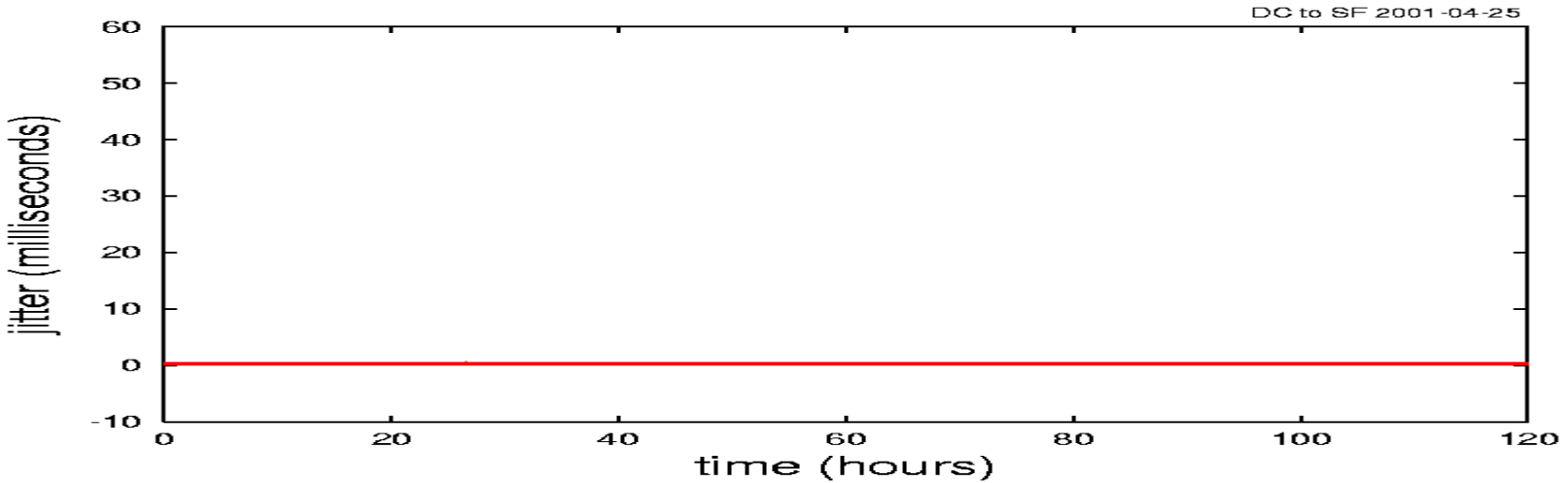
The Role of Analytics in Routing, Network Performance and SDN

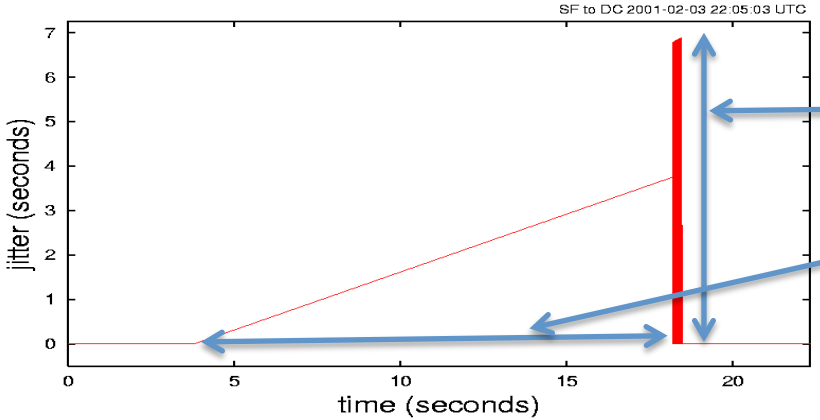
Cengiz Alaettinoglu

It all started with a jitter study (2000)

- Studied jitter on 3 US and 1 European backbones for several weeks
- For 99.99% packets, measured jitter < 1ms

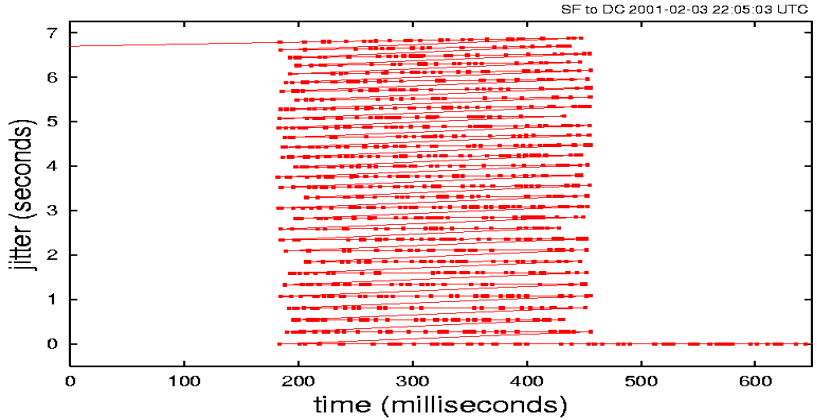


However, 0.01% of Jitter was severe



7 seconds jitter

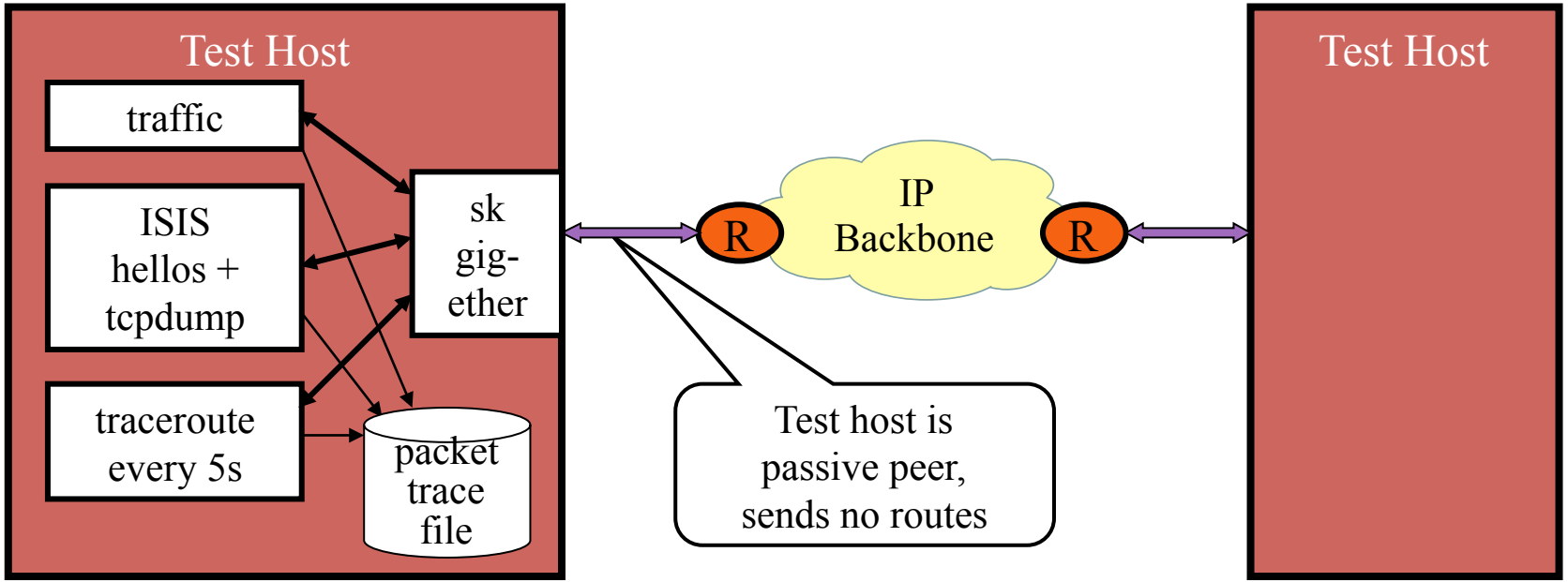
10+ seconds packet drops



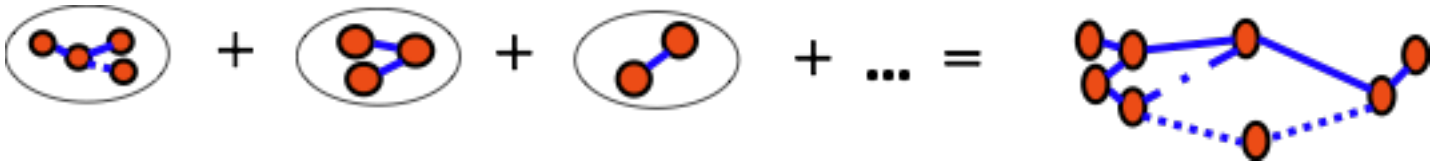
Severe packet reordering

Theory: Packets being spewed out from an unwinding routing loop…

- Did we really have long routing loops in the network?
- Did ISIS really take 10+ seconds to converge?
- So, we ***analyzed routing*** along with jitter

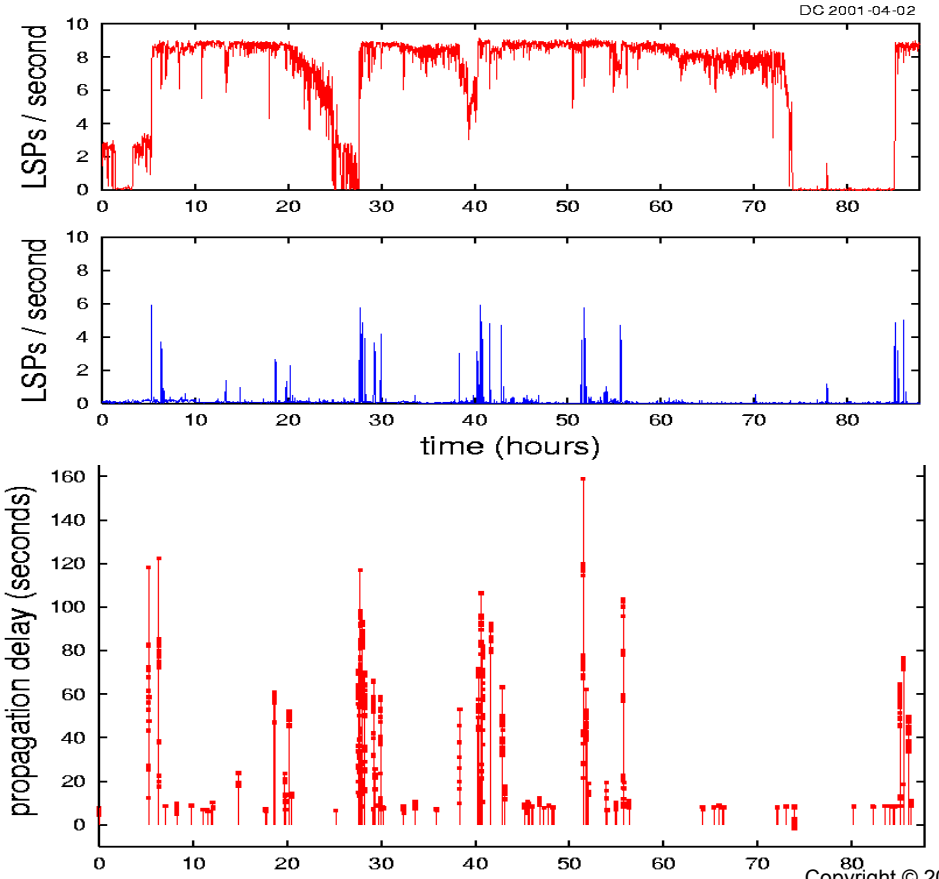


ISIS in a Nutshell



- Each router sends Link State Packets (LSP) that describes its local topology
  - List of links to neighbors (adjacencies), prefixes along with metrics
  - This is refreshes periodically, otherwise LSP content is purged
- This is flooded across the network
  - Each router send new LSPs it receives to its neighbors
  - Neighbors sends to their neighbors, …
- Each router accumulates LSPs into a database (LSDB) and constructs the current view of the topology
- Shortest paths are computed using Dijkstra's SPF algorithm

Excessive ISIS churn caused excessive LSP Propagation Delay



All link state packets (LSPs) including refreshes

LSPs that report a change

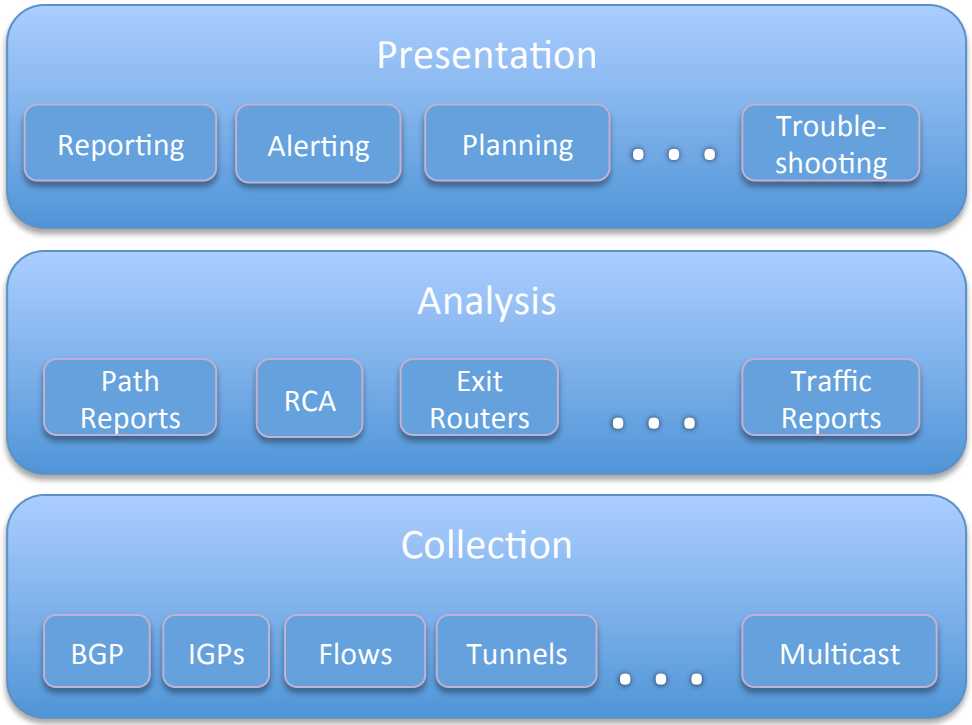
LSP propagation delay

- Seconds between seeing the same LSP in the east and the west coasts of the US

Explanation

- Link state databases were not in sync:
  - Very large LSP databases
  - High churn rate caused many LSPs to flood
  - LSP rate-control slowed down flooding
  - Any topology change could result in a loop under these conditions
- We realized being able to look at routing was key for powerful network performance analysis
- Today, we see very high churn in very large TE databases using auto-bandwidth with large number of tunnels

Route Analytics Today



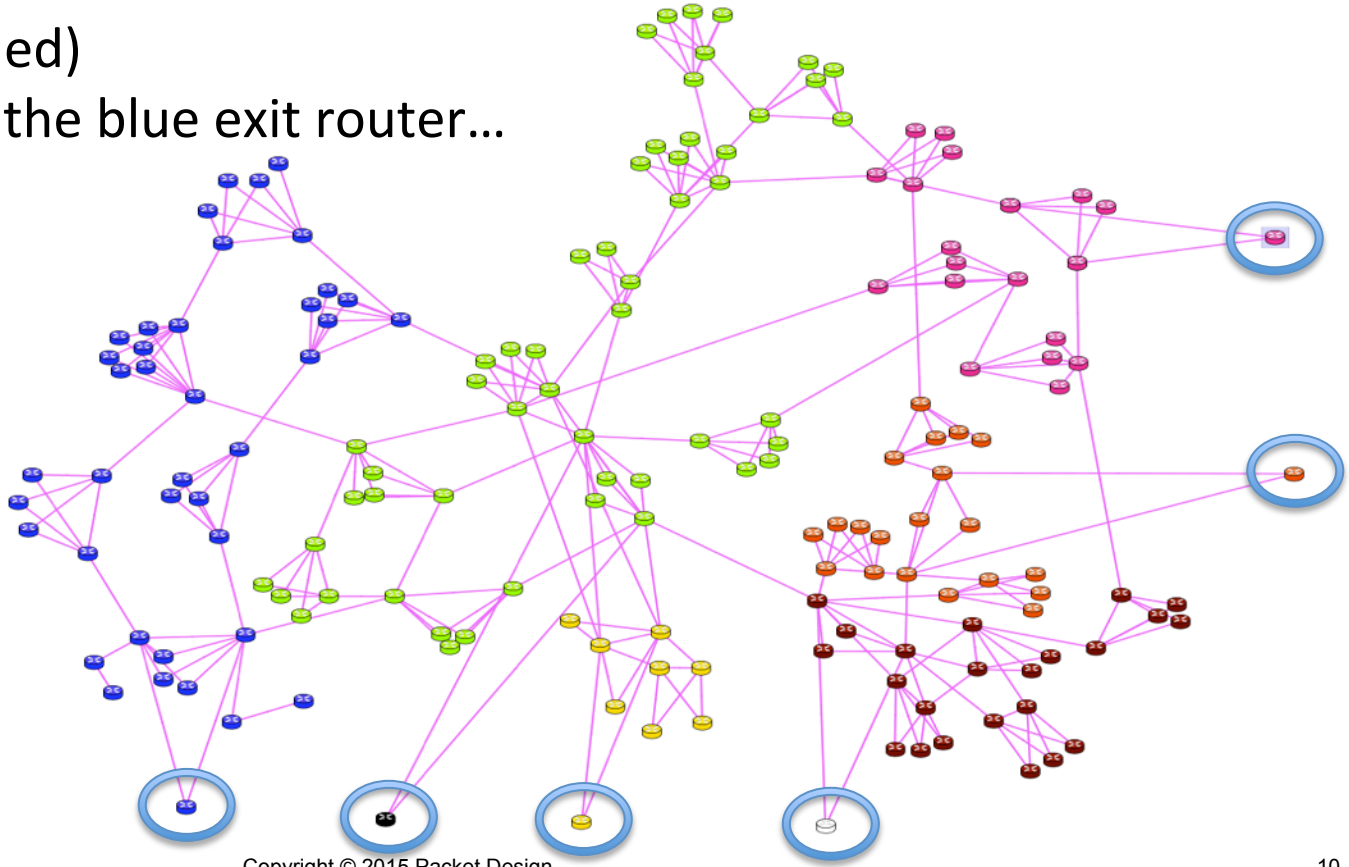
- Troubleshooting and visualization
- Service/application monitoring and alerting
- Network health assessment
- Topology-aware traffic analysis
- Proactive change modeling
- Analytics-driven Software Defined Networking applications

Use Case: Diagnosing Black Holing

- A peering router to a major service provider crashed
  - Hot swappable card was not quite so…
- Traffic to the SP was black-holed network-wide
  - Traffic exiting all 6 locations was black-holed
- About 3 minutes of routing outage
  - 3 minutes was too short to diagnose the issue at human speed
  - Had a 45 minute impact on the services and ad revenues
    - Users who could not use the service did something else

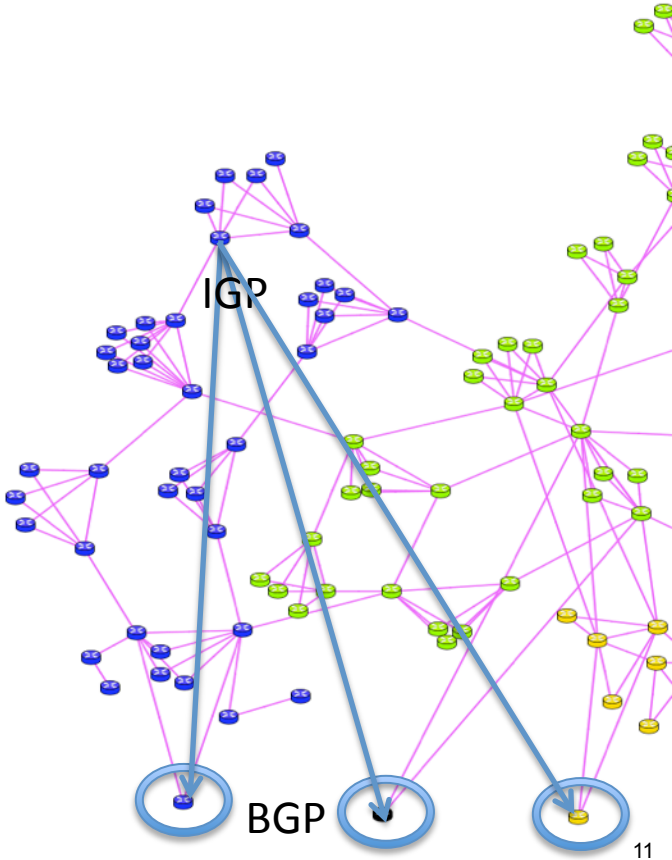
Expected Exit-Points Before Incident

- 6 Exit-Points (circled)
- Blue routers take the blue exit router…

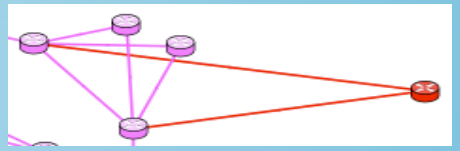


Recursive Route Resolution

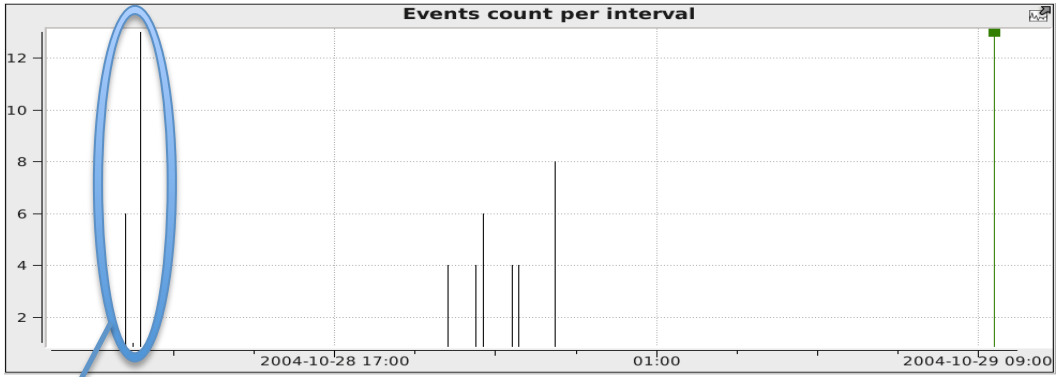
- BGP determines exits
  - NextHop attribute
- Usually IGP distance determines the closest
- More accurately, we recursively find a path to NextHop
  - IGP, static, BGP, or a series…



The Incident



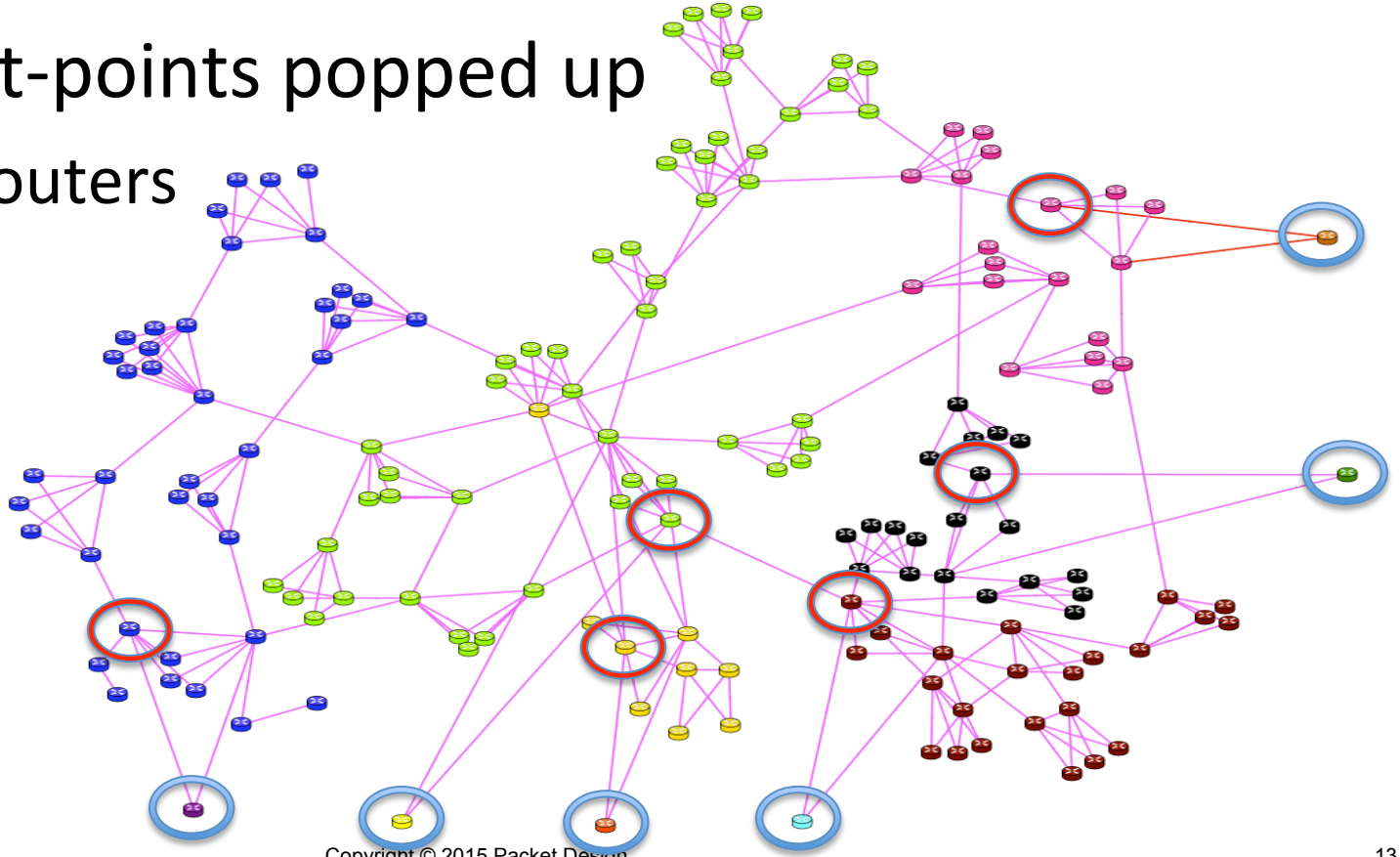
- ISIS activity during incident



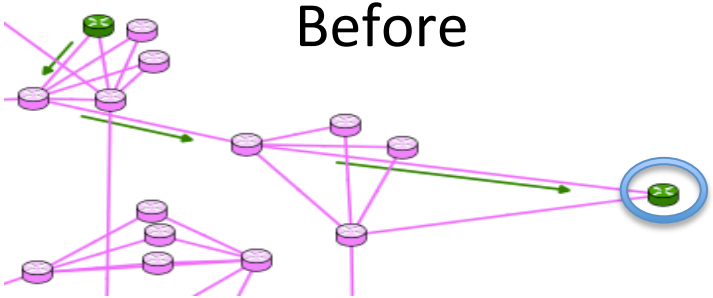
Time	Router	Operation	Operand	Attributes
2004-10-28 07:36:11.974206	core-ord-01	Drop Neighbor	edge-ord-02	Metric: Down (TE)
2004-10-28 07:36:12.374093	core-ord-02	Drop Neighbor	edge-ord-02	Metric: Down (TE)
2004-10-28 07:38:09.063564	core-ord-01	Add Neighbor	edge-ord-02	Metric: 503 (TE)
2004-10-28 07:38:36.071999	core-ord-02	Add Neighbor	edge-ord-02	Metric: 503 (TE)

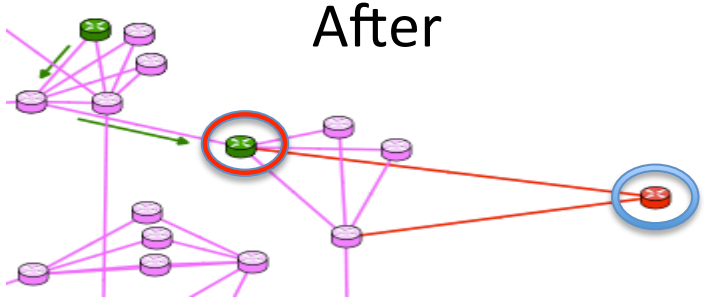
Exit-Points During Incident

- 6 more exit-points popped up
  - All core routers
  - No EBGP



A Path Before and After the Incident





BGP Next hop resolution: before 128.9.129.1/32 in ISIS vs. after 128.9.128.0/19 in BGP

Path	Source Node	Destination Node	Protocol	Resolved by Prefix
edge-dfw-03→ 199.221.80.0/24				
Hop 1	edge-dfw-03	core-dfw-01	BGP	199.221.80.0/24
Hop 2	core-dfw-01	core-aus-01	BGP	199.221.80.0/24
Hop 3	core-aus-01	edge-aus-01	BGP	199.221.80.0/24
Lookup 1			ISIS	128.9.129.1/32

Route Recursion

Path	Source Node	Destination Node	Protocol	Resolved by Prefix
edge-dfw-03→ 199.221.80.0/24				
Hop 1	edge-dfw-03	core-dfw-01	BGP	199.221.80.0/24
Hop 2	core-dfw-01	core-aus-01	BGP	199.221.80.0/24
Self Hop	core-aus-01	core-aus-01	BGP	199.221.80.0/24
Lookup 1			BGP	128.9.128.0/19

Cause of Black Holing

- Every SP announces its address space externally in BGP
  - 128.9.128.0/19 BGP route is for this purpose
  - But it also resolves the NextHop address 128.9.129.1/32
- When the peering router crashed
  - IGP routes from that router were withdrawn in milliseconds
  - BGP routes from that router were not withdrawn
    - 3 KEEPALIVEs of 60 seconds each – router rebooted before this
  - These BGP routes were now resolved by 128.9.128.0/19 in BGP
  - Injected by 6 core routers
  - Distance to a core router from any router is very low
  - Every router uses the dead router's BGP routes

- *We are good at designing networks when everything is up and running, but failure cases are often beyond our imagination*

Remedy

- Insert a really expensive static route for the /19 to ISIS
  - It should cost more than longest possible path in IGP
  - ISIS routes preferred over BGP routes and will hide the /19 BGP route in recursion
  - Now, when a peering router crashes, the traffic will choose a true exit
  - See: http://www.nanog.org/meetings/nanog34/presentations/gill.pdf

- Do not: Make IBGP session converge faster (like running BFD)
  - One may argue the root cause is that BGP was too slow to withdraw
  - You will lose the IBGP session each time the IGP path of the session changes

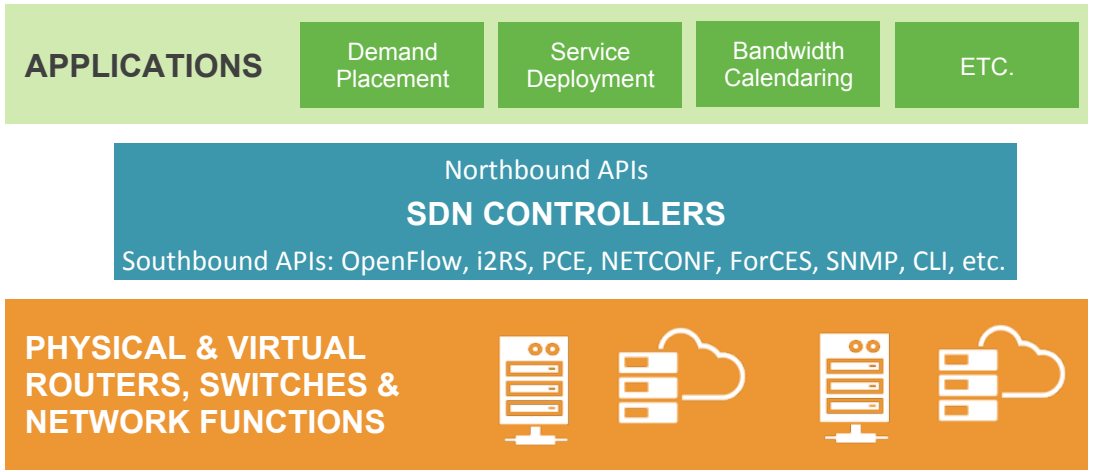
Challenges in Operating SDN

SDN makes networks programmable for

- Network overlays
- Bandwidth reservation
- Demand placement
- Service deployment
- Etc.

-- but --

What governs whether or not these programmatic changes should be made?
What will be their impact?



Need for Analytics-Driven Applications

- When major apps/services are introduced, planning groups validate capacity
  - Quality of Experience expectations
  - Capacity planning
  - Changes to the topology, CoS treatment, …
- If the apps/services are being rolled out without operator intervention, how do you plan for them?
  - SDN analytics addresses this concern

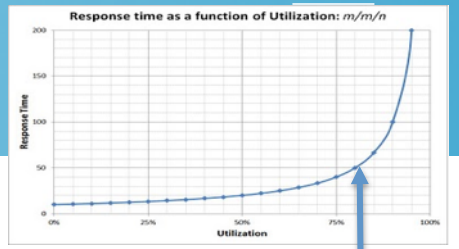
How Rich Analytics Help a Bandwidth Scheduling Application

- Bandwidth scheduling: can I move X bps from A to B at time t ?
- Attractiveness: SPs have abundance of spare bandwidth
  - Most SP networks have less than 50% utilization
    - Verizon: 46% average peak utilization
    - Level 3: 46-56% peak utilization range
- Can an SP profit from this spare capacity?
- But there are good reasons for this spare bandwidth

A Naïve Implementation

- Let's collect link utilizations
  - This is near real time; and SPs already have it
- We need utilizations at time t
  - Use historical link utilizations
  - Baseline: average same 5-minute or hour of the day for several weeks
  - Add projections for growth and safety
- Compute path from A to B and add X bps to the links
  - Go or no-go decision based on new link utilizations
- If go, schedule the SDN controller to set up this path from A to B at time t, and tear it down afterwards

Reasons for Spare Bandwidth



- Increased utilization ⇒ increased delay and jitter
  - Delay vs. link utilization curve has a sharp knee
- Network must accommodate failures
  - Network must have capacity to reroute the traffic around failures
  - Large networks have one link down at any given time, they must tolerate two link failures
- Traffic is growing but adding capacity takes time

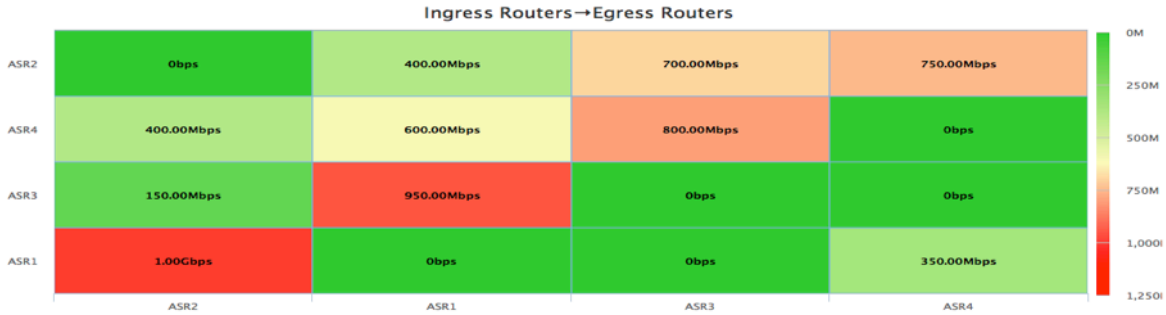
Addressing These Challenges

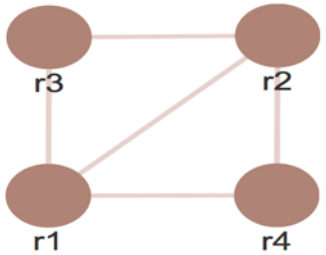
- Increased delay
  - Cap the go/no-go decision at ~65-70%
  - For anything above that we must be moving ***bulk*** traffic
    - Not suitable for uncompressed HD broadcast of an event
    - Not even suitable for best-effort traffic
    - Must deploy differentiated services
- Protect against failures via analytics driven simulation
  - Fail every (or two) link/router and see the impact on link utilizations
    - Not sufficient to fail just the links/routers along the path

Failure Impact: Where will the traffic go?

- We need to know where the traffic is entering the network, how much traffic there is, and where it is exiting the network
  - Link utilizations don't tell where the traffic is entering or exiting the network
- We need to understand the network's routing to compute the new paths

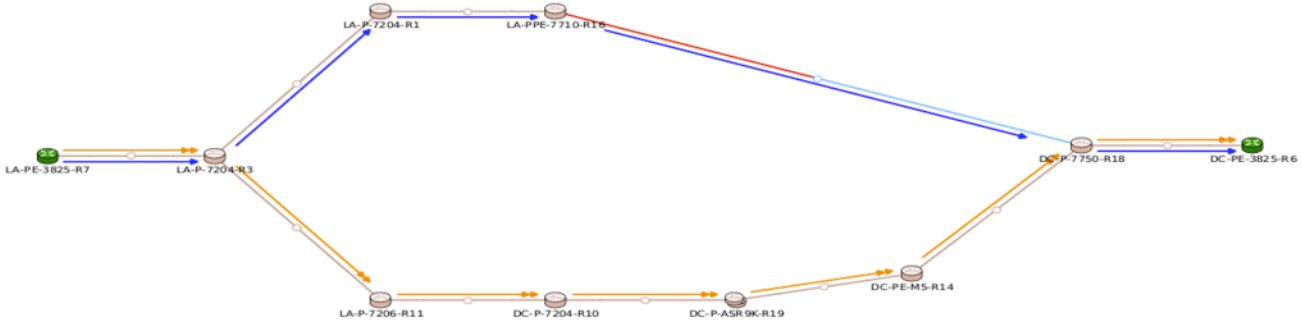
Need for a Traffic Demand Matrix





- Traffic Matrix:
  - For each router pair (r1,r2), how much traffic entering at r1 is exiting at r2?
- Flow data coupled with routing gives us traffic matrix

Simulation and Impact of a Failure



- For each flow on the failed link
  - Go to the ingress router and find the new path for the flow
  - Subtract flow's bandwidth from the old links
  - Add flow's bandwidth to the new links
  - Check to see if congestion crept in
- We need an accurate routing model of the network
  - Route analytics provides this for IGP, BGP, RSVP-TE, VPNs…

Concluding Remarks

- Routing impacts network performance
  - Availability and reachability
  - Sub-optimal paths with longer delays, jitter
- Route analytics proves to be very effective in
  - Troubleshooting, monitoring, alerting
  - Reporting and network health assessment
  - Routing-aware traffic analysis
    - BGP peering analysis
    - Traffic matrices
- Rich analytics are key for successful SDN deployment and applications, including bandwidth scheduling