you to quantify exactly how your network is being used, by whom, and for what purpose. And with the new CBQoS monitoring you can be sure that the policies you've set give your mission-critical traffic the highest level of priority. Orion NTA's NetFlow monitoring makes it easy to get a comprehensive view of your network traffic, find the bottlenecks, and shut down the bandwidth hogs.

- Quickly and easily identifies which users, applications, and protocols are consuming the most network bandwidth and highlights the IP addresses of the top talkers on the network
- Monitors network traffic by capturing flow data from network devices, including Cisco® NetFlow v5 or v9, Juniper® J-Flow, IPFIX, and sFlow®
- Performs Class-Based Quality of Service (CBQoS) monitoring to ensure that your traffic prioritization policies are effective
- Provides a comprehensive, customizable view of network traffic on a single page
- Enables you to quickly drill into traffic on specific network elements, using multiple views to get the perspective you're looking for
- Generates network traffic reports with just a few clicks
- Monitors Quality of Service (QoS) metrics
- Facilitates investigation of fault, performance, and configuration issues thanks to complete integration with Orion NPM and Orion NCM

Get More Info on Orion Netflow Traffic Analyzer here.

Evaluate Orion Netflow Traffic Analyzer in a live demo environment here.

Download the full functioning 30 day trial of Orion NTA here. http://www.solarwinds.com/products/orion/nta/\7

SolarWinds Real-Time NetFlow Analyzer Free Tool

Real-time NetFlow Analyzer displays inbound and outbound traffic separately for granular analysis that makes problem diagnosis quick and easy. Even better, you can view the historical NetFlow data broken out by application, conversation, domain, endpoint, and protocol. That way you know exactly how your bandwidth is being used and by whom.

Put an end to complaints about the network being slow! With Real-time NetFlow Analyzer, you'll get the visibility into NetFlow data that you've been waiting for. This free tool will allow you to:

- Investigate, troubleshoot, and quickly remediate network slowdowns
- Easily identify which users, devices, and applications are consuming the most bandwidth
- Isolate inbound and outbound traffic by conversation, application, domain, endpoint, and protocol
- Personalize NetFlow data displays to view traffic by specified time periods (up to 60 minutes) and by traffic type
- Customize refresh rates and display units for NetFlow traffic
- SolarWinds Real-time NetFlow Analyzer supports NetFlow Version 5 and records up to 60 minutes of NetFlow data.

About SolarWinds

SolarWinds is rewriting the rules for how companies manage their networks. Guided by a global community of network engineers, SolarWinds develops simple and powerful software for managing networks, small or large. Our company culture is defined by passion for innovation and a philosophy that network management can be simplified for every environment.

SolarWinds products are used by more than one million network engineers to manage IT environments ranging from ten to tens of thousands of network devices. Comprised of fault and performance management products, configuration and compliance products, and tools for engineers, the SolarWinds product family is trusted by organizations around the globe to design, build, maintain, and troubleshoot complex network environments.

SolarWinds is headquartered in Austin, Texas, with sales and product development offices around the world. Join our online community of experts at thwack.com!

About the Author

Andy McBride is a Technical Specialist for SolarWinds focusing on making knowledge of networking and network management accessible to customers and prospects of all levels. The "Networking Fundamentals" series is specifically written for an audience with limited prior exposure to these technologies. Andy's technical background includes seven years at International Network Services (INS) as a Network Engineer and Managing Consultant, three years as a Novell Certified Instructor and five years as a Network Performance Products Manager with BT-Infonet. Prior to entering technology, Andy worked in aerospace on projects such as the SR-71, F-117, F-22, L-1011, F-18 and the space shuttle main engine. Andy has a degree in Chemistry but was wise enough to never work in that field. Andy invites you to follow him on Twitter, @McBrideA, and can be contacted on Thwack, McBrideA.