Netflix Open Connect
Not a trick: From 0 to 2 Terabits in 12 hours
“It’s not an illusion. It’s Netflix.”

AD 2013

NETFLIX
For years we heard...

- ISPs wanted to work directly with us to manage their traffic
- By working with us, ISPs would be able to accurately forecast their traffic volumes and budget accordingly
- We could help ISPs reduce their cost of delivery
- We needed to put some skin in the game
- There’s always money in the banana stand...
“Streaming on Netflix? I don’t understand the question and I won’t respond to it.”
So, we built a (test) CDN...

I swear, we used velcro later..
Found the velcro...
Then, in production...

- We built a CDN based on the premise that we would roll out clusters of 40 machines and two routers
  - Each machine has 2x10G (capable of 15G at peak)
  - A Juniper MX480 or MX960 with 16x10G cards fit this nicely (with room to grow)
Then, we decided we could make things more efficient…

- Design a low profile, 100% flash system to host the most popular content
- Because we know what users want to watch in advance we can be highly efficient in pre-positioning content
- Most of our colos share a similarly designed footprint
  - 5-7 racks
  - ~5kW of power per cabinet (208V/30A pri + red)
Enter SSD-based Open Connect Appliances.

In sites with 1+ Tbps of Netflix traffic at peak:

- 14 TB per 1U system
  - Commodity SSD (< 60c/GB, Micron m500)
  - 1 TB in 2.5" form factor
- 3x 10 Gbps SFP+ NIC
  - 4th left unused due to bus limitations
    - Except on Juniper installations to manage oversubscription
- Total system power 125W per 1U
- Software stack (same as spinning disk systems, which these complement)
  - FreeBSD / nginx / bird / Netflix application code
This drastically increased our port count

- 20 servers @ 2 ports each = 40 10G ports
- 30 servers @ 3 ports each = 90 10G ports
- Uplinks out = 130 10G ports

----------------------

260 10G ports

This leaves us with very few choices if we want to keep a single cluster…

(simple = good, reliable, supportable)
Arrested Development…

- Juniper MX960 w/ 16x10G cards = 176 10G ports
  - Nearly 100 ports short
- Build an aggregation layer
  - Might be able to save some uplink ports but downlink we save, at most, 15%. Still not enough.
- Move to Juniper 32 port “Snorkel” cards
  - Requires 12.3 code (bleeding edge)
  - Oversubscribed 3:2 during normal operations
  - Oversubscribed 2:1 during fabric failures
- Move to Cisco ASR9K w/ 36 port Typhoon cards
  - No oversubscription during normal operations
  - Well established code base
I'VE MADE A HUGE MISTAKE
Not really…

- We converted two of our major POPs to Juniper “Snorkel” cards
- Our smaller POPs still remain Juniper
- Our other major POPs were converted to ASR9K, however…
Caveats…

- Converting from JunOS to XR isn’t *that* painful
  - No equivalent functionality for Configuration Groups (yet-coming in 5.0) but there is inheritance
  - No commit scripts
    - There are op scripts, just have to convert from SLAX to TCL
  - dmzlink-bw functions differently on XR than JunOS
    - Cisco treats it per-interface, while Juniper treats it per prefix
      - Cisco wrote a SMU (patch) for BGP & FIB processing to make it per-prefix in about a week. The functionality will go main-line in the next major release
Downtime…

- We divide each POP into two “stacks”
  - A stack is one router, 20 “spinning disk” appliances, and 30 “flash” appliances
- We cut over each stack at approximately 7AM local time
- Total downtime per stack? ~20 minutes
Netflix Prefix Transitions
Blue dots are advertisements
Yellow-red dots are withdrawals
Nearly 400 IPv4 peers represented here, one Renesys peer per line
Percentage peers with valid routes plotted at top
LAX2: Friday 8 March 2013 (1h35m)
ATL1: Friday 22 March 2013 (22m7s)
LAX1: Monday 1 April 2013 (31 seconds)

ORD1: Tuesday 16 April 2013 (7m42s)

ATL2: Tuesday 23 April 2013 (1m28s)

How do we do it so quickly?

- Pre-staged configurations
  (of course)
- MTP cabling
  - There are no home runs anywhere
MTP Cabling

- Each host uses a MTP to LC whip that allows for rapid deployment of cabling to each rack
- A rack of 30 Flash Hosts (120 10G ports) takes approximately 45 minutes to wire
MTP Cabling (Demarc)

- We do the same for demarcation
  - Colo providers never touch routers
2 Terabits in a day

- We keep configurations templated and homogenous
- Cabling are custom made pre-wire bundles (MTP to LC breakout) – the only options we select are length
- Every colo looks basically the same – 5-7 racks
  - We decide how much infrastructure to deploy based on geographic sizing
- Colo vendors never touch our routers
  - Cross connects are run to MTP panels which are pre-wired to routers

All of this means that we can deploy 2T of infrastructure in ~1 day
Questions?