Perils of the Network

A Series on Network Idiosyncrasies and Degradations

Traffic and Everything Like Traffic:
Dealing with Network Performance Degradation

First published • April 2003

Canadian Head Office • The Hudson House • 400 - 321 Water Street • Vancouver, British Columbia, Canada, V6B 1B8 • http://www.jaalaM.com
United States Head Office • Suite 400 • 2401 - 4th Avenue • Seattle, Washington, USA, 98121 • http://www.jaalaM.com

Company

jaalaM Technologies is a leading innovator of network intelligence software. jaalaM's technology, appareNet, a network intelligence system, operates non-intrusively on live networks, to and from any location worldwide. Without requiring specialized hardware or remote agents, appareNet views the network from the application's perspective. In doing so, appareNet rapidly identifies the locations and causes of network bottlenecks anywhere in the world so that companies can boost the performance of, and gain more value from, the network infrastructure they already have. jaalaM improves its customers' businesses by helping organizations reduce operational costs, increase IP availability, and protect revenues.

Contact Information

Canadian Head Office

The Hudson House
400 - 321 Water Street
Vancouver, BC
Canada V6B 1B8

Sales: 1.800.508.5233
Support: 1.800.664.4401
Telephone: 604.433.2333
Fax: 604.433.2311

http://www.jaalam.com
marketing@jaalam.com

United States Head Office

Suite 400
2401 – 4th Avenue
Seattle, WA
98121

Sales: 1.800.508.5233
Support: 1.800.664.4401

http://www.jaalam.com
marketing@jaalam.com

Technical Series

This is the fourth in a technical series of white papers from jaalaM Technologies examining the Perils of the Network. This series explains network idiosyncrasies and degradations and how appareNet is capable of identifying these network problems, leading to their resolution.

Executive Summary

On a typical IP network, traffic is anticipated as part of the design. Packets bound end-to-end from one host to another are expected to encounter traffic in the form of other packets sent between other hosts. Competing traffic impedes the exchange of packets, introducing delays and an overall reduction in available bandwidth. Therefore, traffic represents a form of network performance degradation, although reasonable levels of traffic are normal and tolerable. In cases where traffic conditions achieve a level of congestion, network paths may no longer function reasonably and the situation may demand some attention. Extreme cases of traffic congestion can result in packet loss. In other cases, conditions may exist that introduce **traffic-like effects, even if no traffic is present**, degrading performance needlessly. These conditions can be identified and remedied.

The challenge in troubleshooting your network is to detect traffic-like behaviours, distinguish them from real traffic and determine what is reasonable and what performance degradation can be avoided.

If you were to treat each cause of performance degradation as a unique kind of problem, optimizing your network's performance becomes a near impossible task. Dealing uniformly with traffic, and everything that looks like traffic, makes it possible to locate and identify issues quickly. In the end, this practice will greatly simplify the challenging job of troubleshooting your network's performance.

appareNet offers unique capabilities for rapidly measuring network capacity and diagnosing network problems – whether they be actual traffic or other problems causing network performance degradation. Once the problem source is isolated, a simple reconfiguration of the network equipment is all that is required to achieve significant and noticeable improvements in IP network performance.

Not all traffic is equal

Reasonable levels of traffic are normal and tolerable. In cases where traffic conditions achieve a level of congestion, packet loss may result. Eventually congestion reaches a level where the network may no longer function adequately.

Fortunately, many of these conditions can be identified, although with some difficulty, and then easily remedied. The result is the recovery of stranded bandwidth taken up by the "ghost traffic". The challenge is to discern when the level of traffic is reasonable and the corresponding performance degradation is inevitable, and when steps can, and should, be taken to regain the full value of the network.

The Apparent Network

The **apparent network is defined as the viewpoint from which the application sees the network**. This isn't necessarily the network conceived of by most Network Engineers or Application Programmers. But it is the most important view to consider when answering the question "**Is it the network or the application?**". This is the view that appareNet was designed to deliver.

There are many APIs available to application programmers. Generally, for simplicity, a programmer will choose the highest level API available to them and rely on operating system components to deliver messages through lower layers. On the other hand, network engineers like to deal with the physical devices and their behaviors at lower layers: routers, switches and WAN links. The appareNet approach synthesizes a compromise between these two opposing views of the network. Let us use the OSI model as a reference and find a demarcation point for the apparent network.

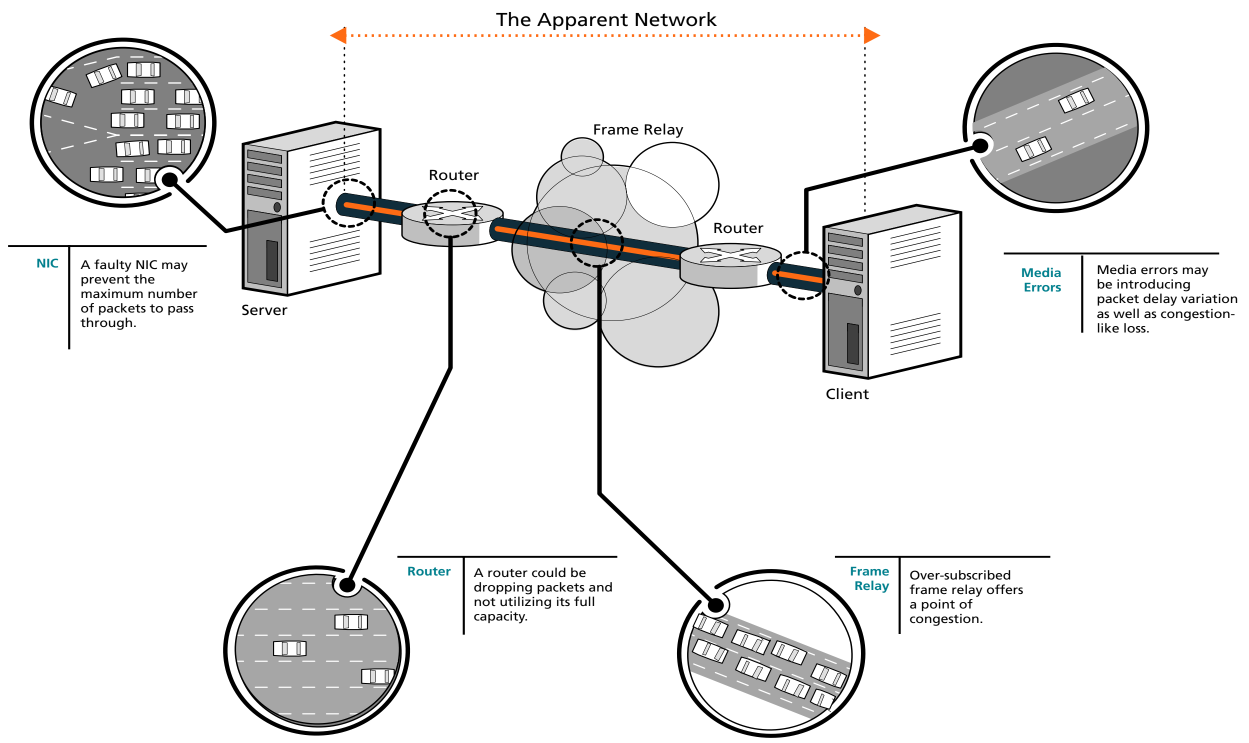
OSI Layer	OSI Description	API Examples	
7	Application	Web App	Application susceptible to delays from all other layers
6	Presentation	HTTP	HTML encode/decode overhead
5	Session	SSL	Certificate exchange, encryption/ decryption overhead
4	Transport	TCP	Connection, Tx (slow start) and Rx window delays
3	Network	IP	Reasonable network measurement point
2	Data Link	802.3	Link protocol dependant
1	Physical	NIC	Hardware API (available to manufacturers only)

By examining the above diagram, it becomes clear that making Layer 6 the demarcation point for our apparent network would not be an effective approach. Delays and overheads created by an inefficient script may produce numbers that are irrelevant to a network engineer. For example, if a web page takes between two and six seconds to load, a network engineer is unable to discern whether or not this is an indication that the network needs tuning. Likewise, telling programmers that a 100Mbps NIC is only being utilized at 15% will not tell them if the application needs further optimization.

For appareNet, we have identified Layer 3 (IP) as our demarcation point for the apparent network. It is the lowest common layer that the end-to-end path is defined at. appareNet samples, models and reports what the Layer 1 and Layer 2 protocols make available to Layer 3. This isolates influences at higher layers. By defining the network boundary as the IP layer, we can determine with confidence whether a problem is network- or application-based.

The end-to-end network path

Sampling at the IP Layer, gives appareNet access to the entire end-to-end network path. This end-to-end view is defined by a host machine at either end, connected together by a continuous Layer 3 path that may include many other physical devices and media. This is the same view that any application will see: it starts at the operating system's API to the network; goes down through the network driver to the NIC; travels out onto the media; passes through switches, routers, and gateways (each with their own NICs, drivers and cables); and finally reaches the destination host where it goes back up through the NIC, driver and operating system to the target application.


The Apparent Network
Frame Relay
Router
Router
Server
Client
NIC
A faulty NIC may prevent the maximum number of packets to pass through.
Media Errors
Media errors may be introducing packet delay variation as well as congestion-like loss.
Router
A router could be dropping packets and not utilizing its full capacity.
Frame Relay
Over-subscribed frame relay offers a point of congestion.

Each aspect of this path can potentially be a limiting influence on its performance. For example, a poorly written NIC driver can keep packets from being sent out as quickly as Layer 1 might allow. Or, a badly constructed cable can make it vulnerable to electromagnetic interference, causing packet delay or loss.

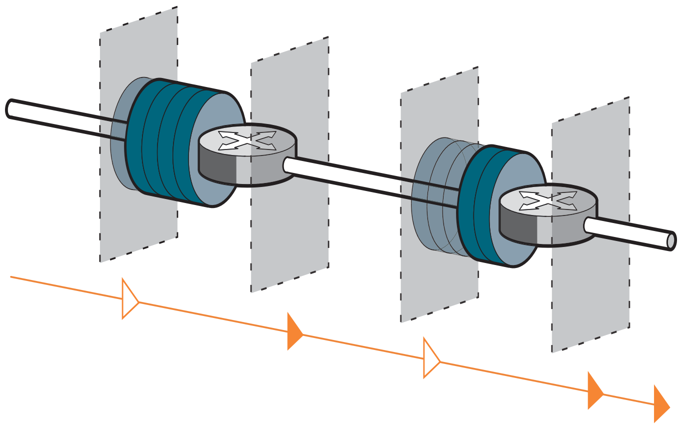
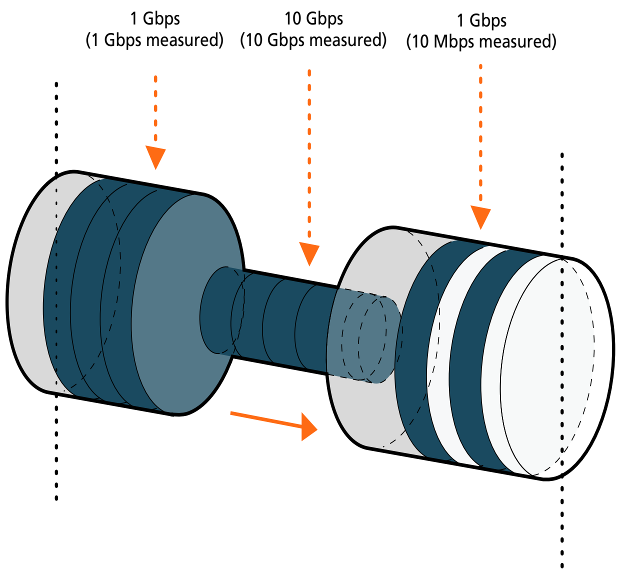


Figure 1 Packets arriving at a router often find other packets ahead of them in the queue. Each packet must wait until all packets ahead have been processed, before being sent along to its destination.

The most notable element of the end-to-end path is the typical router. It is a store-and-forward device that accepts packets into a queue as it processes them first-in/ first-out (FIFO). It is typically modeled as a processing engine and an input queue as shown in Figure 1. Packets arriving at a router often find other packets ahead of them in the queue. Each packet must wait until all packets ahead have been processed, before being sent along to its destination. The length of the wait is proportional to how many packets are already in the queue.

When there is a significant amount of competing traffic, packets traversing from end to end may experience large variations in transit time as they encounter nearly full queues. Since the queues are of a finite size, it is also possible that an arriving packet may find a router's queue completely full. In this case, a packet is discarded and lost.

The rate at which the engine can process packets defines the bandwidth of the path at that point. The bottleneck bandwidth is the smallest rate of transfer on the path – it defines the maximum achievable bandwidth possible from end to end. In Figure 2, the second hop has the smallest capacity and limits the rate of data transfer from end to end.



Packet Behavior through Bottlenecks | Once a bottleneck is encountered, neither appareNet nor any application will "see" the higher-speed downstream links.

Figure 2

The presence of traffic on the path further limits the end to end bandwidth that is available. Maximum achievable bandwidth for that path is shared between all packets on the path. The percentage of bandwidth used by traffic is often referred to as *utilization*. The balance then is bandwidth available for use by other traffic, sometimes referred to as the *available bandwidth*.

For further information on the concept of the Apparent Network, please refer to the first paper, *The Apparent Network* in the white paper series from jaalaM Technologies examining The Perils of the Network.

Traffic

Traffic is a normal part of a network, and networks are constructed to support traffic robustly while meeting certain design specifications. These specifications may vary for different implementations and applications. They might require that packets be delivered to their destinations without loss or corruption; that they arrive within a certain range of delay (time between transmission and reception); or that they arrive in the order sent.

Under normal conditions, traffic affects the end-to-end path by consuming part of the overall capacity of the different devices on the path. For example, a 100Mbps link that is occupied with 30Mbps of traffic (30% utilized) can only offer the unoccupied capacity, or 70Mbps. However, the concept of utilization generally implies constant occupation over some period of time.

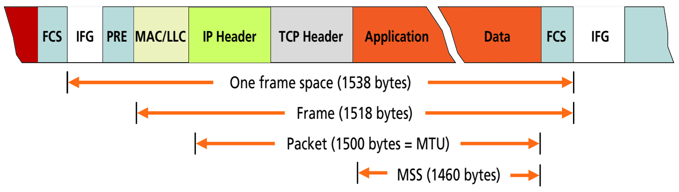
Traffic is rarely constant. Depending on the source, traffic can be highly variable. Interest in characterizing traffic has led to various studies and analyses of traffic[1 2]. Mostly notable was the identification of "bursty" behavior[3 4] on the Internet; the level of utilization changes over time very rapidly, varying from nothing to full capacity and back again. In some cases[5], the load may be relatively constant.

In general, the effect is that each packet travels from one end to the other experiencing the path slightly differently. Each store-and-forward device is encountered with greater or fewer packets already in queue, awaiting delivery. So each packet is delayed for different amounts of time at different points in the path.

The effect on the application is that, at any given moment, it can only get packets to the destination at a certain rate. At 100Mbps (measured at Layer 2), sending 1538-byte frames, with no cross-traffic, an application can theoretically manage

100Mbps / (8 bits/byte x 1538 bytes/frame) = 8127 frames/second

where 1538 bytes is the size of a Layer 2 frame with a 1500-byte packet as payload (see Figure 3).



	Description	Layers	Bits	Bytes
IFG	Inter-frame gap	N/A	96	12
PRE	Preamble (clocking)	1	64	8
MAC/LLC	Media Access Control	2	112	14
IP Header		3	160*	20*
TCP Header		4	160	20
App Data	May contain other layers	5,6,7	11680*	1460*
FCS	Frame Check Sequence)	1	32	4

* Typically 20 bytes but optionally up to 60 bytes
** Maximum Segment Size (MSS) assuming IP Header is 20 bytes

OSI Layer	Description
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

Figure 3

When other packets are present from other sources, fewer packets can be sent along a particular end-to-end path, and overall available bandwidth on that path is decreased. Traffic then acts as a variable bottleneck, limiting an application's access to another host. At the packet level, groups of packets are slowed down by packets ahead of them, and are also separated as traffic is inserted in between (see Figure 4).

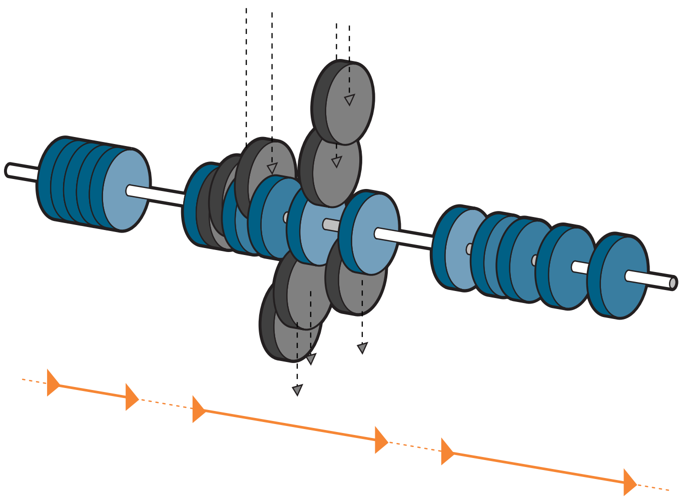


Figure 4 At the packet level, groups of packets are slowed down by packets ahead of them, and are also separated as traffic is inserted between.

Traffic levels can become so high that network performance becomes unacceptably impaired – in other words, there is congestion. Congestion can manifest itself as increasingly long packet delays and eventually, as packet loss. In a normal network, loss is typically a consequence of filled queues – arriving packets have no place to be stored and so packets are deliberately discarded. Since traffic is often bursty, congestion conditions can also be bursty.

Traffic-like problems

Up to this point, we have discussed occurences of "normal" traffic. Now we will review "traffic-like" problems causing performance degradations.

Many problems that lead to performance degradation can be characterized in terms of how they generate traffic-like effects. In some cases, the effects appear at Layer 2 or 3. In other cases, they appear at higher layers. Here are a few examples:

Stuttering NIC

A particular combination of NIC hardware and driver software can result in sub-optimal performance at either end of the network path. Sometimes NICs are not able to transmit packets as quickly as they should. While there are a range of technical reasons why this might happen, it looks the same from a packet perspective – packets sent out are spaced (regularly or irregularly) as if they had encountered other packets in their path.

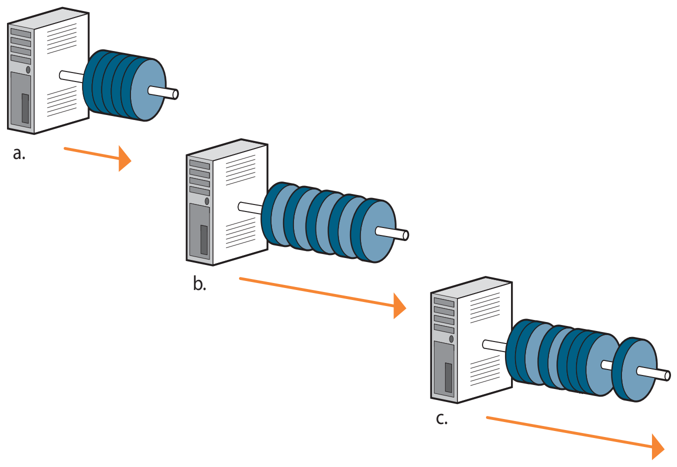


Figure 5 Examples of NICs transmitting: a) tightly packed packets, b) regularly spaced packets, c) irregularly spaced packets.

The effect is that the network appears busy to the transmitting host simply because it cannot transmit packets any more effectively than if it were competing with other traffic sources. Measurements made with appareNet have shown many degraded NICs limited to 60 or 70% of full capacity. Some of the worst instances have included a 100Mbps Fast Ethernet PCMCIA card offering 8 Mbps (8% of capacity) and a 1000Mbps Ethernet PCI card offering 30Mbps (a mere 3% of capacity). These measurements identify why file transfers crawl, data backups take forever, and every network-dependent application using that NIC is running sub-optimally.

Problem Routers

Routers passing packets can suffer a variety of problems that introduce traffic-like behaviors. Normally traffic influences are introduced at mid-path devices where packets on the end-to-end path encounter cross traffic. Typically routers are designed to handle traffic efficiently. The component of a packet's end-to-end delay associated with queue wait and router processing time is referred to as Router Induced Delay (RID).

Overloaded routers, especially those that have been tasked with non-router tasks such as packet filtering, can introduce excessive RID in packet transit time. Filtering is a CPU-intensive task requiring header inspections that routers are not typically designed for. To cope with some situations like DDOS attacks, network administrators will sometimes implement filtering on nodes that were otherwise not intended for the task. In these cases, end-to-end packet delays will often increase dramatically, magnifying the influence of the actual traffic.

Older routers that have particularly slow CPUs are impacted the most. Even without filtering, under conditions where there are numerous ICMP errors generated such as TTL Expiry, Router Redirect and Destination Unreachable, packets passing through that node can accrue additional RID.

Other sorts of problems associated with routers can introduce similar traffic-like affects. These include oversized buffers that hold packets in busy networks for excessively long periods, and media errors such as flawed optical cards or cables.

TCP Congestion Avoidance

For applications utilizing TCP (which would be most applications), relatively insignificant levels of packet loss can invoke TCP *congestion avoidance*. In most popular versions of the TCP stack, a lost packet is assumed to be a consequence of congestion at some point in the network. The transmitting TCP stack reduces the rate of transmission each time a packet is lost (i.e. not acknowledged/ ACK'd by the receiving host), to reduce its load on the network. If no further loss is experienced, the transmitting TCP stack slowly increases its rate until it encounters loss again. This is normal and desirable as it provides applications with the basis for self-limiting in the presence of congestion.

However, under certain circumstances, packet loss resulting from problems other than traffic can cause an application to behave as if traffic is present. Dropping one or two packets relatively infrequently can reduce transfer rates to a crawl, even though there is no traffic and no advantage to be gained by slowing the transfer down. This problem is magnified by large propagation delays found on long links (> 100 ms) where ACKs from the receiving host take longer to return – the longer the ACK takes to return, the longer it takes to speed up again.

So, from a Layer 4 perspective, packet loss from an otherwise low impact problem (such as a cable with poor shielding near an RF noise source) can cause an application to "experience" high levels of traffic. In other words, the TCP stack slows down the transfer in response to traffic that isn't really there. Sending at a higher rate would complete the transfer sooner despite the loss.

This particular problem has a much greater importance than just being a source of performance degradation. Much work has been done to study the limiting nature of current TCP implementations and the related issues[1]. However, this example shows how the conception of traffic extends beyond simply transmitting packets along a network path. In this case, it is a Layer 4 effect that impacts a wide range of applications.

Layer 2 retransmission

Similar to *congestion avoidance*, the mechanism of *Layer 2 retransmission* can result in an effect that resembles traffic. In this case, packet loss that would normally be reported to Layer 3 is obscured at Layer 2. When a particular media is known to cause packet loss (such as 802.11b/WiFi), some Layer 2 interfaces will compensate by negotiating retransmission to avoid the *congestion avoidance* effects.

Consider a WiFi client communicating with an access point. If in sending a packet, that packet is lost (or rather the Layer 2 frame is not acknowledged), then a Layer 2 retransmission takes place before the loss is detected at Layer 3. Each attempt requires additional time and contributes to the apparent delay.

Normally the lack of an ACK would be detected at Layer 3 as packet loss. So when Layer 2 retransmission succeeds, at Layer 3 the time to send a packet seems to be varying but without any apparent loss.

Detecting traffic and problems like traffic

In a perfect network world, every packet is transmitted at the best possible rate as defined by the limiting elements in the path, unobstructed by any other packet. In other words, there is no traffic at all. By design, other packets compete for limited resources, usually in a loosely coupled fashion, through various mechanisms such as congestion avoidance. Packet delay times vary with traffic; sometimes there is packet loss, caused by heavy congestion.

Detecting traffic is easy or hard, depending on how you characterize the traffic and whether you want to discriminate between real traffic and problems that look like traffic. For example, how do you detect the presence of traffic on an end-to-end path?

You can count individual packets as they pass a particular point on the path (i.e. use SNMP to get router counts over certain periods of time). This seems the most obvious definition of traffic and likely closest to what most people associate with the term. Using SNMP also allows you to determine how much packet loss is actually attributable to congestion. This is extremely useful if you have access to a particular router; it's even more useful if you have access to every router on an end-to-end path. If you do, then detecting traffic is "easy".

The problem is that the performance of a network-dependent application is not solely based on this definition of traffic, or those traffic measurements. Applications are also bound by other factors such as the limiting effects described in the previous examples. While they can be treated as if they are very different forms of performance degradation, in the end they affect packets in ways that are typical of traffic – delay, loss, and finally, reduced throughput.

So, armed with an appropriate definition of traffic, and a means to detecting it, a wide range of problems can be rapidly identified and isolated. This greatly simplifies the challenging job of troubleshooting a network, and frees up your time for other things.

Benefits of resolving traffic-like problems

The important distinction between real network traffic and problems that look like network traffic is that, more often that not, you can fix the problems, and that can make all the difference to your operation.

• If a router is overloaded or under stress, detecting the effect can lead to early resolution – especially when you don't have direct access to the router (i.e. it belongs to your ISP).

• If mild packet loss has a critical effect on your TCP-based WAN applications, you can identify and eliminate the source resulting in a dramatic improvement in performance.

• If your WiFi network is under-performing due to environmental conditions, you can modify the configuration, add access points, or compensate for physical obstructions.

If real traffic congestion **is** the problem, you want to be able to determine if it is at acceptable levels, or exceeding the specifications of the network devices in the path. The old answer to every network problem was to **throw bandwidth at it.** Today's answer has to be more prudent and more effective – expensive bandwidth solutions should only be applied to problems that warrant the cost. The rest should simply be identified and fixed.

Utilization by appareNet

When appareNet samples a network path, it carefully notes the effect of the path on the packets. By probing with a variety of types, sizes, and groupings of packets, appareNet quickly builds up a characterization or "view" of the network path. That view produces a wide range of measures and diagnostics, including an analysis specific to competing traffic effects.

appareNet shows the user the effective utilization of the network path. This measure identifies how much of the otherwise available bandwidth is inaccessible to the application. Within the expert system, a deeper analysis separates the true traffic effects from potential problems like poor drivers and non-congestion-based packet loss. The specific hops where the problems originate are isolated and the user is directed to problems that can be resolved, quickly and often very easily.

Conclusion

In the end-to-end view, traffic is more than just packet counts at a router or a gateway. It is an aggregation of effects that begin at the transmitting host, high up in the network stack before packets even reach the NIC, extend out across numerous network devices and media, including the influence of competing traffic, and finally end at the receiving host, again deep within the operating system. Treating each cause of performance degradation as a unique kind of problem makes optimizing your network's performance a near impossible task.

Dealing uniformly with traffic, and everything that looks like traffic, makes it possible to locate and identify issues quickly.

appareNet makes it easy to isolate traffic-related problems and resolve them. It deploys rapidly, without need for remote agents, and runs non-intrusively to almost any IP address. It produces results in real time, showing you what a network is doing **right now**. It provides unparalleled visibility into your networks and the networks of service providers alike.

So, the next time you are confronted with congested networks, treat every problem as if it were traffic. Implement a means to identify and resolve the causes of performance degradation. In the end, this practice will greatly simplify the challenging job of troubleshooting your network.

For further information on appareNet, or to see it live, please contact us at marketing@jaalam.com or toll free at 1 800 508 5233 or visit our website at www.jaalam.com.

Endnotes

1 Brownlee, N., Claffy, K. (2002) Understanding Internet Traffic Streams: Dragonflies and Tortoises. [Online]. Available at: http://www.caida.org/outreach/papers/2002/Dragonflies/cnit.pdf

2 Marron, J.S., Hernandez-Campos, Felix, Smith, F.D. Mice and Elephants Visualization of Internet Traffic [Online] Available at: http://citeseer.nj.nec.com/531734.html

3 Taqqu, Leland M., Willinger, W., Wilson, D. (1994) On the Self-Similar Nature of Ethernet Traffic, IEEE/ACM Transactions on Networking. vol.2, 1-15

4 Cao, J., Cleveland, W., Lin. D. and Sun, D. On the nonstationarity of Internet traffic, Proceedings of the ACM SIGMETRICS, Cambridge, MA, 102-112.

5 Cao, J., Cleveland, W., Lin. D. and Sun, D, (2001) Internet traffic tends to Poisson and independent as the load increases. Technical Report, Bell Labs.

6 Mathis, M., Semke, J., Mahdavi, J., Ott, T. (1997) The Macroscopic Behavior of the Congestion Avoidance Algorithm, Computer Communications Review. volume 27, number 3.