SDN at Google

Opportunities for WAN Optimization

Edward Crabbe, Vytautas Valancius
8/1/2012

some slides taken from Urs Hölzle's ONS 2012 keynote

Topics

Google™

- SDN at Google today
- Example SDN Use Case: TE
- Our SDN Experience So Far
- Research Opportunities

Topics

Google™

- **SDN at Google today**
- Example SDN Use Case: TE
- Our SDN Experience So Far
- Research Opportunities

Google's WAN

Google

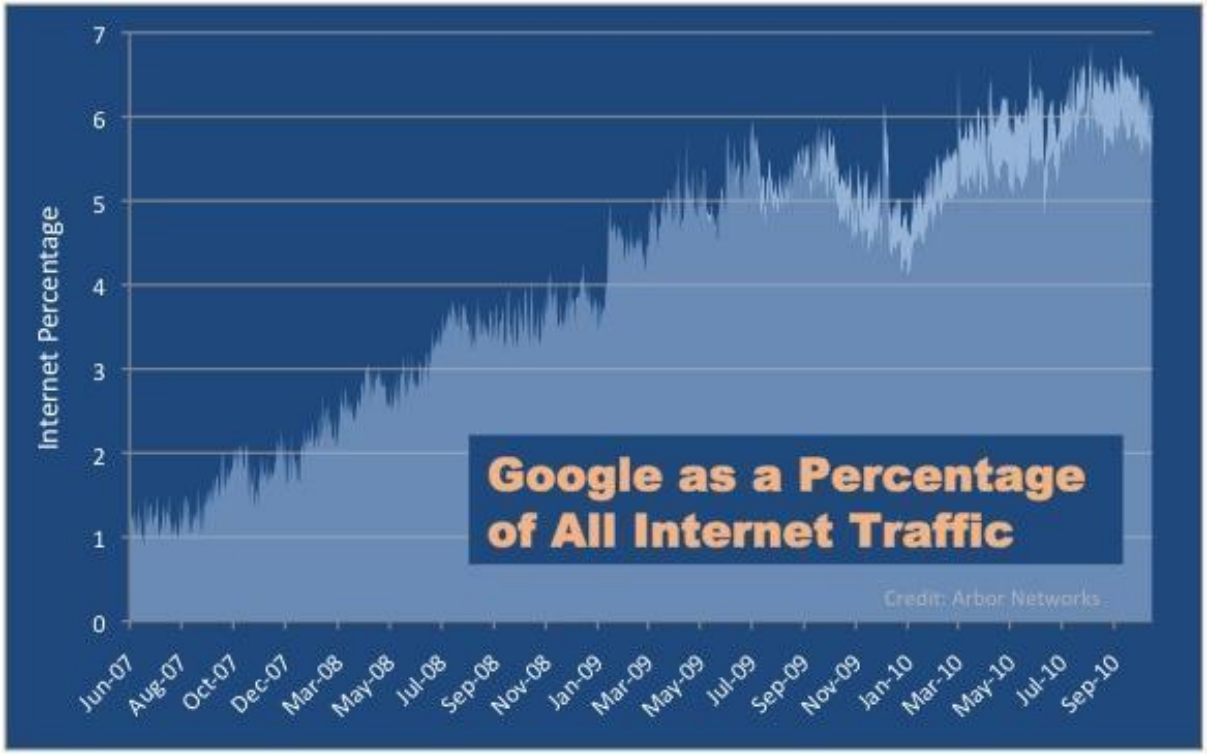
- Two backbones
  - Internet facing (user traffic)
    - smooth/diurnal
    - externally originated/destined flows
  - Datacenter traffic (internal)
    - bursty/bulk
    - all internal flows
- Widely varying requirements: loss sensitivity, availability, topology, etc.
- Difference in node density, degree and geographic placement
- thus: built two separate logical networks
  - I-Scale
  - G-Scale

Internet Backbone Scale

Google

"If Google were an ISP, as of this month it would rank as the second largest carrier on the planet."

[ATLAS 2010 Traffic Report, Arbor Networks]



WAN TCO

Google

- Cost/bit should go down with additional scale, not up
  - Consider analogies with compute and storage

- However, *cost/bit doesn't naturally decrease with size*
  - Complexity in pairwise interactions and any-to-any communication requires more advanced forecasting and control mechanisms
  - Lack of control and determinism in distributed protocols necessitates worst case over-provisioning
  - Complexity of automated configuration to deal with non-standard vendor configuration APIs
  - existing routing mechanisms do not allow for
    - scheduling
    - optimization of explicit objectives

A Solution: WAN Fabrics

Google

- Goal: manage the WAN as a *system* not as a collection of individual boxes

- Current equipment and protocols don't allow this
  - Internet protocols are node centric, not system centric
  - lack of uniformity in support for monitoring and operations
  - Optimized for survivability and “eventual consistency” in routing

Why Software Defined WAN

Google™

- Separate hardware from software
  - Choose hardware based on necessary features
  - Choose software based on TE requirements (*not* protocol requirements)

- Logically centralized network control
  - More deterministic
  - More efficient

- Separate monitoring, management, and operation from individual boxes

- *Flexibility and Innovation Velocity*

Advantages of Centralized TE

Google™

- Better efficiency with global visibility
- Converges faster to *target optimum* on failure
- Higher Efficiency
  - allows for explicit definition of cost functions
  - allows for in-house development of optimization algorithms
- Deterministic behavior
  - simplifies planning vs. over-provisioning for worst case variability
  - Can directly mirror production event streams for testing
- Supports innovation and more robust SW development
- Controller uses modern server hardware
  - significantly higher performance

Topics

Google™

- SDN at Google today
- **Example SDN Use Case: TE**
- Our SDN Experience So Far
- Research Opportunities

Practical SDN TE Use Cases

Google™

- Deadlock Resolution
- Bin Packing
- Scheduling / Calendaring
- Predictability
- Adaptive TE Control Loops
- Constraint Relaxation
- GCO
- Max-Min Fairness

⋮

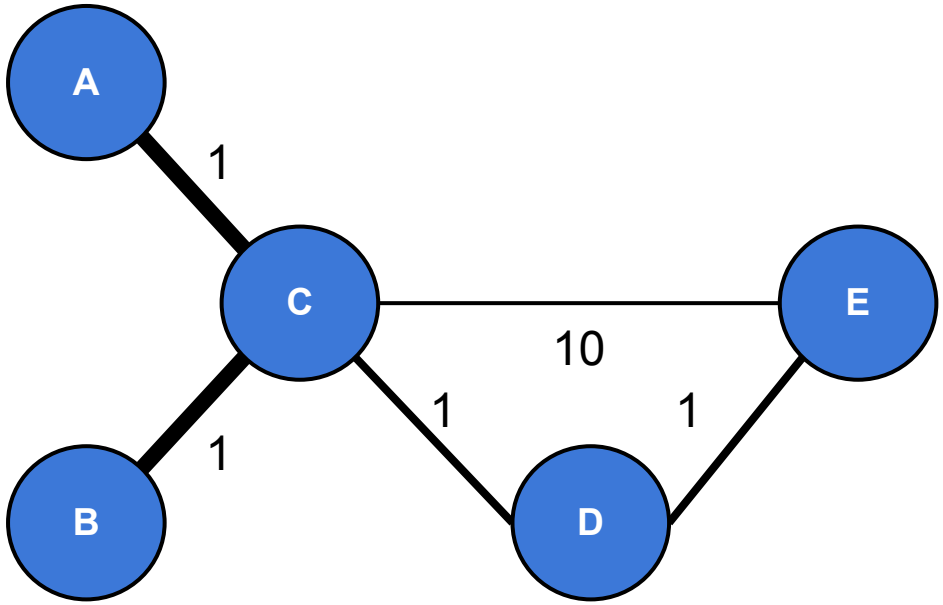
Practical SDN TE Use Cases

Google™

- Deadlock Resolution
- Bin Packing
- Scheduling / Calendaring
- Predictability
- Adaptive TE Control Loops
- Constraint Relaxation
- GCO
- Max-Min Fairness

⋮

Deadlock



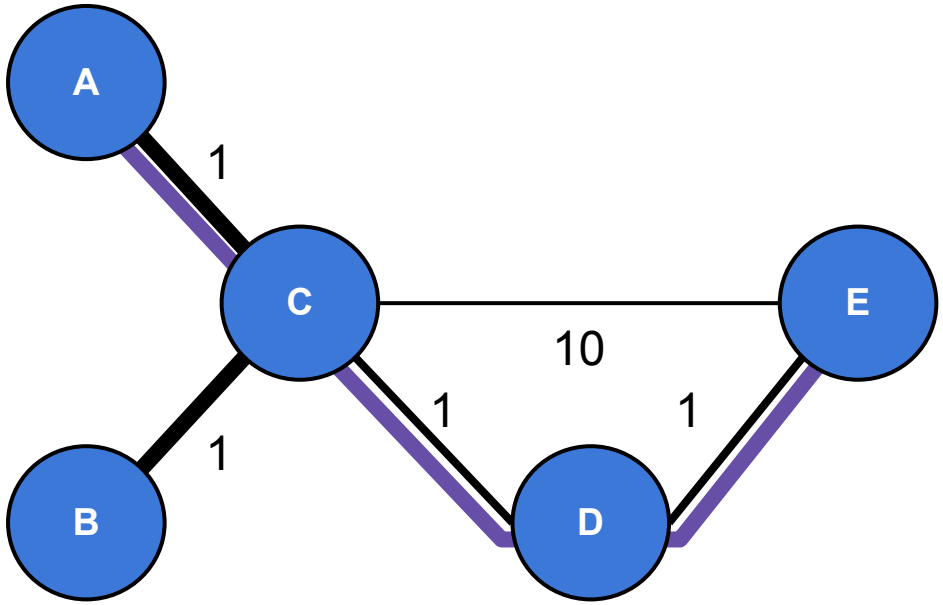
causes:

- control / dataplane decoupling
- rfc3209 implies no teardown on reservation increase failure
  - demand will be miss signaled for long periods
- lack of global LSP state
- lack of LSP level ingress admission control
  - would require another online or offline control mechanism
  - tension between overprovisioning level and transport elasticity

Link	Metric	Capacity
A-C	1	20
B-C	1	20
C-E	10	5
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	2
3	1	A	E	20

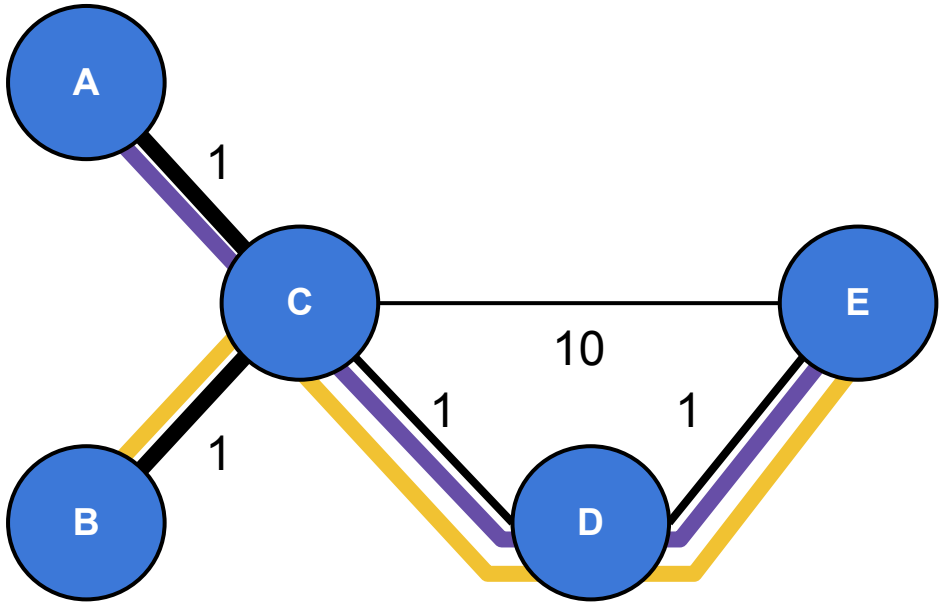
Deadlock



Link	Metric	Capacity
A-C	1	20
B-C	1	20
C-E	10	5
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	2
3	1	A	E	20

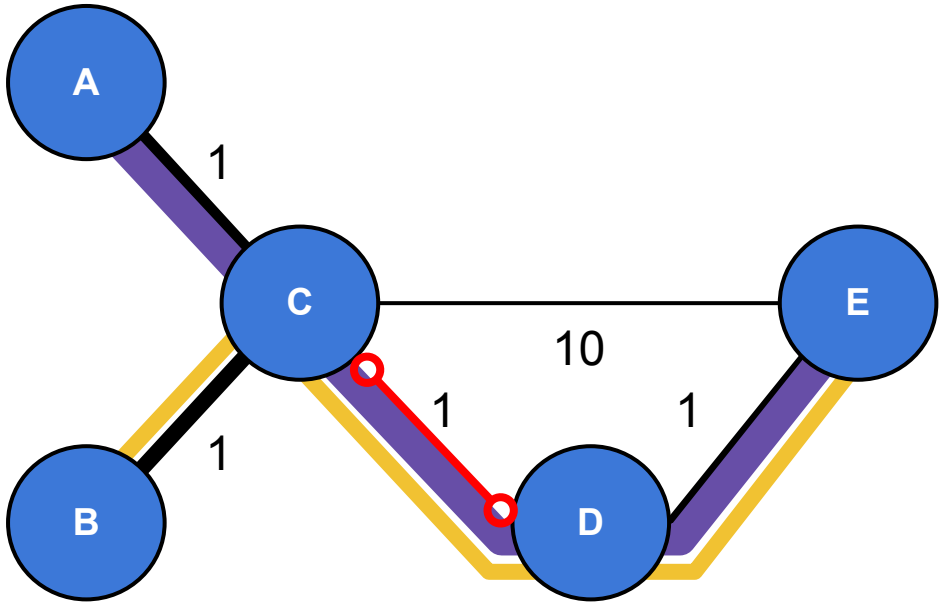
Deadlock



Link	Metric	Capacity
A-C	1	20
B-C	1	20
C-E	10	5
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	2
3	1	A	E	20

Deadlock

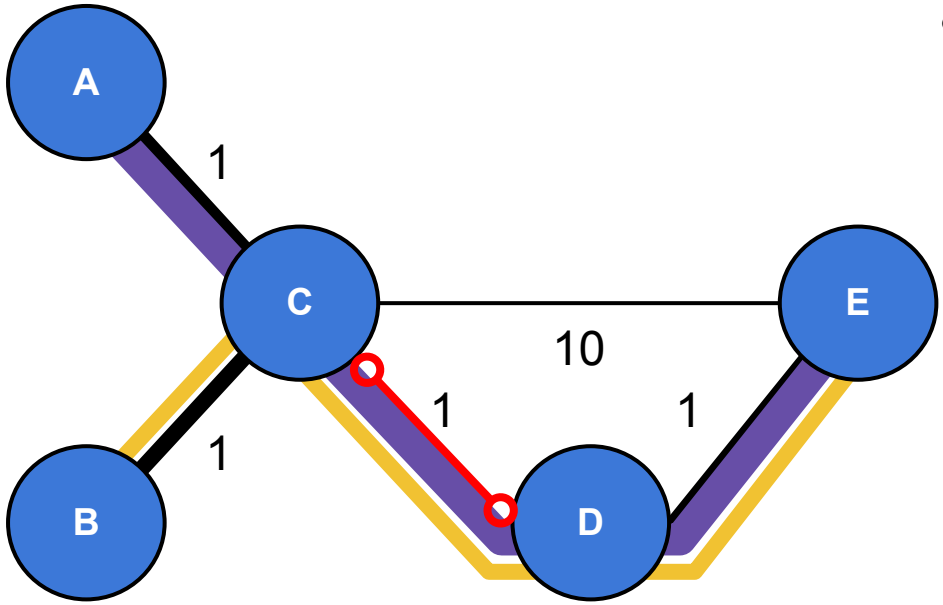


- LSP 1:
  - demand cannot be satisfied
  - LSP not torn down due to 3209
  - usage controlled due to control/data plane decoupling
  - ⇒ information in IGP, RSVP is inaccurate
- LSP 2
  - lack of visibility w/r/t LSP 1 misbehavior results in unecessary, potentially prolongued degradation in service
  - could be rerouted along C-E link modulo flow performance constraints

Link	Metric	Capacity
A-C	1	20
B-C	1	20
C-E	10	5
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	2
3	1	A	E	20

Deadlock

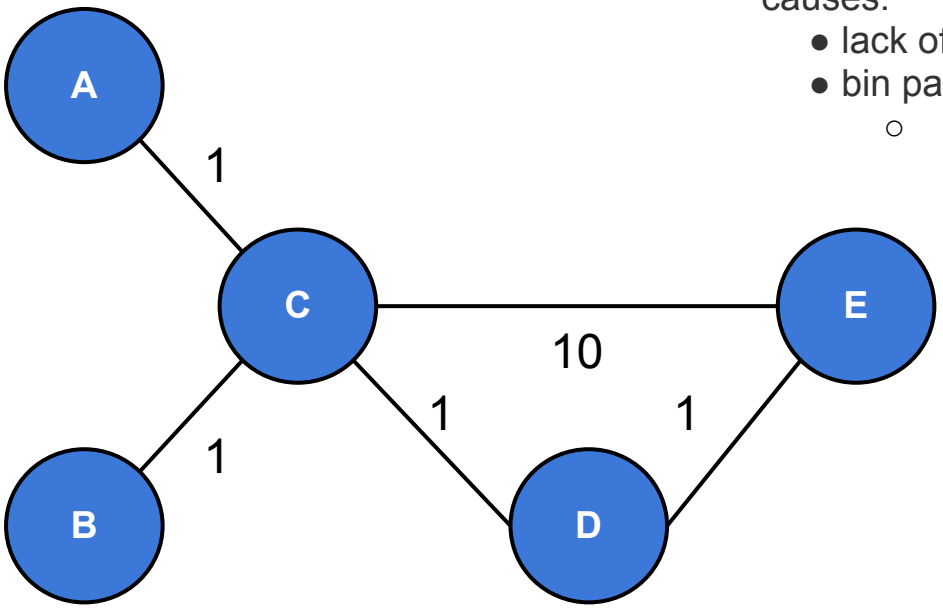


- lack of LSP level ingress admission control
  - would require another online or offline control mechanism
    - offline: need northbound API
    - online: back to autopbw issues
  - tension between overprovisioning level and transport elasticity

Link	Metric	Capacity
A-C	1	20
B-C	1	20
C-E	10	5
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	2
3	1	A	E	20

Bin Packing



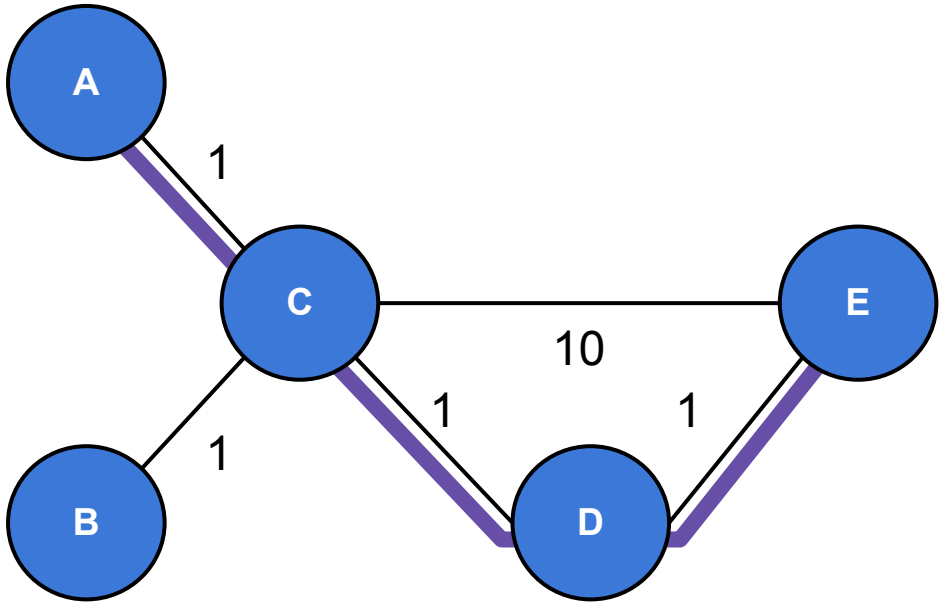
causes:

- lack of global LSP state
- bin packing is a sequencing problem - NP-Hard
  - Better to solve w/ some throughput optimization

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	5
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	5
2	2	B	E	10

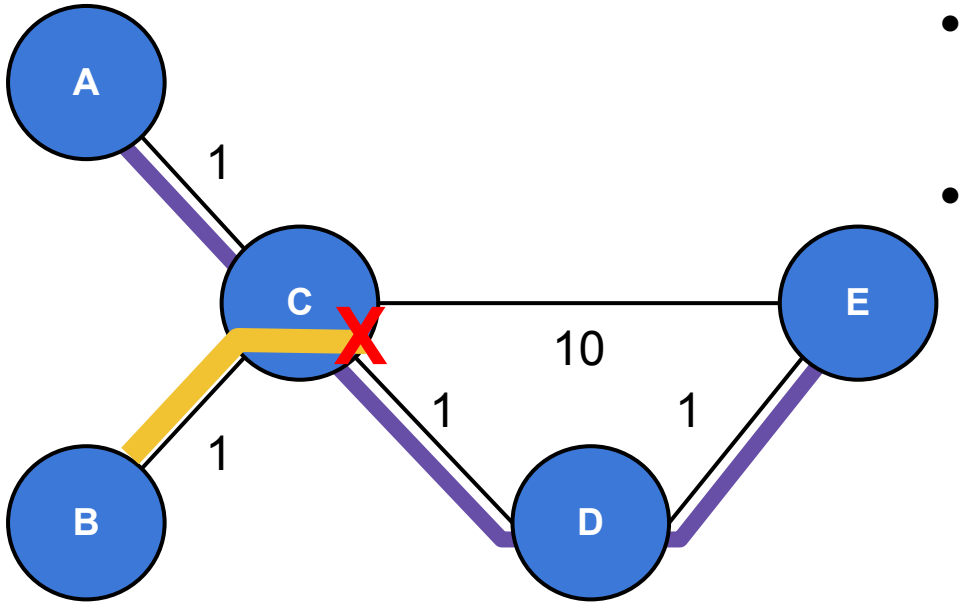
Bin Packing



Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	5
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	5
2	2	B	E	10

Bin Packing

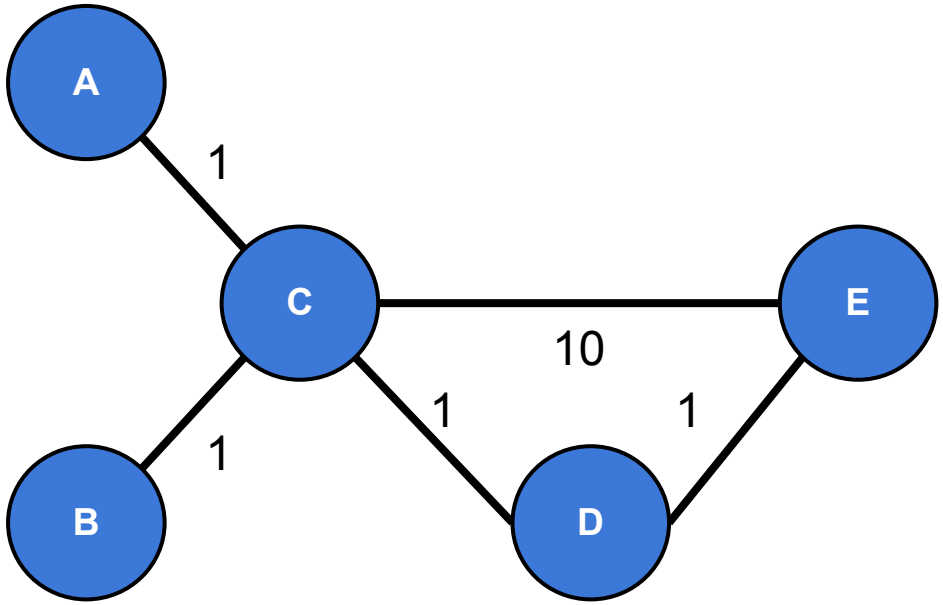


- unable to shuffle demands w/o
  - some offline control
  - stateful knowledge network LSPs
- 33% efficiency in capacity usage
  - efficiency dictated by order of event arrival

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	5
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	5
2	2	B	E	10

Scheduling



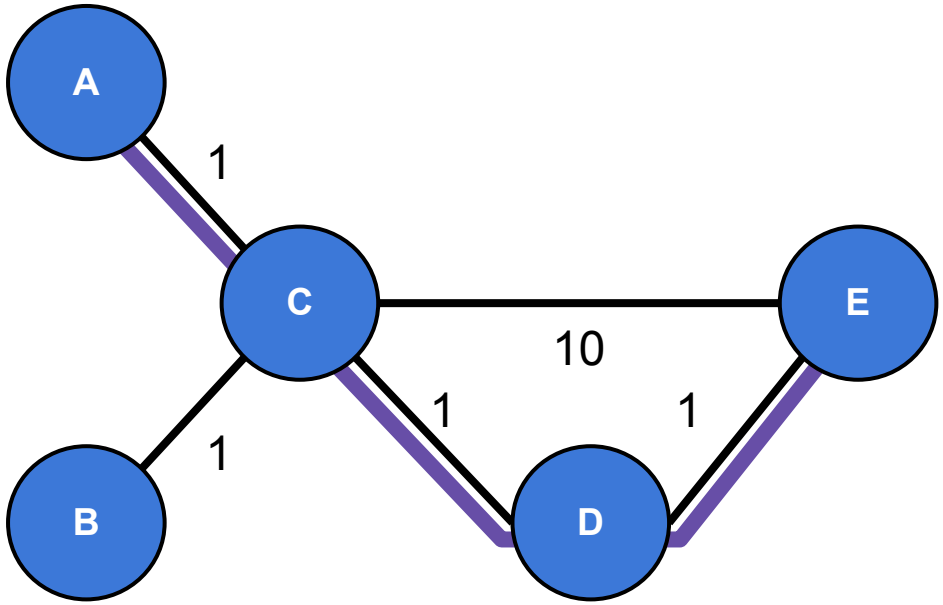
causes:

- autobw empirically derives demand with single period hysteresis
  - unable to use
    - historical timeseries
    - apriori knowledge of demand
  - network must be overprovisioned for either
    - offline: worst case demand over reopt interval
    - (⇔) online: (autobw) reopt trigger threshold + safety margin

Link	Metric	Capacity
A-C	1	20
B-C	1	20
C-E	10	10
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	7
3	1	A	E	7
3+k	1	A	E	7

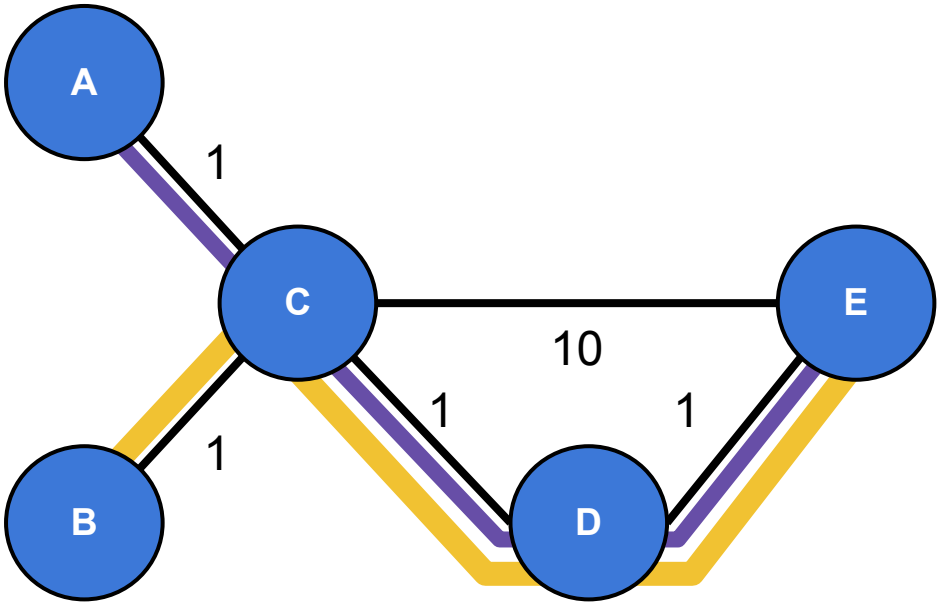
Scheduling



Link	Metric	Capacity
A-C	1	20
B-C	1	20
C-E	10	10
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	7
3	1	A	E	7
3+k	1	A	E	7

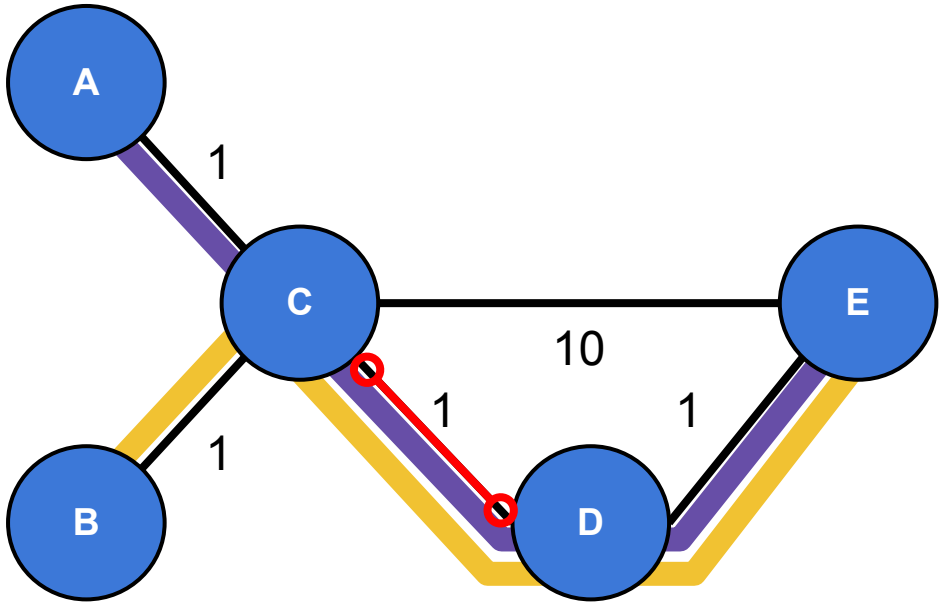
Scheduling



Link	Metric	Capacity
A-C	1	20
B-C	1	20
C-E	10	10
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	7
3	1	A	E	7
3+k	1	A	E	7

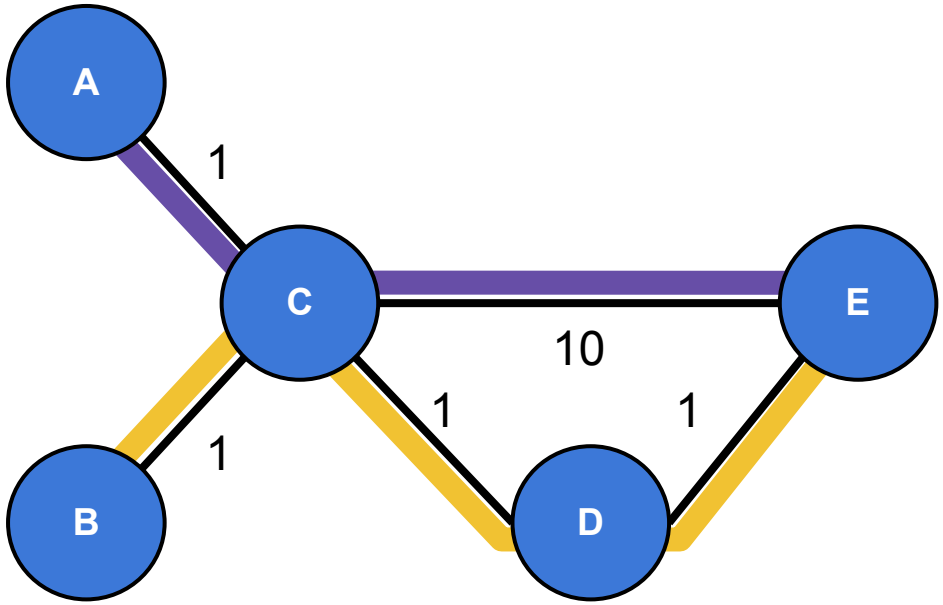
Scheduling



Link	Metric	Capacity
A-C	1	20
B-C	1	20
C-E	10	10
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	7
3	1	A	E	7
3+k	1	A	E	7

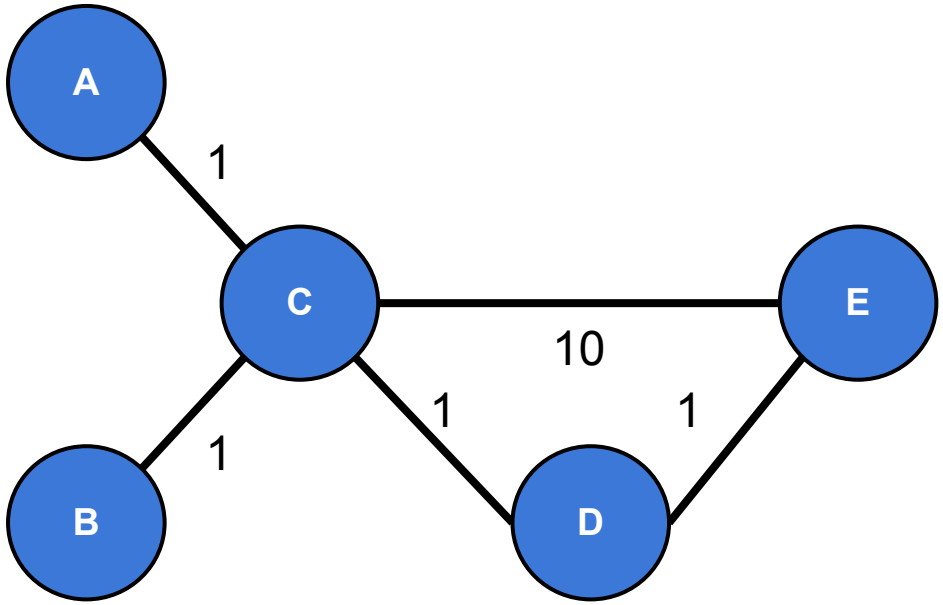
Scheduling



Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	10	10
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	2
2	2	B	E	7
3	1	A	E	7
3+k	1	A	E	7

Predictability



causes:

- routers act independently and asynchronously ⇒ path dictated by order of event arrival

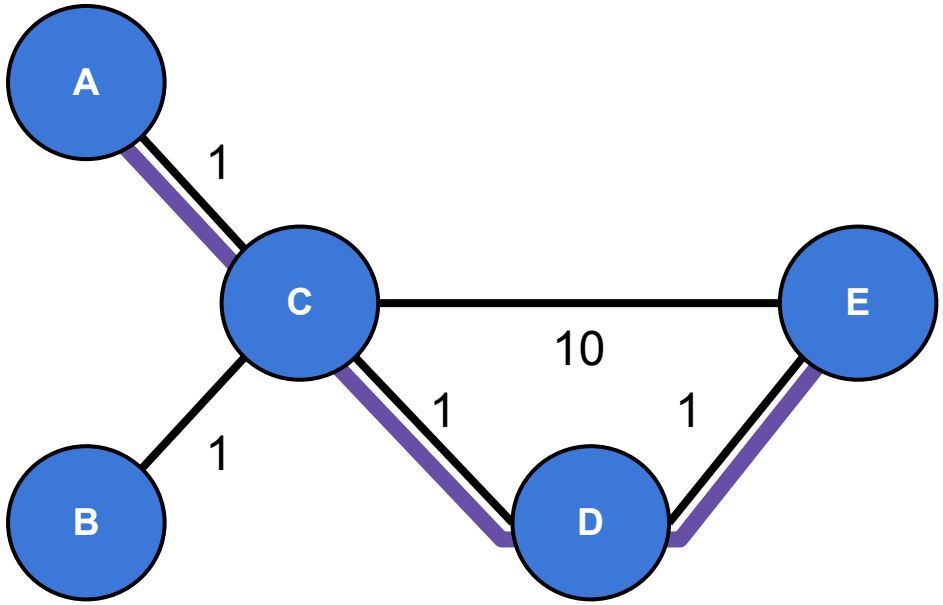
Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	1	10
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	7
2	2	B	E	7

VS

Time	LSP	Src	Dst	Demand
1	2	B	E	7
2	1	A	E	7

Predictability



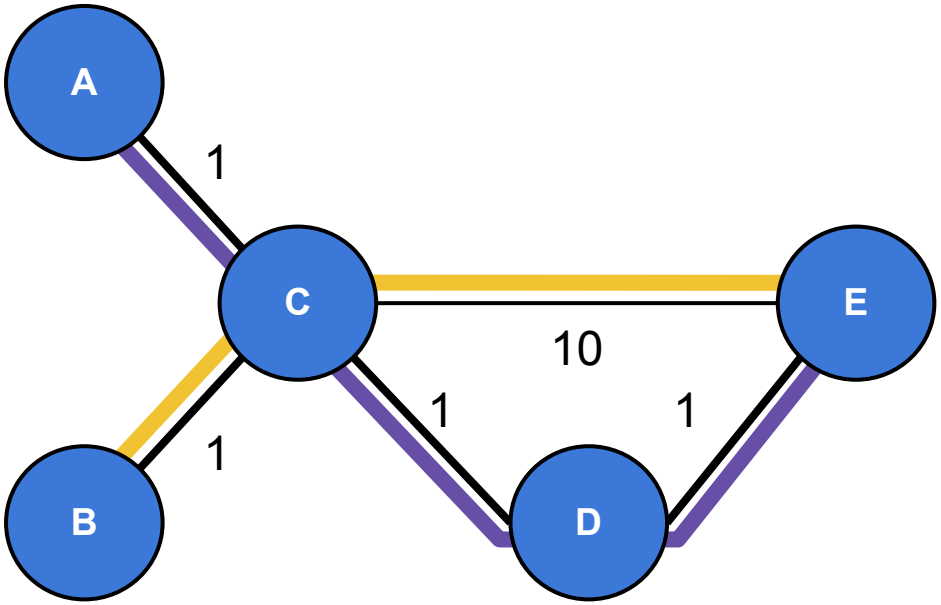
Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	1	10
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	7
2	2	B	E	7

VS

Time	LSP	Src	Dst	Demand
1	2	B	E	7
2	1	A	E	7

Predictability



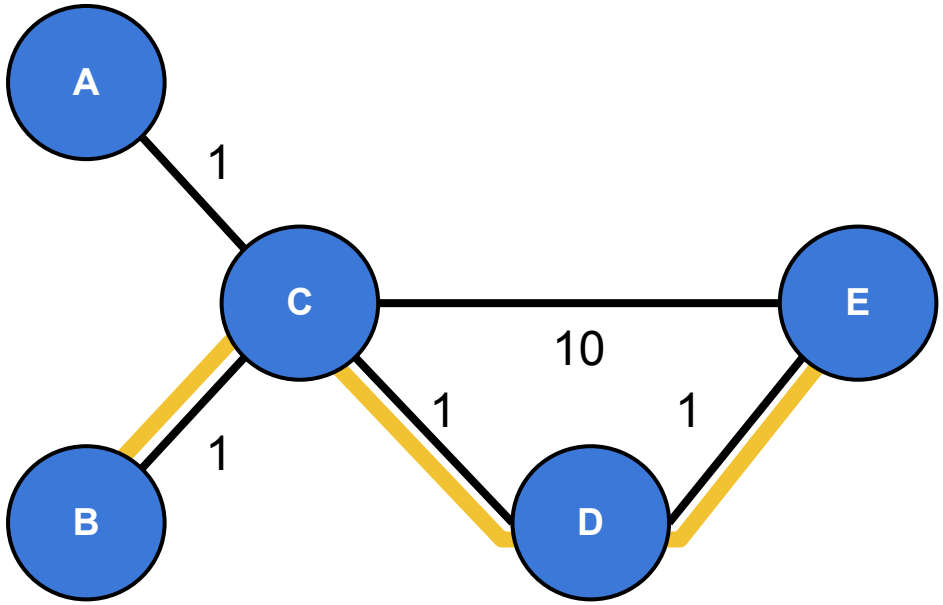
Time	LSP	Src	Dst	Demand
1	1	A	E	7
2	2	B	E	7

Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	1	10
C-D	1	10
D-E	1	10

VS

Time	LSP	Src	Dst	Demand
1	2	B	E	7
2	1	A	E	7

Predictability



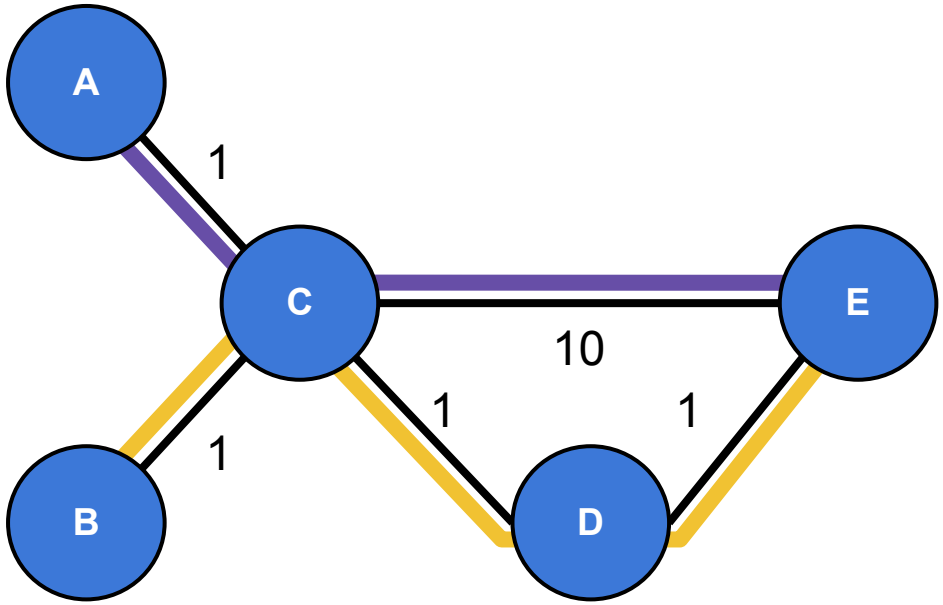
Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	1	10
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	7
2	2	B	E	7

VS

Time	LSP	Src	Dst	Demand
1	2	B	E	7
2	1	A	E	7

Predictability



Link	Metric	Capacity
A-C	1	10
B-C	1	10
C-E	1	10
C-D	1	10
D-E	1	10

Time	LSP	Src	Dst	Demand
1	1	A	E	7
2	2	B	E	7

VS

Time	LSP	Src	Dst	Demand
1	2	B	E	7
2	1	A	E	7

Topics

Google™

- SDN at Google today
- Example SDN Use Case: TE
- **Our SDN Experience So Far**
- Research Opportunities

Google SDN Experiences

Google

- Much faster iteration time: deployed production-grade centralized traffic engineering in two months
  - fewer devices to update
  - much better testing ahead of rollout
- Simplified, high fidelity test environment
  - Can emulate entire backbone in software
- Hitless SW upgrades and new features
  - Almost no packet loss and no capacity degradation
  - Most feature releases do not touch the switch
    - most state does not have to carried by network protocols

Topics

Google™

- SDN at Google today
- Example SDN Use Case: TE
- Our SDN Experience So Far
- **Research Opportunities**

SDN had been Around for Quite a While

Google

Ipsilon GSMP	1996
Cambridge's The Tempest	1998
IETF FORCES	2000
IETF PCE	2004
Princeton's Routing Control Platform	2004
4d Initiative	2005
Ethane	2007
Openflow	2008

SDN Opportunities

Google™

And yet all of SDN is in it's infancy:

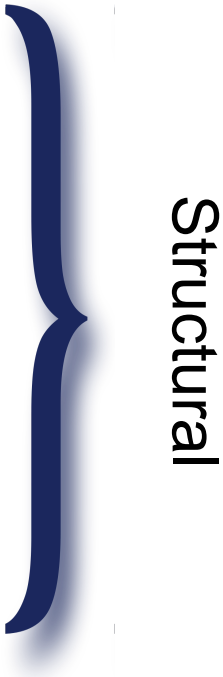
1. Controller ↔ Switch abstractions
   south-bound
2. Controller ↔ Application abstractions
   north-bound
3. Controller ↔ Controller abstractions
   east-west
4. Applications

SDN Opportunities

Google™

And yet all of SDN is in it's infancy:

1. Controller ↔ Switch abstractions
   south-bound
2. Controller ↔ Application abstractions
   north-bound
3. Controller ↔ Controller abstractions
   east-west
4. Applications



SDN South-Bound

Google

- **OpenFlow**: Still bare-bones but enough for initial production deployment with apriori knowledge of system capabilities
- **ForCES**: untested, no opensource implementation currently
- **PCEP**: low adoption currently
- **IRS**(???), many other less developed protocols.

All of these abstractions are lacking in expressiveness and/or adoption.

SDN North-Bound

Google

- What should the north-bound API look like?

- Should industry:
  - standardize?
  - wait for a de-facto controller to emerge with its own interfaces and an app store?

- policy
  - composition
  - decomposition
  - optimal state distribution

- Some researchers are tackling this problem
  - Stanford ONRC
  - Nick@(?): Procera
  - JRex@ Princeton: http://www.frenetic-lang.org/papers/

SDN East-West

Google™

- Inter-domain SDN...

SDN Applications

Google™

Having a centralized view allows new applications.

Many of these applications require novel research. A few of the most interesting to us are:

- Traffic Engineering
  - Intra-domain
  - Inter-domain egress
  - optimization
  - scheduling
  - control theory

- Security
- Event Based Control

Some Examples of Recent Google Research from InfoCom 2012:

Google

- *How to split a flow* by Tzvika Hartman, Avinatan Hassidim, Haim Kaplan, Danny Raz, and Michal Segalov

- *Upward max-min fairness* by Emilie Danna, Avinatan Hassidim, Haim Kaplan, Alok Kumar, Yishay Mansour, Danny Raz, and Michal Segalov (runner up for best paper)

- *A practical algorithm for balancing the max-min fairness and throughput objectives in traffic engineering* by Emilie Danna, Subhasree Mandal, and Arjun Singh

Conclusions

Google

- Despite it's relative immaturity, SDN is ready for real-world use
  - Google's datacenter WAN successfully runs on SDN (OpenFlow)
  - Enables rapid rich feature deployment

- Many Research Opportunities