NIPFW – still more

I will be integrating this information into my existing NIPFW document at some time.

NIPFW has been field tested for a month, and also has some new features worth mentioning.

- includes traffic shaping similar to the NAA's Toilet Tank, which can eliminate many types of user bandwidth issues, particularly peer-to-peer.
- can be set to automatically zero counters upon login
- automatic updates can be triggered into active use easily

Traffic Shaping

The campus networks and external connections are limited resources, and we must ensure they are shared fairly by all in the UW community.

The open and cooperative nature of the Internet means users compete for available bandwidth. While we have some ability to classify certain data as more 'important' than others, 'greedy' applications like peer-to-peer software or users downloading DVDs generally have similar or greater[1] priority than important uses like actual research or education. (Most protocols follow established rules like TCP's *slow start* mechanism, but often P2P software

Traffic shaping is the generic name for any attempt to control network traffic flows in order to optimize or guarantee performance, low latency and/or bandwidth. This is normally accomplished with concepts like classification, queue disciplines, enforcing policies, congestion management, quality of service (QoS), etc. and is used to impose some control where there formerly was chaos.

Rate limiting can impose some order for our clients. Two powerful methods have been used internally on campus

IST has used Cisco's CB-WFQ/WRED (Class-Based Weighted Fair Queuing with Weighted Random Early Detection – see http://ist.uwaterloo.ca/cn/shaping.html) for both the villages and academia. It has been particularly effective in the villages, but was discontinued for the rest of the campus.

A different technique was created for the campus wireless space. The Network Authentication Appliances (NAAs) have a capability called Toilet Tank Traffic Shaper, or TTTS. It seems particularly clever because it automatically discourages overuse while having negligible effect on most reasonable client uses.

TTTS works on the same principle of most home toilet tanks. The strategy allows users to enjoy a 'burst' of high bandwidth, but that bandwidth cannot be sustained.

[1] *Most communications protocols inherit TCP's congestion control mechanisms to gracefully reduce their network bandwidth when the network demand is very high. Many P2P applications use UDP and do not incorporate any congestion control. The end result is that they can move more bytes per second than applications which play fair.*

Similar to a toilet, we do refill the displaced quota slowly, so the user will have continuous low bandwidth, or bursty high bandwidth but not both.

In the real world, users may use continuous medium bandwidth for viewing video, or bursty bandwidth as they download then read web pages.

NIPFW offers TTTS-styled capabilities, but further allows the shaping constants to set at the level of ports or protocols. In essence, we can surgically allow many types of desired traffic and simultaneously discourage undesirable traffic patterns.

The shaping is accomplished by defining the capacity of the burst quota and then the refill rate. The burst should be large enough that a user can fully download a large web page or PDF file – eg. 500 MB. The refill rate is expressed in bytes per minute. A refill rate of 500 kB would allow the user to continuously download 500 kB per minute (but no more), or up to 30 MB per hour, or 720 MB per day.

The refill rate actually stops when the available bandwidth reaches the burst quota rate. So leaving your machine over lunch does not give you 60 minutes times 500 kB, the most you ever have available is the predefined quota.

Even setting a refill rate of 1 MB per minute, which should allow most web pages to display, limits the user to a maximum of 1.4 MB/day, which would fall below the campus radar of over-using computers.

The TTTS strategy may seem to interfere with users who wish to download academic material, like the latest Linux CDs. However, they should instead look to one of the campus mirrors, then their speeds will be much faster, and the campus external bandwidth is used more efficiently.

Using TTTS

NIPFW's TTTS is best defined in the workgroup OU, because different settings may be better for student labs than for professor's computers.

Using NIPFW's NEXUS command, the values are easily set by creating a file called c:\nexus\nipfw\workgroup\allow.bandwidth.conf containing text like:

```

# don't restrict uw at all
add 110 allow tcp from uw to uw established

# let me always contact any willing UW device
add 120 allow tcp from me to uw setup


# limit offcampus access to a certain rates
#
# we can say "from any to any", but here we are dividing
# traffic by in versus out.  Clients should not be
# uploading GB's to the world!
#
```

```
# Downloads:
#
# tcp is often bursty, so here we allow up to 1.5 MB at a time,
# and 1 MB/min which leads to maximums of 60 MB/hr and max 1.4 GB/day
#
add 130 allow tcp from not uw to me quota 1500000 rate 1000000

# next we restrict outbound world traffic, so people don't run
# their own high bandwidth servers
#
add 140 allow ip from me to not uw quota 500000 rate 100000

# udp is often used for video and peer-to-peer, allow low
# bandwidth downloads
add 150 allow udp from not uw to me quota 1000000 rate 100000
```

Running `NIPFW NEXUS` adds the rules to the running system, `NIPFW ZERO` can be used to reset the counters while maintaining the rules.

Login Characteristics

(NOT INCLUDED IN AVAILABLE BINARIES TODAY)
~~At login time, Nexus can automatically zero the counters associated with TTTS, but leave the other counters intact. This means future users are not penalized by their predecessors.~~

Automatic Installation and Upgrades

(NOT INCLUDED IN AVAILABLE BINARIES TODAY)
~~A new Windows system service named "NEXUS" is being added to connect various system features.~~

~~Among its tasks, this service automatically starts NIP_FW.SYS only if any NIPFW rule files are defined in either the c:\nexus\localhost or c:\nexus\workgroup subdirectories. If such rules are not found, the firewall is not started. This exact behaviour may be changed in the future if desired.~~

~~The nexus service also updates the NIP_FW configuration if the c:\nexus\nipfw\update.ok file is created. GPOs which install rule files should create the update.ok file as the last thing they do. Its contents are meaningless, but it will be quickly noticed and acted upon.~~

Updated: October 12, 2005
Erick Engelke