Directing Traffic: Managing Internet Bandwidth Fairly

Using open source software to build a traffic management system gave our campus network neutrality, low costs, and low maintenance for managing bandwidth

By **Thomas A. Paine** and **Tyler J. Griggs**

Educational institutions today face budgetary restraints and scarce resources, complicating the decision of how to allot bandwidth for campus network users. Additionally, campus concerns over peer-to-peer networking (specifically outbound Internet traffic) have increased because of bandwidth and copyright issues. The University of Wisconsin–Eau Claire employs an in-house bandwidth management system to administer Internet traffic. Opting for network neutrality over content management, we chose a low-cost, low-maintenance, impartial system to manage bandwidth.

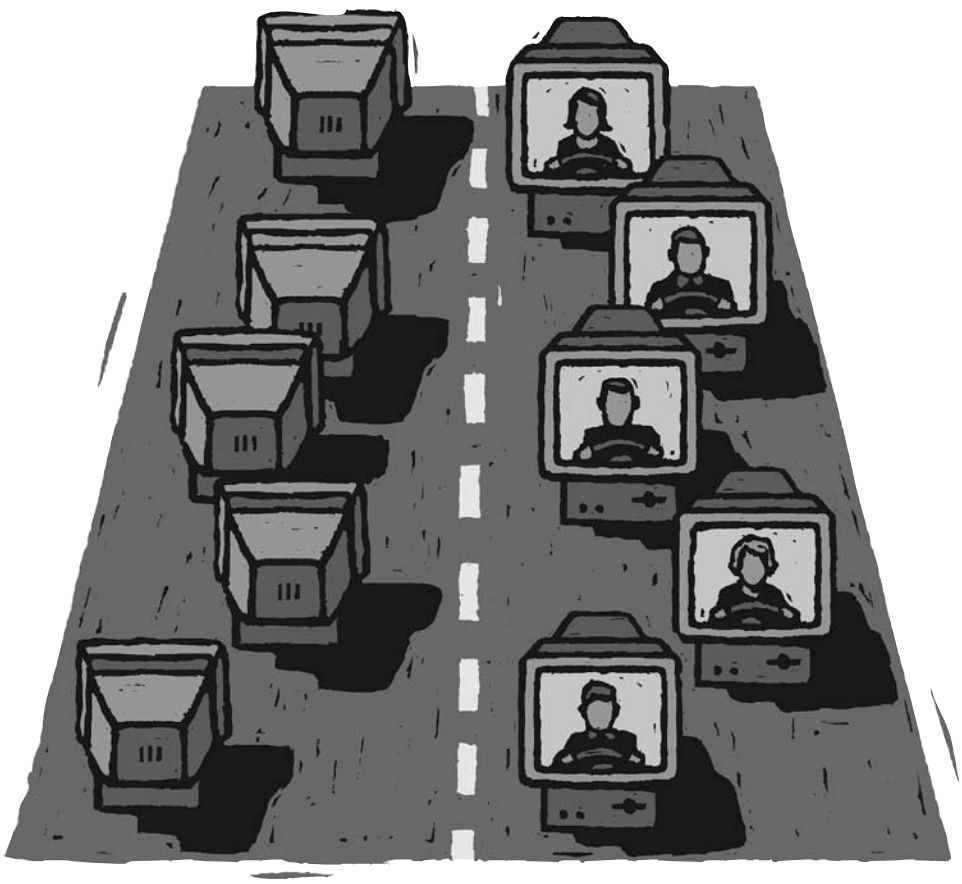
Our Environment

There are 10,000 students and 1,500 faculty and staff on the UW–Eau Claire campus. About 3,500 students live on campus, a majority of whom have at least one registered network device (such as a personal computer, laptop, router, or gaming console). Student-owned computers, along with approximately 4,000 additional devices in the form of office and lab computers and servers, put a heavy demand on bandwidth.

Altogether, the campus has approximately 100 Mbps of Internet bandwidth purchased from two ISPs. The bandwidth purchased from the providers is not equal, so bandwidth management is performed on each of the provider links independently.

The sheer number of devices and users creates a diverse and unpredictable demand for bandwidth and makes bandwidth a high-profile asset. Bandwidth can become prohibitively expensive to expand, and the diversity of campus Internet usage can make it difficult to justify the cost. Simply expanding capacity can also lock the campus into a long-term commitment.

To get the best rates from an ISP typically requires entering into long-term agreements (often several years). As in any business, those costs must be recovered internally from different budget areas through such methods as setting up global accounts or instituting departmental charge-backs. Like many techni-



cal resources, the Internet is something everyone needs but some are unwilling to pay for.

More times than not we need to get the most from resources we already have in place. Changing service levels and tinkering with contracts can spur budget debates, and such solutions are not timely enough to address technical needs when bandwidth contention arises, which can happen at any time.

Beginnings: First Generation

The current UW–Eau Claire bandwidth management system was created to ration the finite bandwidth space allotted to campus users. The system benefits from advances made over three generations of bandwidth monitoring and analysis systems.

The first generation (2000) looked at core routers, which sent out NetFlow data as a synopsis of what was going on and collected and stored the information in a database. (See the sidebar for an explanation of NetFlow.) NetFlow measured the direction and magnitude of traffic but never the content. Scripts then ran database queries every 15 minutes to identify which campus machines were consuming the most bandwidth. The results of the queries guided modification of router configurations, which were used to enforce limitations on individual users. Rate limitation was a committed bit rate applied to an access list (first come, first serve, with a maximum rate). Like a garden hose, it let data out at a certain rate without regard to what else was waiting in line or how imbalanced the backlog became. At the time, this solution worked well, but as campus bandwidth needs grew, the system needed amendments.

Second Generation

The second generation (2003) bandwidth management system needed to provide more dynamic and real-time performance. These improvements were developed in-house.

All UW–Eau Claire network traffic passed through an enhanced filter that, again, identified the machines consuming the most bandwidth. Those machines were queued according to their place on the list (a simple round-robin queue). Snapshots were taken every few seconds rather than every 15 minutes. This second iteration worked well for a few years, but could not scale to the needed packet rates.

NetFlow

NetFlow is a protocol introduced by Cisco that enables network administrators to collect information about network traffic in a summary form. As packets (data) flow through NetFlow-enabled routers and switches on the network, NetFlow can track which machines are talking and how, when, and how much they are talking. This information can be exported for collection, reporting, accounting, or even forensic work. Like a telephone bill, it shows who called whom, but not what was said (the content).

Third Generation: Traffic Management System

The third generation (2006), our Traffic Management System (TMS), provided a way to compensate for our increased packet-per-second rate. We did not want to change the system's design simply to address performance needs, so we used Click, a software-based modular routing framework, which has the ability to run in kernel space.

Click Software

The Click framework arose from a doctoral thesis and project work at MIT. It focuses on producing software-based network routing and switching solutions. Other uses for Click include researching new network protocols, wireless mesh networks, packet handling, and packet schedulers. As a result, Click has a strong user community and an active mailing list.

Using Click requires skills in programming, compiling, debugging, and testing. With the appropriate skills (or having hired the right talent), you can produce some powerful solutions. Think of Click as a toy block set, but better. If the right block (element) doesn't exist, you can build or modify one to meet your specific needs. Much of Click's power comes from this modularity and the fact that it can be loaded as a kernel module.

The Kernel

The kernel (Linux, in our case) is the operating system software that manages the computer and any other software running on it. Computer applications, file management utilities, e-mail clients, web browsers, and games all depend on a kernel. Running software in kernel space compared to user space means software no longer works on top of the kernel but within it. This approach better harnesses the raw power of the hardware. It's like the difference between driving a powerful sports car and riding in one. In the case of our TMS, we want to drive.

Modern computers have a lot of power that sits idle most of the time. Because our TMS is the only thing driving the computer, we can take advantage of that otherwise unused computing power. By doing so, we can better scale to meet current performance needs.

As long as speed improves in computing hardware, with faster memory, faster CPUs, or even more CPUs, the software has an inherent ability to run faster as well. So if today's appliance starts falling short in performance, we can first look at newer hardware before considering drastic changes to our software solution.

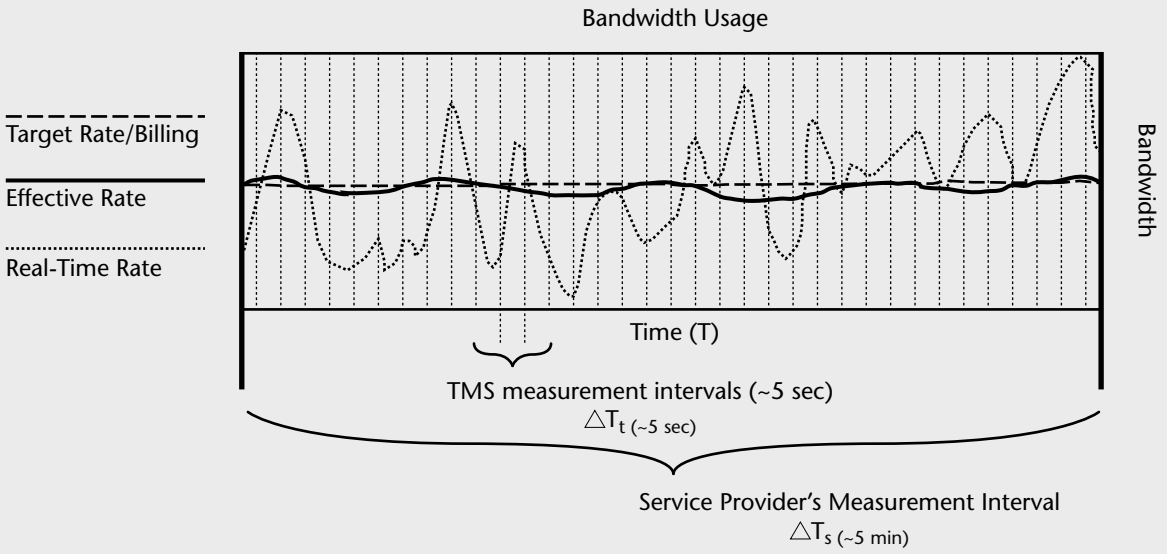
By combining Click, some customized elements (those toy blocks), a modified Linux kernel, and an appliance to run it on, we have a box that can manage bandwidth and still meet our performance demands—without looking at content.

How Our TMS Works

In this iteration of our bandwidth management system, the majority of

Figure 1

TMS Interface Measurements



computers' packets are queued on a first-in, first-out (FIFO) basis, with packets waiting in line for their turn. The system services the FIFO queue first, until it's empty, by using a strict priority scheduler. Other packets, from selected computers, are queued and then scheduled using a weighted deficit round-robin (WDRR) scheduling algorithm. Ultimately, clearing of the system's queues is limited only by the physical line rate of the network interface (the one going to the ISP). The goals of the TMS are to

- allow low-latency/full-line rate bursting (that is, to the extent physically possible given the technical limitations) while producing an effective rate that hits a targeted bandwidth service level, and
- selectively queue and slow long-term individual traffic flows, which would ultimately drive up the long-term average (see Figure 1).

The TMS process assumes, and depends on, the physical line rate being substantially higher than the logical target rate. This is typically the case with sub-rate Ethernet handoffs (for example, a 1 Gbps physical link, with a 50 Mbps service level). It also assumes that not everyone uses the same amount of bandwidth (that is, top-talkers exist).

Typically, Internet service providers (ISPs) measure usage at the 95th percentile, collect data in 5-minute intervals, and bill accordingly. At these measurement points, the service provider takes the interface's 5-minute average data rate. Since 5 minutes of computer time is considerable, the TMS uses much smaller intervals (~5 seconds) to perform usage analysis and to make queuing decisions and system adjustments. It also allows for full-line rate bursting throughout the interval. The process works as follows:

The system is configured with a target rate (if you pay your ISP for 50 Mbps, you set it for 50 Mbps). The system egress (transmit) rate is monitored several times per minute for both inbound and outbound traffic independently (at approximately 5-second intervals).

The system stores data on IP usage during the previous and current intervals. When it determines that the egress rate exceeds the target rate, the interval's data are analyzed. The system assumes that if an IP address was the top talker for this interval, it will be for the next interval as well. It determines which IPs (based on their usage) should be rate-limited in order to hit the target rate. The system then flags those IPs as top talkers. Packets entering the system to or from those IPs (depending whether they are inbound or outbound traffic) are routed through a lower priority queuing path.

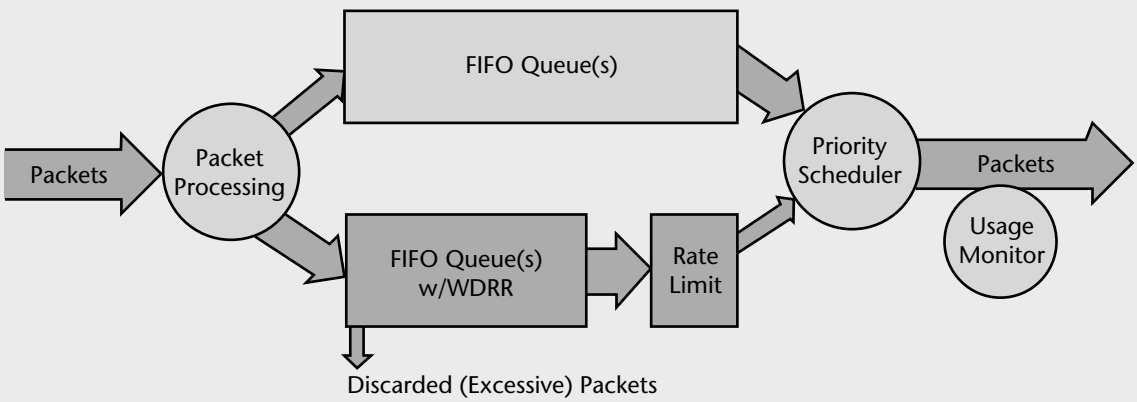
The period in which an IP remains flagged as a top talker depends on its frequency of being flagged. IPs that are repeatedly flagged over close intervals are aged out slower each time they are re-flagged, and they are also weighted differently (given less bandwidth). Each flagged IP address is provided its own FIFO queue. Packets are removed from these queues using WDRR (see Figure 2) only after the FIFO queue servicing the majority of computers' packets is empty.

If the system determines that the egress rate was less than the target rate, nothing is done. The system is also configured for special quality-of-service (QoS) markings and white-listing for devices and services that require low latency (voice/video).

Another way to picture the process is to consider an eight-lane highway in which a few lanes are dedicated to rate-limited traffic. Police officers monitor the number of vehicles on the roadway. When an officer discovers that the number of vehicles would cause too much traffic, future vehicles (coming from the

Figure 2

TMS Packet Flow Overview



same person) are moved to the rate-limited lanes. If too many vehicles are rate-limited, they are removed from the highway. Eventually rate-limited traffic is returned to the general lanes, but for each time those vehicles are moved to the rate-limited lanes again, they stay there longer.

Why We Don't Classify

Many bandwidth management systems classify content or applications. University networks have a unique need, even a requirement, to enable their users. We did not want to be in the position of classifying any one user's application as more important than others. For example, a first-year computer science student could be trying to learn an application he has never used before. Prioritizing applications or users in order to select those that deserve bandwidth can become a matter of discrimination and subjectivity. We have chosen the stance of "network neutrality," which does not prioritize network traffic based on content. Rather, network traffic is queued and transmitted based on bandwidth availability and individual usage trends. Doing this gives all individuals the most bandwidth possible while preventing a few users from hogging all the bandwidth. At the same time, it attempts to fully utilize the bandwidth already purchased.

Additionally, the classifying process requires a great amount of support and time and can rely heavily on vendor updates. Applications are becoming more complex, traffic is becoming less port-specific, and content is becoming harder to classify due to increasing encryption trends. For all of these reasons we chose not to deploy an application-classifying (content) bandwidth management system.

There is, however, one situation in which we classify and prioritize bandwidth: Our campus's non-streaming video demands (such as videoconferencing) are relatively small, but we need to be sure that the TMS does not induce packet loss or latency for real-time video. (Two-way video and audio protocols—real-time protocols—have a very small latency budget and cannot tolerate any packet loss, so this is more a requirement of, than an exception to, our design.)

P2P and the RIAA

Undoubtedly the increase in peer-to-peer (P2P) sharing of files or data has a direct impact on campus Internet usage. But what does that really mean? P2P is technology—a means, not intrinsically a problem. P2P networks are increasingly used for legitimate software distribution. That's because P2P works, and works very well, where a more common client/server paradigm is not feasible. Essentially, it decentralizes distribution. In the open source community (but not limited to it), where users collectively write and contribute to software and solutions, it makes sense to decentralize resources (costs) as well as ideas. Not only can everyone contribute to the building of a project, they can contribute to its distribution.

Are the RIAA's concerns over copyright violations a technology problem? Is there a technology solution? The more we try to control a given technology, the more it seems to evolve. P2P networks used to exhibit well-known behavior that was easy to identify. Once technology solutions were introduced to impede that behavior, P2P became less deterministic, doing things like port-hopping, masquerading, and encrypting data, none of which make future technology solutions any simpler or cheaper.

With over 11,000 users on our campus, we don't want to attempt to classify P2P file-sharing activity. Nevertheless, due to persistent RIAA inquiries about student downloading and file-sharing (we received 12 letters in April 2008, for example), and by request of the Student Senate, starting in January 2008 a more active approach was taken with campus users consuming disproportionate amounts of bandwidth. Through NetFlow data we can identify which users are consuming suspiciously large amounts of bandwidth, which in

Additional Resources

Cisco NetFlow: http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

Click Modular Router Project: http://www.read.cs.ucla.edu/click/

Click Publications: http://www.read.cs.ucla.edu/click/publications

Efficient fair queuing using deficit round robin: http://portal.acm.org/citation.cfm?doid=217382.217453

Linux kernel: http://www.kernel.org/

AxiomTek appliances: http://www.axiomtek.com/

Burrows, Peter, and Olga Kharif, "The FCC, Comcast, and Net Neutrality," *BusinessWeek.com*, April 21, 2008, http://www.businessweek.com/print/technology/content/feb2008/tc20080225_498413.htm

the context suggests P2P file sharing, spam, viruses, compromised computers, or other activities. These users receive e-mail warnings that explain the campus is aware of the anomaly and offer links to more information about the potential issues and lists other helpful resources. E-mails are sent daily until the user's bandwidth usage returns to typical levels. The intent of the e-mail notices is to educate users about the risks involved with some software and to raise awareness about the responsibilities of campus Internet use.

Cost

Another obvious consideration in bandwidth management is cost. The cost of our bandwidth management system is far more economical than application-classifying systems. Including software purchases and ongoing active management and maintenance, the cost of those systems would be tens of thousands of dollars more than UW–Eau Claire's TMS. The time and money we have put into all three generations pales in comparison to expenses for off-the-shelf solutions. Our startup cost was under $4,000 and involved one staff person's time. The money was spent on three appliances: two for production and the other for research and emergency hardware replacement.

What Now?

Very little development has been done to the system after implementing the third generation. In 2006, we stopped finding ways to enhance the TMS. The system has proven itself by the lack of maintenance it requires—it involves few if any staff hours. And, because we aren't managing content, the only feedback received so far has typically been from students with rate-limited machines. They have contacted ResCom (the residence hall computing office) inquiring why their machines were running slow. In those few cases, after looking at the NetFlow data, we could typically explain to the students why they were being rate-limited. Usually the students are instructed on how to turn off file-sharing, remove viruses and malware, and schedule regular updates of their operating systems.

We will continue to observe advancements in the Click framework. Considering the continuing success of our bandwidth management system, no further development or alterations are planned. However, replacing the appliances is inevitable. We have a four-year rotation policy on these types of computers.

Conclusion

Although the low cost of our TMS sounds attractive, cost should not be the first consideration in deciding to implement such a system. The first decision must be whether you want to, or feel you need to, manage content. We know that other campuses within the University of Wisconsin system are blocking (or trying to block) certain content, yet those campuses have also received RIAA notices. We chose a stance of net neutrality and then implemented a low-cost, low-maintenance solution supported by rate-limiting of high-bandwidth IPs and educating users on network hazards. We will continue to look for potential problems, but so far have found our TMS effective and economical in managing bandwidth. *e*

Thomas Paine (paineta@uwec.edu) is a Network Engineer and Tyler Griggs (griggstj@uwec.edu) is a Documentation Specialist in Learning and Technology Services at the University of Wisconsin–Eau Claire.