Software Defined Infrastructure: a direction for networked computing

Fred Baker
Cisco Fellow

28 April 2014
Topics today

- Routing
  - Interdomain name-based networking
  - FIB/RIB Route Lookup
  - Estimation of traffic matrices

- Novel techniques
  - Hiding mobility, multiplexing, and multi-homing from an application
  - Experimentation with IP Multicast
  - AQM experimentation

- Content
  - Caching for content-centric networks
  - Interactions between Youtube video and Facebook users

- Interesting:
  - Some topics relatively far out
    - name-based and content-centric networks
  - Some topics looking to improve today’s network
    - The rest

- Industry perspective
  - That’s good
  - We need help today, and direction for tomorrow
I’d like to discuss...

- Changes happening in the industry, in data center, enterprise, and service provider networks
- Principles that, if applied, will help research to influence the global network
Post-Snowdon: pervasive encryption

- Making some waves...

- Today, mobile 3G/4G/LTE networks provide value-added services that insert ads or re-route traffic to cached content
  - Selected by mobile network, not user

- Starting to see http 2.0 (SPDY) proxy networks that prevent that
  - Re-routes all HTTP traffic, TLS encrypted, to Google servers that act as “trusted proxies”
  - Selected by Chrome browser, not user

- “Don’t be evil”...

Data from a nameless mobile customer

Figure 1: Mobile Streaming Destinations

70% to SPDY sites

Figure 2: Mobile HTTP Destinations

60% to SPDY sites
KAIKAKU FOR IP NETWORKS
INDUSTRY LEADERSHIP

From
- IPv6
- MPLS
- PPPoE
- DHCP
- Tunnel
- TE
- FRR
- MPLS TP
- OTN
- XXX GE
- GMPLS
- ATM
- SDH

To
- IPv6
- Tunnel
- XXX GE
- DHCP

TeraStream
- Drastic simplification of IP networks
- IP & Optical integration
- Infrastructure Cloud model
IPv6 deployment

- Growing allocations
  - APNIC, RIPE, LACNIC, and ARIN now below one /8 of IPv4
  - About 17% of AS’s worldwide have IPv6 allocations, and growing
    - [http://v6asns.ripe.net/v/6?](http://v6asns.ripe.net/v/6?)
      - s=_ALL;s=_RIR_RIPE_NCC;s=_RIR_ARIN;s=_RIR_APNIC;s=_RIR_LACNIC;s=_RIR_AfriNIC

- Growing traffic
  - US: today about 7% of traffic. May 2015, 13–20%
  - Germany: today, about 8% of traffic. May 2015, 18–50%
  - Consistent with standard logistic curve
    - [https://www.vyncke.org/ipv6status/project.php](https://www.vyncke.org/ipv6status/project.php)

- Facebook internally IPv6-only for most services
  - Reason: reduction of cost in various forms

- Consistent with projections showing IPv6 replacing IPv4 over time
Software Defined Networking

• Not a technology: It’s a paradigm
• Many implementations calling themselves SDN

Key characteristics:
• Centralized configuration management
• Often centralized routing control
• In data centers, network and host virtualization (“cloud”)

Widely experimented with, no single solution at this point to say is “deployed”

“When will SDN have universal deployment?”

(Question I was asked not long ago by media)
Things I’m thinking about...
Data Center Latency Control
Best shown using an example...

Ping RTT from a hotel to Cisco overnight
RTT varying from 278 ms to 9286 ms

Delay distribution with odd spikes about a TCP RTO apart;
Suggests that we actually had more than one copy of the same segment in queue

What is buffer bloat? Why do I care?

*Because few applications actually worked*
Persistent Deep Queues

- In access paths (Cable Modem, DSL, Mobile Internet)
  - Generally results from folks building a deep queue with permissive drop thresholds
  - One DSL Modem vendor provides ten seconds of queue depth

- In multi-layer networks (WiFi, Input-queued Switches)
  - *Channel Acquisition Delay*
  - Systems not only wait for their own queue, but to access network
  - In WiFi, APs often try to accumulate traffic per neighbor to limit transition time
  - In Input-queued switches, multiple inputs feeding the same output appear as unpredictable delay sources to each other
  - In effect, managing *delay through queue*, not queue depth
- Names withheld for customer/vendor confidentiality reasons
- Common social networking applications might have
  - $O(10^3)$ racks in a data center
  - 42 1RU hosts per rack
  - A dozen Virtual Machines per host
  - $O(2^{19})$ virtual hosts per data center
  - $O(10^4)$ standing TCP connections *per VM* to other VMs in the data center
- When one opens a <pick your social media application> web page
  - Thread is created for the client
  - $O(10^4)$ requests go out for data
  - $O(10^4)$ 2–3 1460 byte responses come back
  - $O(45 \times 10^6)$ bytes in switch queues *instantaneously*
  - At 10 GBPS, *instant 36 ms queue depth*
Taxonomy of data flows

- We are pretty comfortable with the concepts of mice and elephants
  - “mice”: small sessions, a few RTTs total
  - “elephants”: long sessions with many RTTs

- In Data Centers with Map/Reduce applications, we also have **lemmings**
  - $O(10^4)$ mice migrating together

- Solution premises
  - Mice: we don’t try to manage these
  - Elephants: if we can manage them, network works
  - Lemmings: Elephant-oriented congestion management results in HOL blocking
ECN/Loss-based TCP Congestion Control (CUBIC and NewReno)
- Works reasonably well on 10 ms and 100 ms timescales
- Works marginally well on geosynchronous satellite RTTs (slow start issues)
- Do we all agree that it works on 1000 microsecond and shorter timescales?

Would latency control work better if we had the endpoints measuring RTT and actively maximizing throughput while minimizing latency?
Simple model of TCP throughput dynamics

\[ \text{mean throughput} = \frac{\text{effective window in bytes}}{\text{mean round trip time}} \]

- **Effective Window**: the amount of data TCP sends each RTT
- **Knee**: the lowest window that makes throughput approximate capacity
- **Cliff**: the largest window that makes throughput approximate capacity
- Note that throughput is the same at knee and cliff.

Increasing Measurable Throughput

Increasing TCP Window

Bottleneck Capacity

“knee”

Queue Depth

“cliff”

Yes, there is a more complex equation that takes into account loss. It estimates throughput above the cliff.
Delay-based Congestion Control

- Arguably the most stable approach
- Several algorithms:
  - Vegas
  - CalTech FAST
  - Swinburne CDG
- Applicable to TCP, DCCP, or SCTP

- CalTech FAST
  - Simple,
  - IPR issues
  - Yields systemically to loss-based models,
  - Tunes to knee plus alpha

\[
cwnd' := cwnd \times \frac{\text{base RTT}}{\text{mean RTT}} + \alpha
\]

- Swinburne CAIA Delay Gradient
  - Implemented in FreeBSD 9.2 and later
  - Tunes to minimize variation in delay when it can, loss if it determines it is competing with a loss-based competitor
Swinburne CAIA Delay Gradient

- *Neither loss-based nor delay-based*
  - Responds to observed jitter
  - Switches to a NewReno mode when necessary
  - Recent papers suggest should enhance Data Center communications

Software Defined Infrastructure:

simplification of the network

simplification of the networked application
Relationship between the application and the network

- **End to end principle:**
  - “functions placed at low levels of a system may be redundant or of little value when compared with the cost of providing them at that low level”
    - A plea for simplicity...
  - Network configuration that disrupts the intent of the application ultimately hurts both
    - Classic examples around network address translation, transcoding, etc

- *The network is the application’s “brother”, not its “friend”*
<table>
<thead>
<tr>
<th>Simplicity Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Complexity is the primary mechanism which impedes efficient scaling, and as a result is the primary driver of increases in both capital expenditures (CAPEX) and operational expenditures (OPEX).”</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Amplification Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>“There are non-linearities at large scale which do not occur at small to medium scale.”</td>
</tr>
<tr>
<td>Make local changes have only local effect</td>
</tr>
<tr>
<td>Small local changes with global effect destabilize a system, and</td>
</tr>
<tr>
<td>Attempts at global changes have significant local effect</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coupling Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>“As things get larger, they often exhibit increased interdependence between components.”</td>
</tr>
<tr>
<td>Issues with translation and session management often come down to coupling between assumptions about addressing</td>
</tr>
</tbody>
</table>

RFC 3439: “Some Internet Architectural Guidelines and Philosophy”
Simplification of configuration: Netconf/Yang

- An issue in network configuration has been – and continues to be – the differences in data models between software and hardware from different minds
  - Different vendors
  - Different open source projects
  - Different viewpoints

- Not a new observation:
  - A ... goal is that the architecture be, as much as possible, independent of the architecture and mechanisms of particular hosts or particular gateways.

- Configuration paradigms moving to a model in which
  - A configuration expresses the “intent” of the administration
  - The underlying software interprets it in the local context
  - Actual configuration now limited to values that actually have to be assigned for things to work, such as prefixes (and maybe not those)
Embedded operational intelligence:
Simplification of monitoring and diagnosis

- When something is broken in the network, how do you find it?
- When something is broken in the network, how do you find it?
- Software raises alerts when it detects changes, rather than waiting for the operator or management system to come looking for them.
- SLAs and traditional Reliability, Availability, and Serviceability (RAS) continuously measured

- Example: large data center customer manages incast traffic
- Can improved algorithms help?
  - Would a delay-based or jitter-based latency control procedure in TCP/SCTP work better?
Embedded operational intelligence: Simplification of deployment

- Autonomous networks:
  - A newly-installed system identifies itself to a server
  - Server tells it what the administration's intent for it is
  - It then configures itself accordingly

- draft-ietf-ospf-ospfv3-autocfg
  - OSPF IPv6 prefix distribution within an area
  - What if we could number, and renumber, a network without operator interaction?
  - (The things that make renumbering hard are usually poor software methodologies)
A word...
In your research...

- Think about the fundamental principles of networking:
  - End to end: services that do what is expected of them
  - Simplicity vs Complexity
  - Amplification: managing non-linearity
  - Coupling: managing interconnectedness of components

- Test out your ideas
  - Prove that they work scaleably in real world, not just simulation