SDN

• B4: Experience with a globally deployed software defined WAN – Jain et.al. SIGCOMM 2013

• SWAN: Achieving high utilization with software defined WAN, Hong et al., SIGCOMM 2013

• OPenSketch: Minlan YU et.al., NSDI 2013

• Ack: Slides from Conference presentations
B4: Experience with a Globally-Deployed Software WAN
B4: Google’s Software Defined WAN
B4: Google’s Software Defined WAN

• Google’s private WAN connecting its data centers
  – Elastic bandwidth demands
    • Can tolerate periodic failures with temporary BW reduction
  – Small number of sites
    • Allows special optimization
  – Complete control of end application
    • Application priorities and control bursts
  – Cost Sensitivity
    • Unsustainable cost projection with traditional approach (2-3x cost of a fully utilized WAN).
B4 SDN architecture

• Switch hardware (Google custom designed with commodity silicon)
  – Forwards traffic
  – No complex control software

• OpenFlow controllers (OFC – ONIX based)
  – Maintain network state based on network control application directive and switch events
  – Instruct switches to set forwarding entries

• Central application
  – Central control of the entire network
B4 architecture overview
Traditional WAN routing

- Treat all bits the same
- 30% ~ 40% average utilization
- Cost of bandwidth, High-end routing gear
Traffic priority

- User data copies to remote data centers for availability/durability (lowest volume, most latency intensive, highest priority)
- Remote storage access for computation over distributed data sources
- Large-Scale data push synchronizing state across multiple data centers (highest volume, least latency intensive, lowest priority)

- Centralized traffic Engineering (TE)
  - Near 100% utilization
  - Fast, global convergence for failures.
B4 design decisions

- B4 routers built from merchant switch silicon
  - APPS trade bandwidth for fault tolerance
  - Edge control $\rightarrow$ reduced the buffer size, number of B4 site $\rightarrow$ small forwarding table
  - Low router cost $\rightarrow$ scale network capacity
- Drive links to 100% utilization
  - Effective use of expensive long haul transport
  - High average bandwidth over predictability: largest bandwidth consumers can adapt to bandwidth availability
- Centralized traffic engineering
  - Multipathing
  - Application classification and priority
  - Improved over traditional TE schemes
  - Faster, deterministic global convergence for failures
- Separate hardware from software
  - Customized routing
  - Rapid iterating of software protocols
  - Easier to protect against common case software failures
  - Agnostic to range of hardware deployment
Centerlized TE: convergence after failure

- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20
Centerlized TE: convergence after failure

- Flows: R1→R6: 20; R2→R6: 20; R4→R6: 20
Centerlized TE: convergence after failure

- Flows: $R1 \to R6: 20; \ R2 \to R6: 20; \ R4 \to R6: 20$

- R5-R6 link fails
  - R1, R2, R4 autonomously find next best path
Centerlized TE: convergence after failure

- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20

- R5-R6 link fails
  - R1, R2, R4 *autonomously* try for next best path
  - R1, R2, R4 push 20 altogether
Centerlized TE: convergence after failure

- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20

- R5-R6 link fails
  - R1, R2, R4 autonomously try for next best path
  - R1 wins, R2, R4 retry for next best path
  - R2 wins this round, R4 retries again
Centerlized TE: convergence after failure

- Flows: R1->R6: 20; R2->R6: 20; R4->R6: 20

- R5-R6 link fails
  - R1, R2, R4 autonomously try for next best path
  - R1 wins, R2, R4 retry for next best path
  - R2 wins this round, R4 retries again
  - R4 finally gets third best path!
Centerlized TE: convergence after failure

- **Simple topology**

- **Flows:**
  - R1->R6: 20; R2->R6: 20; R4->R6: 20

- R5-R6 fails
  - R5 informs TE, which programs routers in one shot
Centerlized TE: convergence after failure

- Simple topology

- Flows:
  - R1->R6: 20; R2->R6: 20; R4->R6: 20

- R5-R6 link fails
  - R5 informs TE, which programs routers in one shot
  - Leads to faster realization of target optimum
Advantage of Centralized TE

- Better network utilization with global pictures
- Converges faster to target optimum on failure
- Allows more control and specifying intend
  - Deterministic behavior simplifies planning vs. overprovisioning for worst cast variability
- Can mirror production event streams for testing
- Controller uses modern server hardware – better performance (50x!)
Background: Inter-DC WANs

Inter-DC WANs are critical

Inter-DC WANs are highly expensive
Two key problems

Poor efficiency
average utilization over time of busy links is only 30-50%

Poor sharing
little support for flexible resource sharing

Why?
One cause of inefficiency:
lack of coordination
One cause of inefficiency: lack of coordination

![Graph showing background and non-background traffic over time.](image)

- **Norm. traffic rate**
- **Background traffic**
- **Non-background traffic**

Time (~ one day)
Local, greedy resource allocation hurts efficiency

flow arrival order: A, B, C
each link can carry at most one flow

<table>
<thead>
<tr>
<th>Flow</th>
<th>Src → Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 → 6</td>
</tr>
<tr>
<td>B</td>
<td>3 → 6</td>
</tr>
<tr>
<td>C</td>
<td>4 → 6</td>
</tr>
</tbody>
</table>
Local, greedy resource allocation hurts efficiency

flow arrival order: A, B, C
each link can carry at most one flow

<table>
<thead>
<tr>
<th>Flow</th>
<th>Src → Dst</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 → 6</td>
</tr>
<tr>
<td>B</td>
<td>3 → 6</td>
</tr>
<tr>
<td>C</td>
<td>4 → 6</td>
</tr>
</tbody>
</table>
Local, greedy resource allocation hurts efficiency

flow arrival order: A, B, C
each link can carry at most one flow

MPLS-TE

Optimal
System flow

[global optimization for high utilization]

SWAN controller

traffic demand → rate allocation

Hosts

network configuration → topology, traffic

WAN switches
Software-Defined Traffic Measurement with OpenSketch

Lavanya Jose
Stanford University

Joint work with Minlan Yu and Rui Miao at USC
Management is Control + Measurement

- Control
  - Access Control
  - Routing

- Measure
  - DDoS
  - Flow Size Distribution
Questions we want to ask

1. Who’s sending a lot to 10.0.2.0/16? (Heavy Hitters)
2. How are flow sizes distributed?
3. Is someone doing a port scan?
4. Is someone being DDoS-ed?
5. Who’s getting traffic from blacklisted IPs?
6. How many people downloaded files from 10.0.2.1?
Sketches as building blocks

- Sketch
- Data structure
- Support approx. computing some function of data
- Much smaller than actual data
- Streaming, small per-item processing cost
- Provable space-accuracy tradeoffs
Sketches as building blocks

e.g., Count Min sketch
to store counts of frequent source IP addresses

Source IP address: 23.43.12.1

(Cormode 2005)
Sketches as building blocks

e.g., Count Min sketch
to store counts of frequent source IP addresses

# packets from 23.43.12.1?

query

estimate

pick min.

(Cormode 2005)
Bitmap Sketch with the Pipeline

to store number of different destination port numbers

# different destination port numbers?

query

1 0 1 0 0 1 0 0 1 0

estimate

6/10

Six counters out of ten are 0.

(Whang 1990)

estimate

$N = -10 \ln(6/10) = 5$
3-stage pipeline

Packet

Hash

Classify

Count

header fields

pick fields to hash

hash values

pick field to match

hash values

compute counter addresses

header fields

header fields
3-stage pipeline

1. Who’s sending a lot to 10.0.2.0/16? (Heavy Hitters)
2. How are flow sizes distributed?
3. Is someone doing a port scan?

- Identifying heavy “keys”
- Counting, storing statistics
- Picking packets to measure

- Hash
- Classify
- Count
Similar functions, diverse configurations

- Count Min: 3
- Bloom Filters: 7-8
- Fixed size reversible sketch: 5
- Can share hash functions

Hash

4-8 simple hash functions per question
Similar functions, diverse configurations

- Match a prefix/value: 1 rule
- Match a set of values: Bloom Filters

Classify

30-40 TCAM entries per question maximum
Similar functions, diverse configurations

Count

From simulation and worst case bounds for different tasks

up to 8MB SRAM
Conclusion

• Current switches good for flow statistics
• But they don’t answer basic measurement questions
• Like identify heavy hitters, detect DDoS attacks, port scans, traffic from blacklisted IP address etc.
Takeaway

• Hash, classify and count pipeline in the Data Plane

• And sketch based building blocks in the Control Plane

• Make measurement in switches efficient and easy