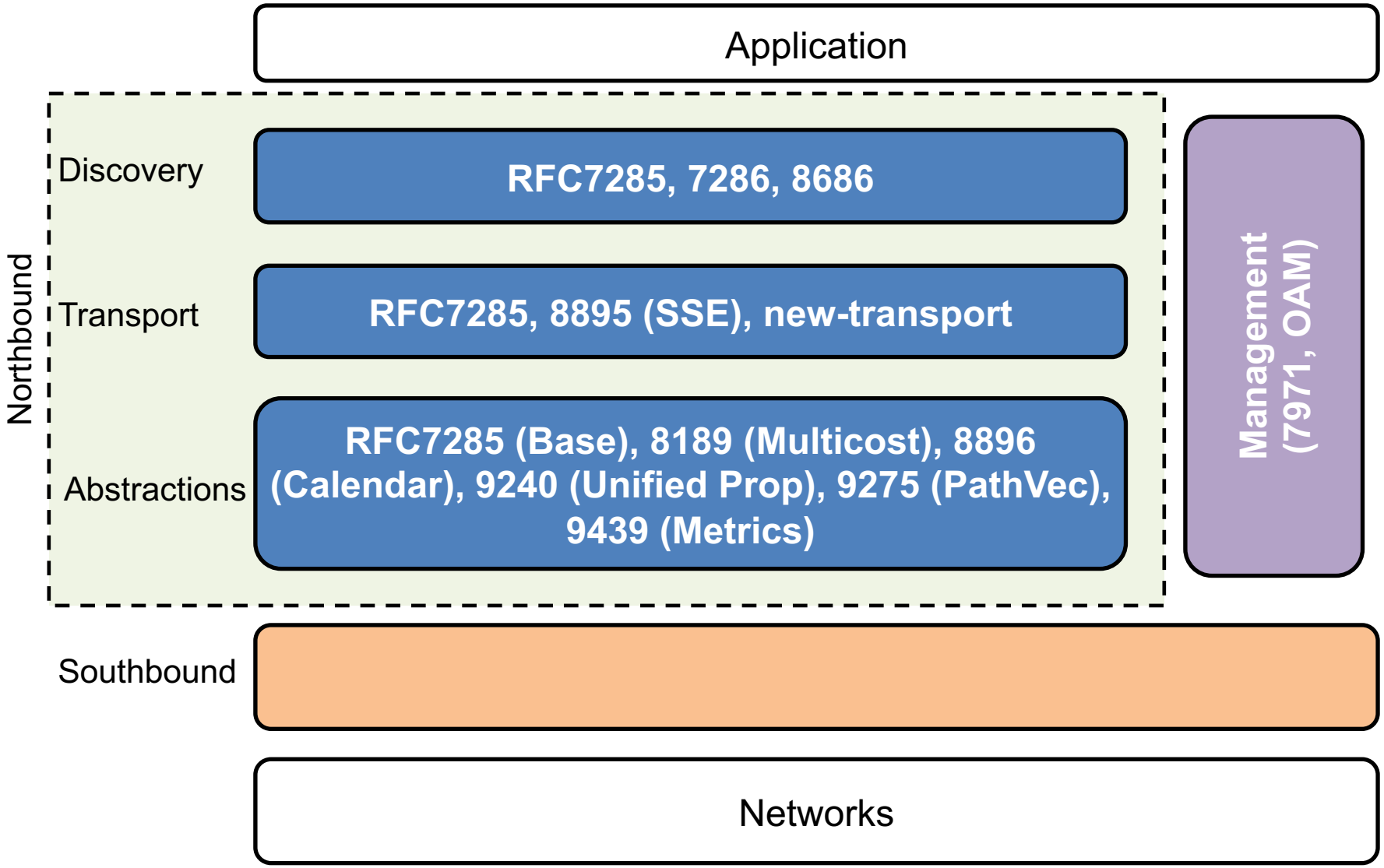
Southbound IETF Mechanisms Supporting Exposure of Network Info to Applications

data-sources, multidomain, cascading-alto

IETF 118 Side Meeting; November 8, 2023

Presenter: Danny Lachos, Sabine Randriamasy, Y. Richard Yang
on behalf of team

Problem Scope



- **Northbound**: What net info to be exposed
  - Clearly defined info [7285,9439]; clearly spec'd transport mechanisms [7285]; Reasonably understood how app may use the net info (e.g., [1,2])
- **Southbound**: how to obtain net info

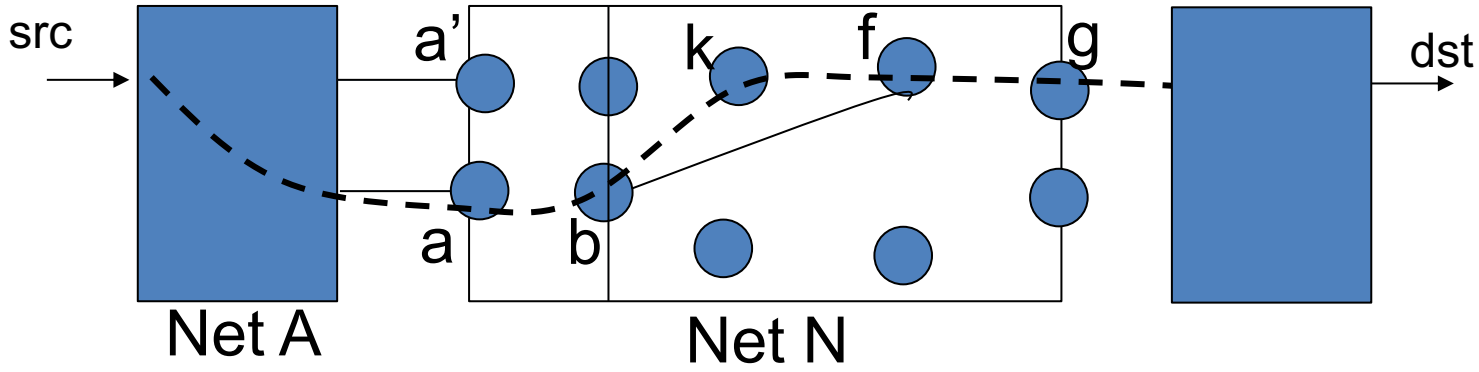
[1] PanDA [https://panda-wms.readthedocs.io/en/latest/advanced/brokerage.html]
[2] Rucio [https://github.com/rucio/rucio/blob/master/lib/rucio/core/topology.py]

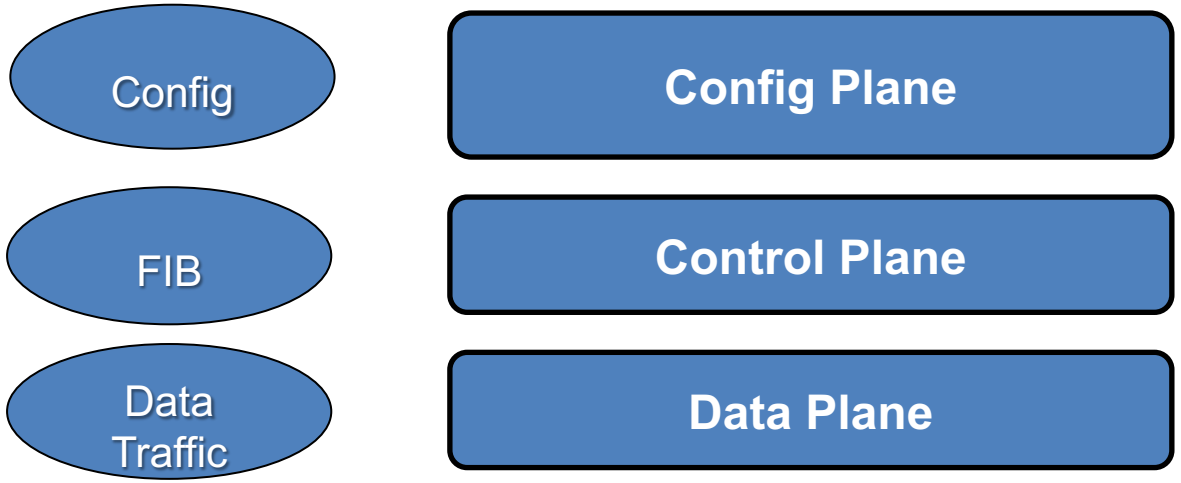
Specific Problem: Southbound Computing Network Path Prop

Path taken by <src-dst> is well-defined network prop

IGP LS shortest path

TE signaled path





Config analysis

Control plane mirror/monitoring

BMP (RFC7854); BGP-LS (e.g., RFC7752, 9085, 9351); YANG topology models (e.g., RFC8345/8795)

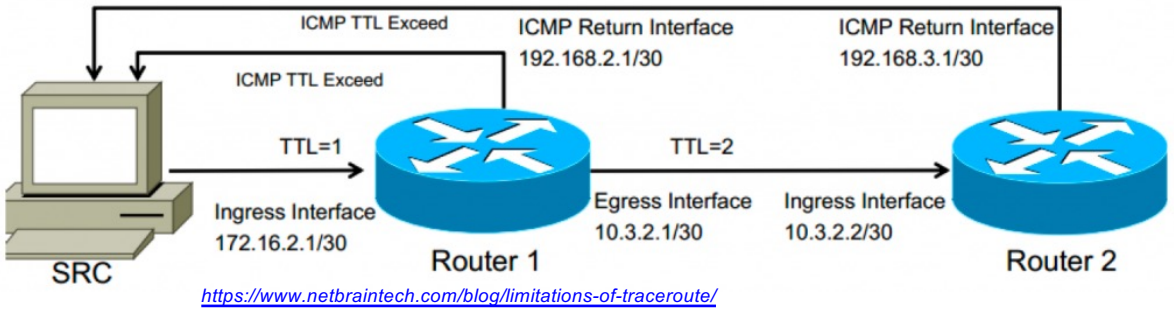
Measurements, probing

ICMP, IPPM, OWAMP

Existing IETF Mechanisms and Issues: Data Path Measurement/Sampling

- Base server using PerfSonar data
  - Based on IETF-defined ICMP (RFC792), OWAMP (RFC4656) mechanisms
  - Many correctness issues; privacy/access control issues

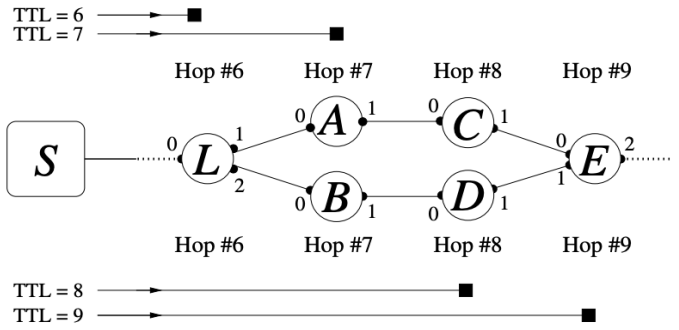
1979	davs://storm-ft.mi.infn.it	davs://storage01.lcg.cscs.ch	10	130.59.38.70	148.187.0.2
1980	davs://storm-ft.mi.infn.it	davs://storage01.lcg.cscs.ch	11	148.187.0.2	*
1981	davs://storm-ft.mi.infn.it	davs://storage01.lcg.cscs.ch	12	*	*
1982	davs://storm-ft.mi.infn.it	davs://storage01.lcg.cscs.ch	13	*	*
1983	davs://storm-ft.mi.infn.it	davs://storage01.lcg.cscs.ch	14	*	148.187.129.15
1984	davs://storm-ft.mi.infn.it	davs://storage01.lcg.cscs.ch	15	148.187.129.15	148.187.19.181
1985	davs://storm-ft.mi.infn.it	davs://tech-gftp.hep.technion.ac.il	1	192.135.14.76	131.154.254.2
1986	davs://storm-ft.mi.infn.it	davs://tech-gftp.hep.technion.ac.il	2	131.154.254.2	192.168.150.156
1987	davs://storm-ft.mi.infn.it	davs://tech-gftp.hep.technion.ac.il	3	192.168.150.156	193.206.128.17

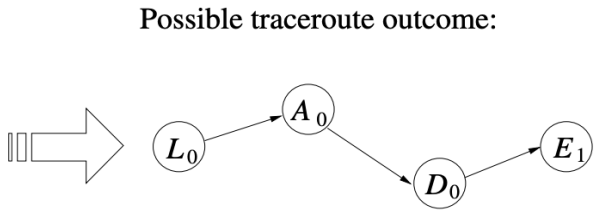


https://www.netbraintech.com/blog/limitations-of-traceroute/

juniper.net/documentation/us/en/software/junos/transport-ip/topics/topic-map/icmp.html

Node aliasing

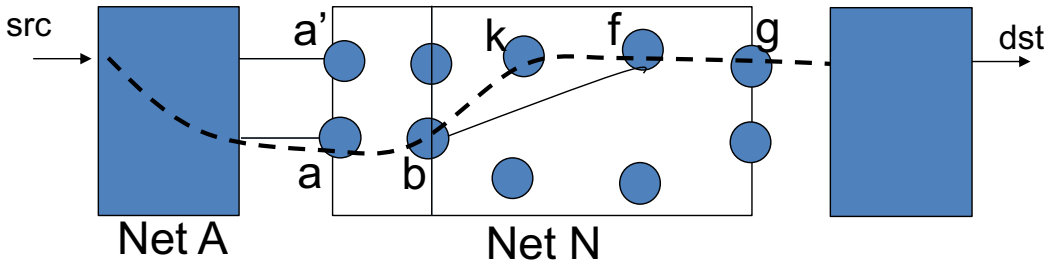


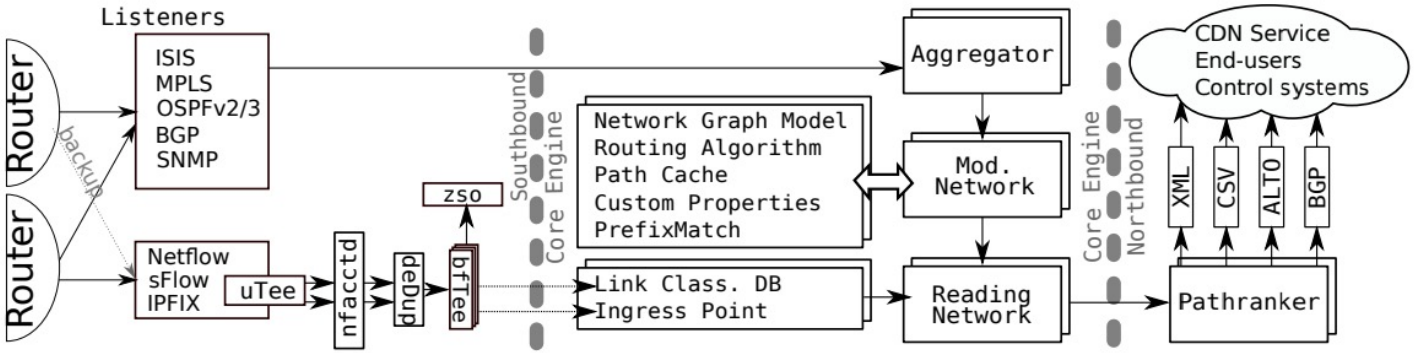


https://dl.acm.org/doi/pdf/10.1145/1177080.1177100

Path aliasing

Existing IETF Mechanisms and Issues: Routing System Listener





Flow Director Software Structure CoNEXT'19

- Sounds simple in theory, but can be complex, error-prune to build reliable system
  - Full state and computation replication of a network: partial or transformed state (e.g., BGP-LS) or computation can lead to mismatch, resulting in unknown "deviation"
    - Partial state can be OK to expose resources to be controlled
  - Upstream state/computation replication: existing approach uses xFlow/IPFIX sampling/report to detect upstream compute; must have traffic; scalable concern [1]

[1] slides-116-cascading-alto.pdf

Expected Work and Outcomes

- Evaluation
  - Evaluate existing IETF southbound mechanisms and deployment barriers, focus on evaluating computing well-defined network info
  - Expected outcome: Informational document organized list of issues, gaps
- Design filling gaps
  - Focus on designing minimal mechanisms completing current gaps, e.g.,
    - Expected outcome: Draft specifying cascading ALTO supporting end-to-end computation; draft specifying state/computation replication discovery and deviation indication
  - Explore clean building block designs, e.g.,
    - Expected outcome: Experimental design for efficient control plane output replication, not state/computation replication

Backup Slides

PanDA

- https://panda-wms.readthedocs.io/en/latest/advanced/brokerage.html

Network Weight

The network data sources are

- the Network Weather Service as the dynamic source, and
- the CRIC closeness as a semi static source.

Given the accuracy of the data and the timelapse from decision to action, the network weight only aims to provide a simple, dynamic classification of links. It is currently calculated as:

$$netWorkWeight = 0.5 \times (queuedWeight + throughputWeight)$$