CS5412:
NETWORKS AND THE CLOUD

Lecture III
Ken Birman
Cloud computing is transforming the Internet!

- Mix of traffic has changed dramatically
- Demand for networking of all kinds is soaring
- Cloud computing systems want “control” over network routing, want better availability and performance
- ISPs want more efficiency, and also a cut of the action

Early Internet: “Don’t try to be the phone system”

Now: “Be everything”. A universal critical resource

- Like electric power (which increasingly, depends on networked control systems!)
- And the phone system (which now runs over the Internet)
Current Internet loads

Global Consumer Internet Traffic
(Petabyte usage per month)
2010-2015

Source: Cisco

Peak Period Aggregate Traffic Composition
(North America, Fixed Access)

Source: Sandvine's Fall 2010 report on global Internet trends
Looking closer

- As of 2010:
  - 42.7% of all traffic on North American “fixed access” networks was attributable to real-time media
  - Netflix was responsible for 20.6% of peak traffic
  - YouTube was associated with 9.9% of peak traffic
  - iTunes was generating 2.6% of downstream traffic

- By late 2011
  - Absolute data volumes continuing rapid rise
  - Amazon “market share”, and that of others, increasing
Implications of these trends?

- Internet is replacing voice telephony, television... will be the dominant transport technology for everything
- Properties that previously only mattered for telephones will matter for the Internet too
- Quality of routing is emerging as a dominant cost issue
  - If traffic is routed to the “wrong” data center, and must be redirected (or goes further than needed), everyone suffers
  - Complication: Only the cloud knows which route is the “right” or the “best” one!
Continuous operation of routers is key to stream quality and hence to VOIP or VOD quality.

A **high availability** router is one that has redundant components and masks failures, adapts quickly.

2004 U. Michigan study of router availability:
Minor BGP bugs cause big headaches

- In this example, a small ISP in Japan sent 3 minor but incorrect BGP updates
- Certain BGP programs crashed when processing these misreported routes
- Triggers a global wave of incorrect BGP activity that lasts for four hours
- Software patch required to fix issue!
Minor BGP bugs cause big headaches

A typo in a BGP configuration file...

... major consequences!
What is BGP and how does it work?

- Modern routers are
  - Hardware platforms that shunt packets between lines
  - But also computers that run “routing software”
- BGP is one of many common routing protocols
  - Border Gateway Protocol
  - Defined by an IETF standard
- Other common routing protocols include OSPF, IS-IS, and these are just three of a long list
What is BGP and how does it work?

- BGP is implemented by router programs such as the widely popular Quagga routing system, Cisco’s proprietary BGP for their core Internet routers, etc.
- Each implementation...
  - ... follows the basic IETF rules and specifications
  - ... but can extend the BGP protocol by taking advantage of what are called “options”
What is BGP and how does it work?

- Any particular router that hosts BGP:
  - Would need to run some BGP program on one of its nodes (“one” because many routers are clusters)
  - Configure it by telling it which routers are its neighbors (the term “BGP peers” is common)
  - BGP peers advertise routes to one-another
  - For example, “I have a route to 172.23.*.*”
Initially, the 174 network advertises a route to 2497
Routing updates occur within the 174 network.
When the 174 network withdraws its route to 2497, the 6461 network activates a backup route and advertises it.
Notations for IP addresses

- IP addresses are just strings of bits
  - IPv4 uses 32-bit addresses
  - In IPv6 these become 64-bit addresses
  - Otherwise IPv4 and IPv6 are similar

- BGP uses “IP address prefixes”
  - Some string of bits that must match
  - Plus an indication of how many bits are in the match part
  - Common IPv4 notations: 172.23.*.*, or 172.23.0.0/7
  - IPv6 usually shown in hex: 0F.AE.17.31.6D.DD.EA.A0
  - The Cogent slide simply omitted the standard “a.b.c.d” notation, but this is purely a question of preferences
BGP routing table

- Basic idea is that BGP computes a *routing table*
- Loads it into the router, which is often a piece of hardware because line speeds are too fast for any kind of software action
- Router finds the “first match” and forwards packet
In 2004 most routers were a single machine controlling one line-card per peer

In 2012, most core Internet routers are clusters with multiple computers, dual line-cards per peer, dual links per peering relationship

In principle, a 2012 router can “ride out” a failure that would have caused problems in 2004!

But what about BGP?
Suppose our router has many processors but BGP is running on processor A
- After all, BGP is just a program, like Quagga-BGP
- You could have written it yourself!

Now we need BGP to move to processor B
- Perhaps A crashes
- Perhaps we’re installing a patch to BGP
- Or we might be doing routine hardware maintenance
Remote peers connect over TCP

- BGP talks to other BGPs over TCP connections
  - So we had a connection from, say, London to New York and it was a TCP connection from X to A.
  - Now we want it to be a connection from X to B.
- BGP doesn’t have any kind of “migration” feature in its protocols hence this is a disruptive event
  - BGP will terminate on A, or crash
  - BGP’ starts running on B
  - Makes connection to X. Old connection “breaks”
How BGP handles broken connections

- If BGP in New York is seen to have crashed, BGP in London assumes the New York router is down!
  - So it switches to other routes “around” New York
  - Perhaps very inefficient. And the change takes a long time to propagate, and could impact the whole Internet
- Later when BGP restarts, this happens again
- So one small event can have a lasting impact!
  - How lasting? Cisco estimated a 3 to 5 minute disruption when we asked them!
What happens in those 3 minutes?

- When BGP “restarts” on node B, London assumes it has no memory at all of the prior routing table.
  - So London sends the entire current routing table, then sends any updates.
  - This happens with all the BGP peers, and there could be many of them.
- Copying these big tables and processing them takes time, which is why the disruption is long.
BGP “graceful restart”

- An IETF protocol that reduces the delay, somewhat
- With this feature, BGP B basically says “I’m on a new node with amnesia, but the hardware router still is using the old routing table.”
  - Same recovery is required, but London continues to route packets via New York. Like a plane on autopilot, the hardware keeps routing
  - However, that routing table will quickly become stale because updates won’t be applied until BGP’ on B has caught up with current state (still takes 3-5 minutes)
High assurance for BGP?

- We need a BGP that is up and in sync again with no visible disruption at all!

Steps to building one

- Replicate the BGP state so that BGP’ on B can recover the state very quickly
  - We’ll do this by replicating data within memory in the nodes of our cluster-style router
  - BGP’ on B loads state from the replicas extremely rapidly

- Splice the new TCP connections from BGP’ on B to peers to the old connections that went to BGP on A
  - They don’t see anything happen at all!
(1) State of BGP replicated within router-cluster nodes

(2) Failure causes BGP to migrate

(3) Reload state from replicas

(4) Attempt to reconnect to peer intercepted, spliced to old connection
How does TCP-R work?

- Role of TCPR is to
  - Detect an attempt to reconnect to the same peer
  - Connect the new TCP endpoint on node B to the old TCP session that was active between London and node A!
  - Can this be done? Can BGP operate over the resulting half-old, half-new connection?

- Need to understand how TCP works to answer these questions
TCP protocol in action

- TCP has a pair of “windows” within which it sends data “segments” numbered by byte offsets.
- Varies window size to match data rate network and receiver can handle.
TCP windows are like a pair of bounded buffers
Sequence numbers established in initial handshake

- Connection creator (say, A) says to B:
  - I want to make a connection to you using initial sequence number $A \rightarrow B$ 1234 (a random number)
  - B replies I will accept your connection using initial sequence number from $B \rightarrow A$ 9171 (also random)
  - A responds “our connection is established”

- Notice that both numbers start at random values
- This protects against confusion if msg redelivered
- Called a “three-way handshake”
Sequence numbers established in initial handshake
Basic TCP-R idea

- TCP-R just notes the old sequence pair
  - When BGP B tries to connect to the old peer, TCPR intercepts the handshake and runs it “locally”, noting the delta between old and new sequence numbers
  - Now on each packet, TCPR can “translate” from new numbering to old and back, fooling the old TCP stack into accepting the new packets
  - Updates the TCP checksum field on packet headers

- This splices the connections together
FT-BGP

- FT-BGP has a bit more work to do
  - Old BGP just accepted updates and processed them
  - FT-BGP must log any updates it sends or receives before TCP acknowledges the incoming update, or sends the outgoing one
  - FT-BGP must also complete any receive or send that was disrupted by the failover from node A to B

- But these are easy to do

- Total time for failover: milliseconds!

CS5412 Spring 2014 (Cloud Computing: Birman)
Thus we’ve made our router more available

- Goal was to improve on the 2004 situation:
  - Replicated links and line cards
  - FT-BGP for failover
  - Better management tools to reduce risk of misconfiguration

... every element of the picture has been “fixed”!

- Other Causes: 9%
- Router Misconfiguration: 36%
- IP Routing Failures: 32%
- Physical Link Failures: 23%

Source: University of Michigan and Sprint, October 2004
How available can the network be?

- Today’s Internet achieves between 2 and 3 “nines” of availability
  - Means that over a period of X seconds, would expect to see between 99% and 99.9% of “good behavior”
  - Between 1% and 0.1% of time, something is seriously wrong
- Hubble project at UW: finds that on a national scale Internet has large numbers of black holes, slow patches, terrible choices of routes, etc at all times
- With work like what we’ve seen could probably push towards a “5-nines” Internet, comparable to voice telephony but at Internet data rates
Could we go further?

- Same idea can harden other routing protocols
- But what about other kinds of router problems?
  - For example, “distributed denial of service attacks” that overload links with garbage data or overwhelm a web site with junk packets?
- Also, how could cloud providers “customize” routing?
  - Cloud operators want a degree of routing control
  - Ideally would want to look inside the packets
These are active research topics...

- Ideas include:
  - Better control over routing within entire regions
  - Some way to support end-to-end “circuits” with pre-authentication between sender and receiver
  - New routing ideas aimed at better support for media streams
  - Monitoring BGP to notice if something very wrong occurs

- Leads to the vision of a collection of “SuperNets” each specialized in different ways, but sharing routers
SuperNet examples

- Google might want to build a Google+ net optimized for its social networking applications.
- Netflix would imagine a NetFlixNet ideally tuned for transport of media data.
- The smart power grid might want a “grid net” that has security and other assurance features, for use in monitoring the power grid and controlling it.
Sharing resources

- The idea is very much like sharing a machine using virtual machines!
  - With VMs user thinks she “owns” the machine but in reality one computer might host many VMs
  - With SuperNet idea, Google thinks it “owns” the GoogleNet but the routers actually “host” many nets
- Could definitely be done today
  - Probably would use the OpenFlow standards to define behaviors of these SuperNets.
Can we “secure” the Internet?

- End-to-end route path security would help...

- ... but if routers are just clusters of computers, must still worry about attacks that deliberately disrupt the router itself
  - Like a virus or worm but one that infects routers!
  - This is a genuine risk today
  - Must also worry about disruption of BGP, or the DNS or other critical services
A secured router

- We would need a way to know precisely what we’re running on it
  - Can be done using “trusted platform modules” (TPM is a kind of hardware repository for security keys)
  - Would need to run trustworthy code (use best development techniques, theorem provers)
  - Then “model check” by monitoring behavior against model of what code does and rules for how network operates

- Entails a way of securely replicating those control rules, but this is a topic we’ll “solve” later in the course
A secured network

NOC, this is the network topology I want you to use.

Central command controls routing for a region, and sets the policy for BGP updates.

Guards supervise router communication but can’t create fake router packets: Lack signature authority (TPM keys).

A monitored router can only behave in ways the policy permits.

Use a hardware-security feature called the TPM to offer hardened virtual machines.

Safe router in a box.
Hosting a SuperNet on a SecureNet

- Secure net is an infrastructure on which the SuperNet runs with no means to disrupt other users!
- SuperNet controls its own virtual resources (maybe even dedicated links)

SuperNet “in a box” benefits from a non-disruptable network

Trusted network

Netflix Movie

Service

CS5412 Spring 2014 (Cloud Computing: Birman)
Conclusions?

- Cloud is encouraging rapid evolution of the Internet

- Different cloud “use cases” will want to customize routing and security in different ways

- Nobody wants to be disrupted by other users or by hackers, and this is a big issue for cloud providers

- Tomorrow’s network will probably have features that allow each provider to create its own super-net specialized in just the ways it wishes. They will share physical infrastructure.