

# Avoiding Network Capacity Collapse

*John Kristoff*

jtk@depaul.edu

+1 312 362-5878

DePaul University

Chicago, IL 60604

## Capacity Collapse

- Scarcity of capacity
- (Dropped Traffic / Offered Traffic) increases
- *Goodput* decreases (approaches zero)
- Response time increases
- Very little or no real work gets done

Goodput, as opposed to badput refers to the amount of data that is transmitted successfully without being lost or retransmitted.

## Statistical Multiplexing

- A primary advantage of data networks
- Available capacity can be used by anyone
- Share capacity on first in, first out basis
- Build network based on average usage
- But IP is arbitrarily bursty
- Hence, will probably have some congestion
- How do you prevent capacity collapse?

## IP Type of Service Field

*"The Type of Service provides an indication of the abstract parameters of the quality of service desired."* - RFC 791, September 1981

Twenty years later and still no Internet QoS!

The DiffServ Working Group of the IETF has redefined the ToS octet in the IPv4 and IPv6 header. See the DiffServ Working Group homepage for more information.  
<http://www.ietf.org/html.charters/diffserv-charter.html>

## TCP Congestion Avoidance

- TCP cannot *control* congestion
- It can *react* based on implicit network signals
- Assumes packet loss is due to congestion
- TCP is quite good - maybe too good
  - Tries to fully utilize network - it can go very fast
- Want TCP to go slow? Drop packets!
- Dropping packets reduces *goodput*

## ICMP, UDP and Multicast

- Some protocols unresponsive to congestion
- Luckily TCP accounts for ~90% of the traffic
- Congestion *control* is needed
  - A function of the network
- How do we do it? is the question
  - RED and ECN
  - Scheduling and rate limiting
  - Price incentives

At first thought it might make sense to rate limit these protocols, but in doing so you not only run into the usual rate limiting problems (see the accompanying whitepaper to this presentation), but you may also break certain network management applications (e.g. pathchar).

## **What about IPv6, ATM, MPLS...**

- Are these ubiquitous in your network?
- Thought so.
- Probably wouldn't be a panacea anyway
- Next slide please...

## **(D)DoS Attacks are Related**

- But we won't talk specifically about them
- Congestion control ideas may help us
  - With some added features
- Probably only a temporary capacity collapse
- Include this in capacity management plan

The following are references to some of the latest work being done to mitigate (D)DoS attacks:

### Pushback

<http://www.research.att.com/~smb/talks/pushback-dodcert.pdf>

<http://www.aciri.org/floyd/talks/pushback-Nov00.pdf>

### Traceback

<http://www.cs.washington.edu/homes/savage/traceback.shtml>

<http://www.research.att.com/lists/ietf-itrace/>

### CenterTrack

<http://www.nanog.org/mtg-9910/ppt/robert/index.htm>

<http://www.us.uu.net/gfx/projects/security/centertrack.pdf>

## Let's Get More Capacity

- LAN capacity is cheap, we can overprovision
- Leased WAN links can be costly
- Internet service can definitely be costly
- Operational versus capital costs
- Ugh... provisioning problems and lead times
- Need simple, cheap and fast
- We only get to pick one, maybe two if lucky

LAN costs are usually only one time capital outlays. This makes it easy to buy more than enough capacity (e.g. Gigabit Ethernet is cheap!)

WAN links can be expensive, but they are coming down. For example, Ameritech is selling its GigaMAN service, essentially gigabit ethernet (1 Gb/s), for less than it would cost for an OC3c (155 Mb/s).

Internet service however is still very expensive in the typical case. Although some new providers have begun to appear with attractive offers (e.g. <http://www.cogent.com>).

## Access Blocking

- DNS black holing
- IP router filters
- Null routes
- Site blocking

Example Cisco ACL to block access to one [www.napster.com](http://www.napster.com) server from a local Ethernet LAN.

```
int Ethernet0
 ip address 140.192.1.0 255.255.255.0
 ip access-group 101 in

access-list 101 deny ip any host 216.52.135.148
access-list 101 permit ip any any
```

## Rate Limiting

- IP, UDP and TCP based - usually
- IP addresses
- Protocol ports
- Strict limits
- Dynamic limits

Example Cisco CAR 2 Mb/s limit configuration on default Napter TCP port 6699.

```
int Ethernet0
 ip address 140.192.1.0 255.255.255.0
 rate-limit input access-group 101 2000000 64000 128000 \
 conform-action transmit exceed-action drop

access-list 101 permit tcp any any eq 6699
access-list 101 deny ip any any
```

## UIUC Rate Limiting Experiment

- Allow full capacity access by default
- "Out-of-profile" users are rate limited
- Increasingly aggressive limits if necessary
- Analyzing cflowd data to determine usage
- Dynamically upload CAR configs once/hour
- Scaling issues - a tad scary

[http://www.ncne.nlanr.net/training/techs/2001/0128/presentations/2000101-kline1\\_files/v3\\_document.htm](http://www.ncne.nlanr.net/training/techs/2001/0128/presentations/2000101-kline1_files/v3_document.htm)

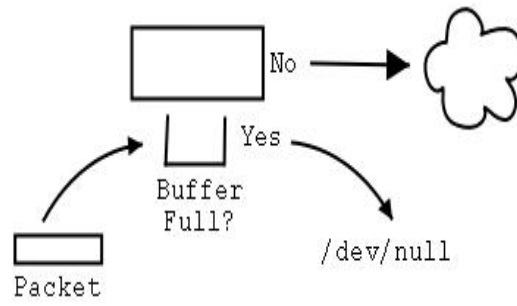
Also see University of Waterloo's technique here:  
<http://ist.uwaterloo.ca/cn/Residence/rn-excess.html>

## Active Queue Management

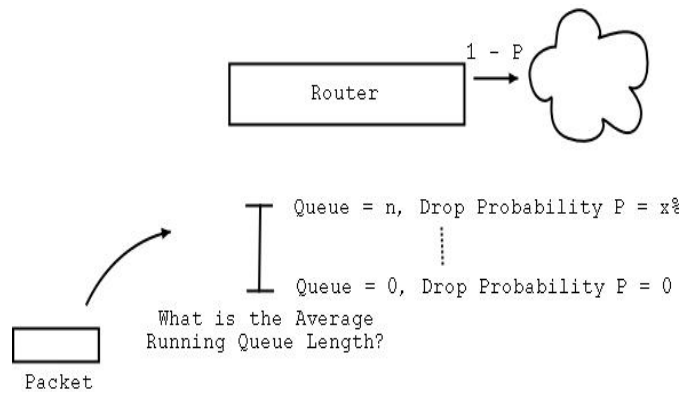
- One way to *control* congestion in the network
- Tail drop (FIFO queueing)
- Random Early Detection (RED)
- Explicit Congestion Notification (ECN)
  - Probably coupled with RED
- Ongoing experimentation and research
- Implementations available

This is what I have kind been playing around with. It have as noticeable affect as some of the other solutions, but it plays nicely in the whole Internet transparent, end-to-end model thing. On our Internet border router, it is only useful for outbound traffic as that is the constrained link and that is the only direction we have any control over. It would be nice if ISPs did more on their side, but typically they are hesitant to install filters, AQM or other mechanisms on behalf of individual customers – unless perhaps in the case of a temporary (D)DoS attack.

## Tail Drop Illustrated



## RED Illustrated



Instead of the average running queue length, you could use the instantaneous queue length.

## Scheduling

- Alter transmission order of packets
- Can be based on:
  - IP Addresses
  - Priority (e.g. ToS bits)
  - Protocols (e.g. SSH)
  - Flow characteristics
- Must define capacity/weight for queues

## Traffic Shaping

- TCP rate control
  - Alter TCP receiver window *on the fly*
- ACK pacing
  - Slow or spread out ACKs to control sender
- Packeteer (middlebox) does this
  - Yes, it can be a little scary
- Can be implemented in end host stacks

## Caching

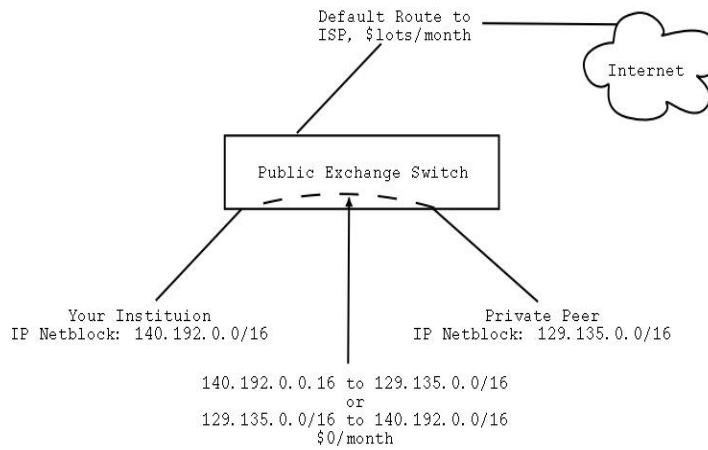
- Transparent
  - It is there, but users don't know it
- Voluntary
  - It is there, but users must know to use it
- Probably only buys a short amount of time

Network caches can cause some problems, particularly when it comes to troubleshooting connectivity problems. For example, I was experimenting with a transparent cache and one user was having a problem getting their Java applet based stock ticker program to work. Everything else seemed to work fine. We (including the vendor) could not track down the specific problem, but it was believed that an upcoming coming release of cache software would fix the problem. It is particularly difficult to troubleshoot in cases like these where network staff would traditionally be able to pull out packet analyzer tools. In this case, the session information looked completely normal. We had no view into the black box that was the cache to determine why or what was failing.

## Private Peering

- There is such a thing as a free lunch!
- Well, OK not really
- Involves some routing complexity
- Startup cost might be high
- You may have a choice of transit provider
- If you can get to an exchange, do so!

# Private Peering Illustrated



## Monitoring

- Some tools:
  - MRTG, RRDTool, cflowd
- Some things to watch:
  - Queue depth
  - Packet drops
  - Link utilization
  - Buffer utilization
- Latency versus throughput

Monitoring queue depth over time would be a good indicator of latency through a router. Unfortunately I haven't found a way to do trending on these values on our Cisco routers like we do with other SNMP statistics (the MIB value does not count up over time, it is an instantaneous value). If anyone has any ideas, please pass them on.

For more on latency versus throughput, see:  
<http://www.stuartchesire.org/rans/Latency.html>

A good place for network monitoring info is:  
<http://www.caida.org>

## Consortia

- Some nice things in your own back yard?
- Might be free, low cost or subsidized
- May at least be lots of capacity
- Might also be worth what you pay
- Examples:
  - Internet2
  - Illinois Century Network
  - STAR TAP

Illinois Centurary network homepage:

<http://www.illinois.net>

Internet2 homepage:

<http://www.internet2.edu>

STAR TAP homepage:

<http://www.startap.net>

## Proxy Servers

- Lots of opportunity for control
- Can do lots of the capacity solutions at once
- Not sure that you want them to
- Lots of *middlebox* issues

There are a number of good documents from the IETF that address the issue of middle box architectures. A good starting point is the MIDCOM working group homepage:

<http://www.ietf.org/html.charters/midcom-charter.html>

Also see RFC 2775, Internet Transparency by Brian Carpenter.

## Content Distribution

- Content providers move data closer to you
- Maybe setup up your own Red Hat mirror
- Akamai is well known in this space
- A form of load balancing

## **Content Subscription**

- Obtain local copies of data for distribution
- Sort of like a library service
- You do not own the content
- May help alleviate copyright issues
- iBEAM is popular in this space

## Network Address Translation (NAT)

- Intended as solution to IP address shortage
- Has a number of well documented problems
- Probably not your capacity solution
- Probably wouldn't help much anyway
- In fact, it would probably hurt you more
- Not recommended if you have addresses
- Bad Juju

## What Would I Do? (Bias Slide)

- Oversubscription before CoS/QoS
- Preserve *End-to-end model*
- Get to an exchange and peer
- Get into a consortium like Internet2
- Do lots of monitoring (understand traffic)
- Be wary of *silicon snake oil*
- Be willing to research and test anything

Oversubscription can be cheaper in the long run.

The end-to-end or transparency model has been argued as one of the most crucial design criteria for the success of the Internet. There are some trade-offs in such an architecture, but there is inherently a lot of power.

Vendors are clamoring for your network dollar. They will be happy to sell you all kinds of boxes (middleboxes) that do all kinds of neat and interesting things. However, the Internet was built and still runs best on the concept of a dumb network and intelligent end hosts.

## References

<http://condor.depaul.edu/~jkristof/>

<http://www.aciri.org/floyd/>

<http://www.ietf.org>

<http://www.nanog.org>

<http://listserv.nd.edu/archives/resnet-l.html>

<http://www.theorygroup.com/Archive/Unisog/>

<http://darkwing.uoregon.edu/~joe/how-to-go-fast.ppt>