TOPOLOGY EXCHANGE AND PATH FINDING

NSI-WG call 18/2/2015

R. Koning, M. Živković, S. Konstantaras, P. Grosso, C. de Laat (UvA)
F. Iqbal, F. Kuipers (TU Delft)

General Remarks

- We present the topology exchange solution that supports:
  - Different topology representations (NML just an example)
  - Different (optimal) path-finding algorithms are supported
  - (finding of) disjoint paths
  - Security (not discussed here)
  - Topology provisioning based on
    - requesting party
    - peering agreements
    - other policies
- Our topology exchange solution
  - has been implemented and deployed within Automated GOLE testbed as proof-of-concept
  - is currently extended to support OpenFlow domains
  - has been successfully demonstrated at SC 14

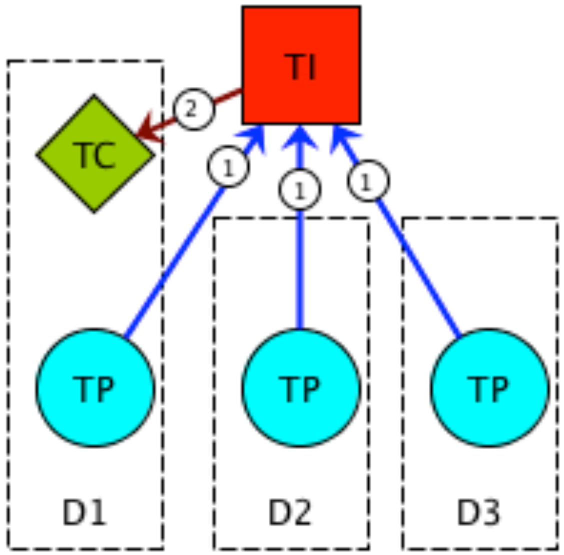
Components

- Three main components (can be implemented as services):
  - **Topology Index (TI)** stores the location of the served topologies
  - **Topology Provider (TP)** serves the topology files.
  - **Topology Consumer (TC)** processes the topology information
    - However, many versions of TCs can be used in parallel:
      - A Lookup service (for keeping an STP-Domain mapping)
      - A PathFinder service (for calculating inter-domain paths),
      - A Monitoring service (for monitoring changes of the topologies)
      - etc.

Considerations

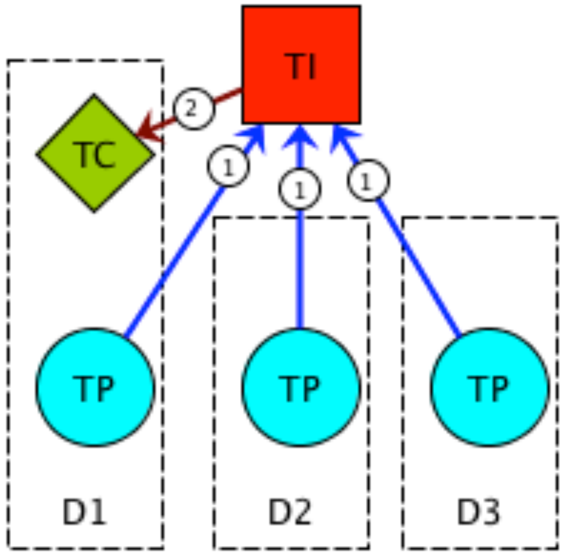
- The **Topology Index** is not the actual information source, those are the topology providers
- The **Topology Provider** deals directly with the consumer and presents topologies to the consumer based on local policies (decision point)
- The **Topology Consumer** uses obtained information and may use signed topology updates and encrypted connections

Topology Distribution (1/2)



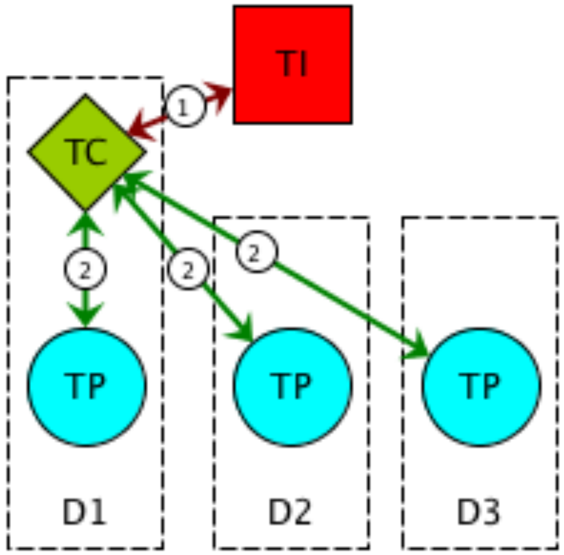
1. The Topology Providers send their updates to the Topology Index
2. The Topology Index notifies the subscribed topology consumers (clients)

Topology Distribution (2/2)



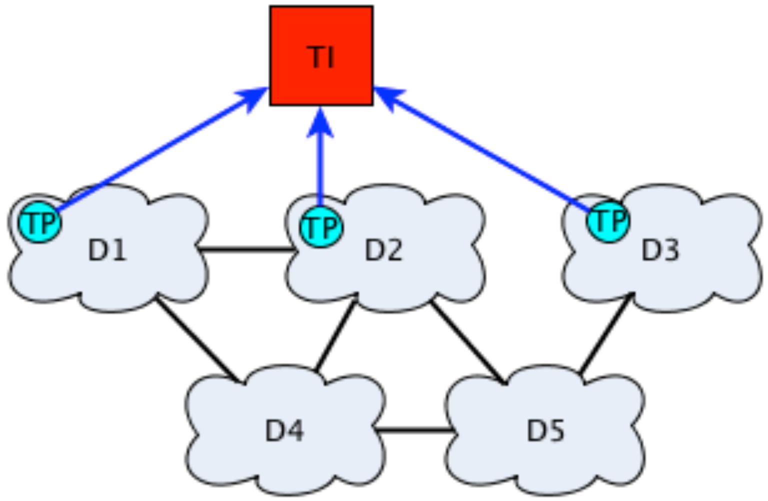
1. The Topology Consumer (client) fetches the summary information from the Topology Index
2. The Topology Consumer (client) obtains the topologies from respective providers

Topology Retrieval



1. The Topology Consumer (client) contacts the TI and looks for the latest updates on interested domains.
2. The Topology Consumer (client) all the TPs it is interested in and retrieves the topology files.

Index format example



Domain	Version	Location	Neighbours	Foreign domains
D1	01	http://d1.net/topo/	D2	D4
D2	02	http://d2.net/topo/	D1	D4, D5
D3	01	http://d3.net/topo/		D5

Neighbors and Foreign Domains

- Neighbors: a list of domains that are directly connected (have peering relationships) and report topology information to the TI.
- Foreign domains: a list of domains that have direct data plane connections to domains listed in the TI but do not report to TI
- The TI is responsible to process the updates of the TP and re-arrange the neighbors/foreign domains list.

TI: Synchronization and Fail Over

In case of TI fail over:

- Degradation of performance is expected but
- Topology information <u>is</u> available (TPs) and retrievable

- TI replication can solve the problem
  - Example: A TC which plays the role of a TI for another TC
  - When a conflict arises TI can request from TPs to resend their summary information
  - TCs need to be aware of this backup server

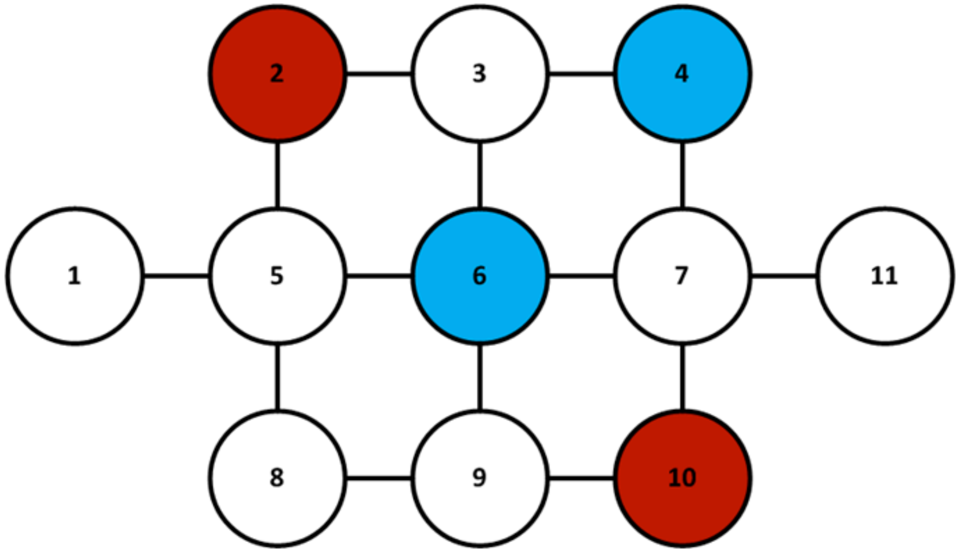
Security concerns

- We use public key techniques to validate topology information
  - Topologies and topology updates are signed by the TP
  - Index Information is signed by the TI
- Public keys have to be known by all parties. This can be achieved by:
  - Distributing public keys via a PKI
  - Managing the Topology Index, adding domains and keys manually
  - Use DNS to distribute keys and DNSSec to sign

Path finding

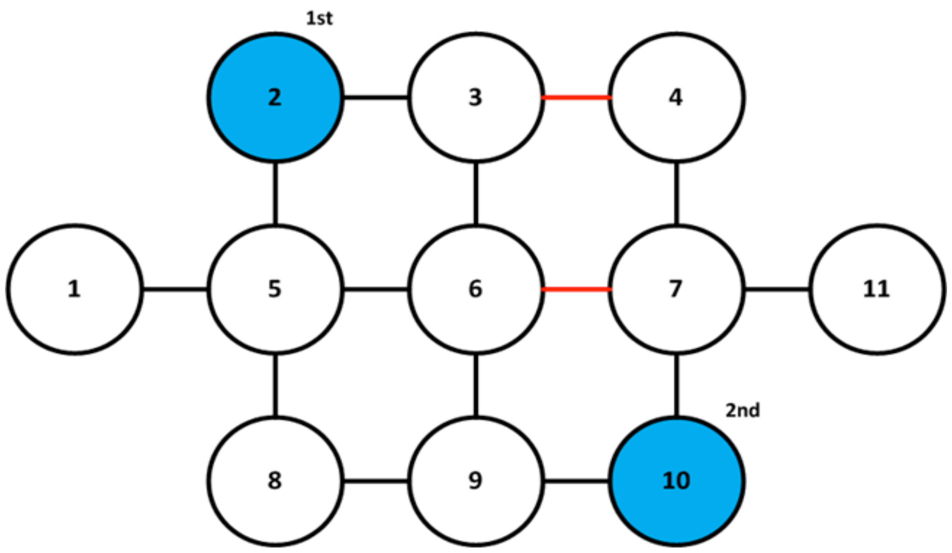
- The multi-domain routing algorithm
  - Provides an inter-domain path that satisfies a multiple of path requirements
  - Inter-domain links may be described using many attributes
  - Multi-constrained (optimal) path problem
  - May or may not support loops (adaptations)

Path requirements



- For an **Inter-Domain (ID) path,** the following requirements may be specified:
  - Certain <u>domains</u> must (or must not) <u>belong</u> to the ID path
  - Certain <u>domains</u> or <u>ID links</u> must be <u>**in**</u> a predefined <u>sequence</u>
  - Certain ID <u>links</u> must (or must not) <u>belong</u> to the ID path

Example



- Find the <u>shortest inter-domain path</u> from domain 1 to domain 11, "<u>not-via</u>" inter-domain <u>links</u> (3,4) and (6,7), and "<u>in-order</u>" <u>domains</u> 2,10.
  - The answer is (1-5-2-3-6-9-10-7-11)

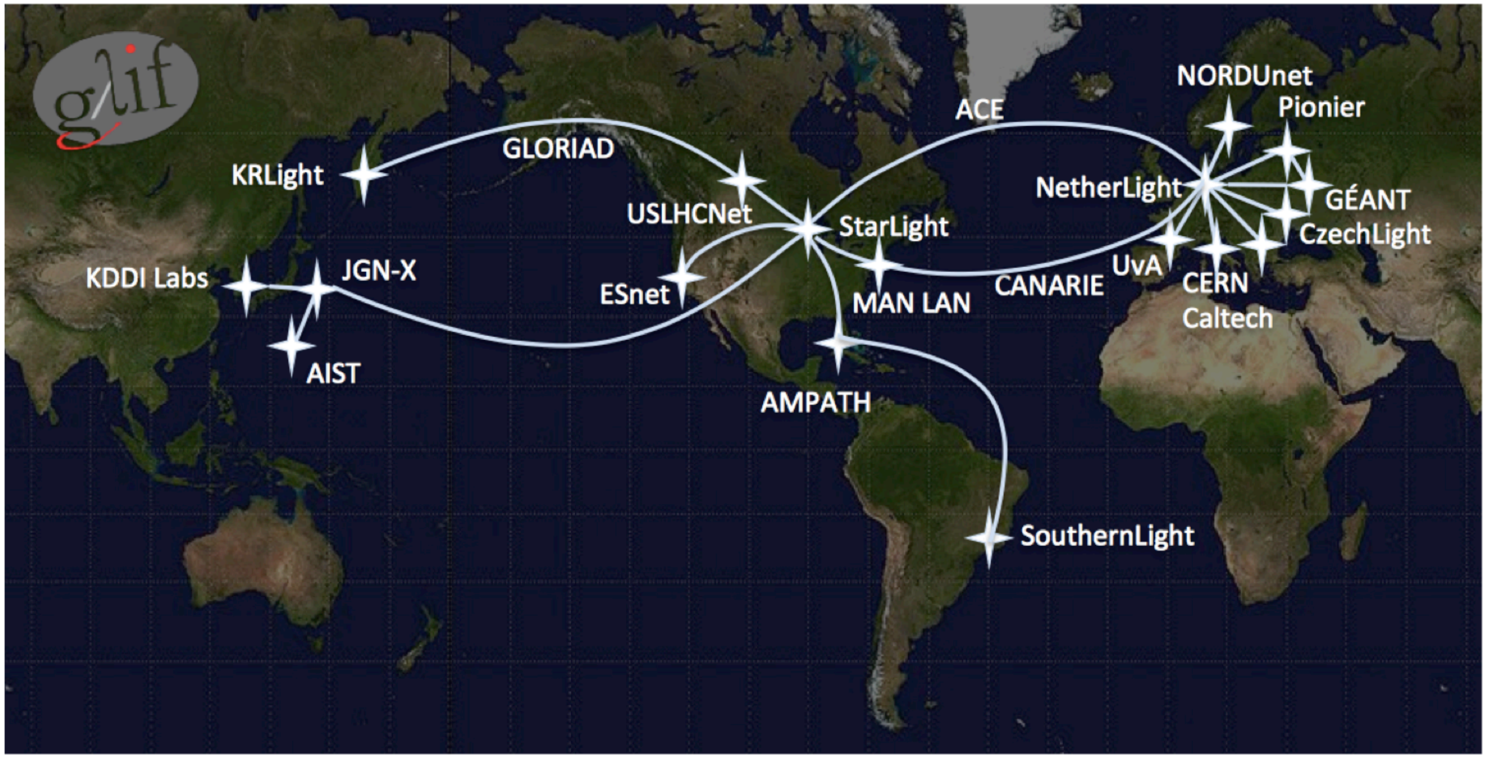
Implementation

- Input
  - A network of domain and inter-domain links.
  - The lists of path requirements (five in total)
- Output
  - A list representing the path that fulfills the requirements.

- An exact algorithm based on k-shortest paths approach, that utilize a look-ahead concept and a path comparison routine to speed up the process.

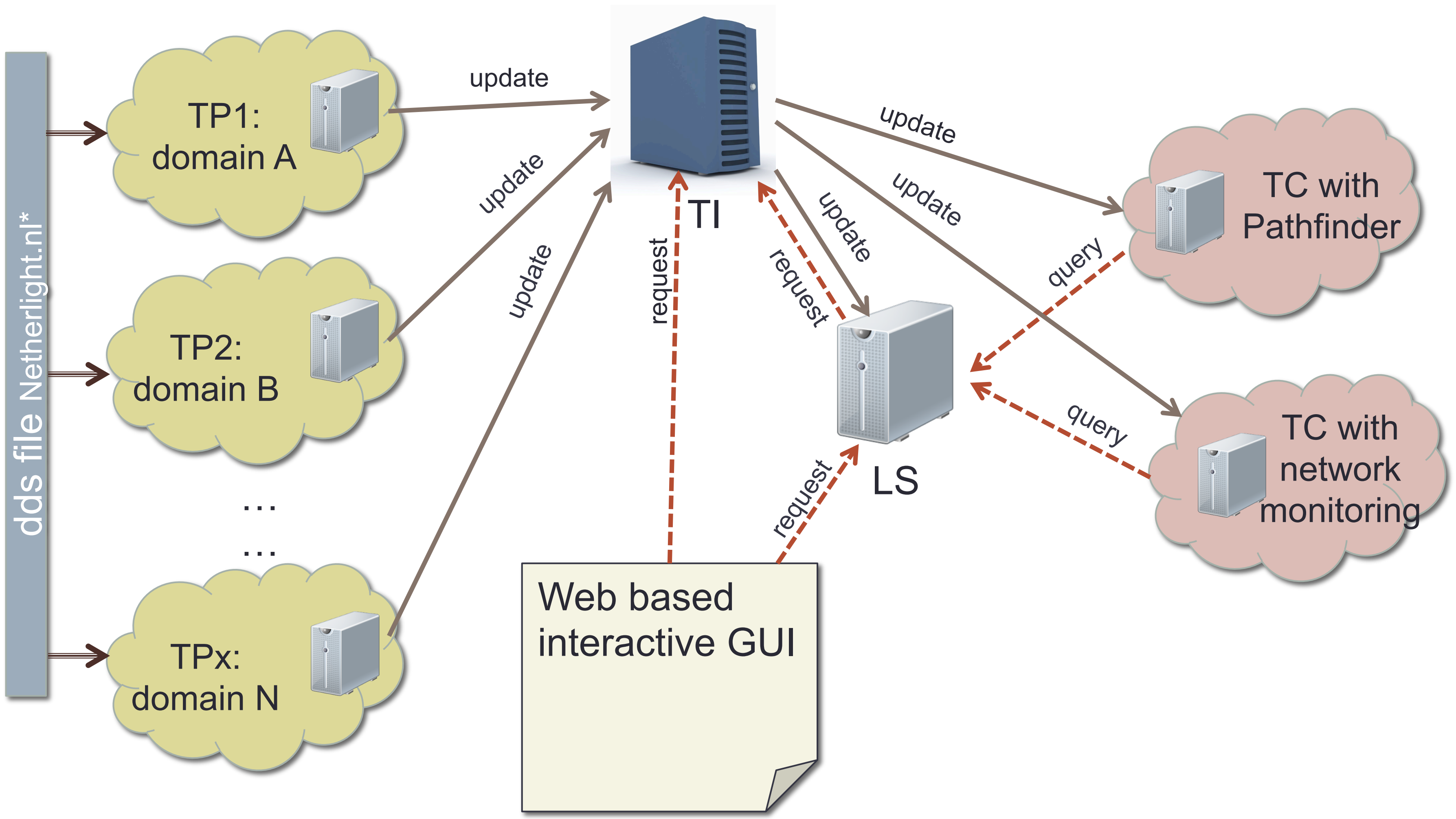
- Implemented in Python.

Use case – Automated GOLE



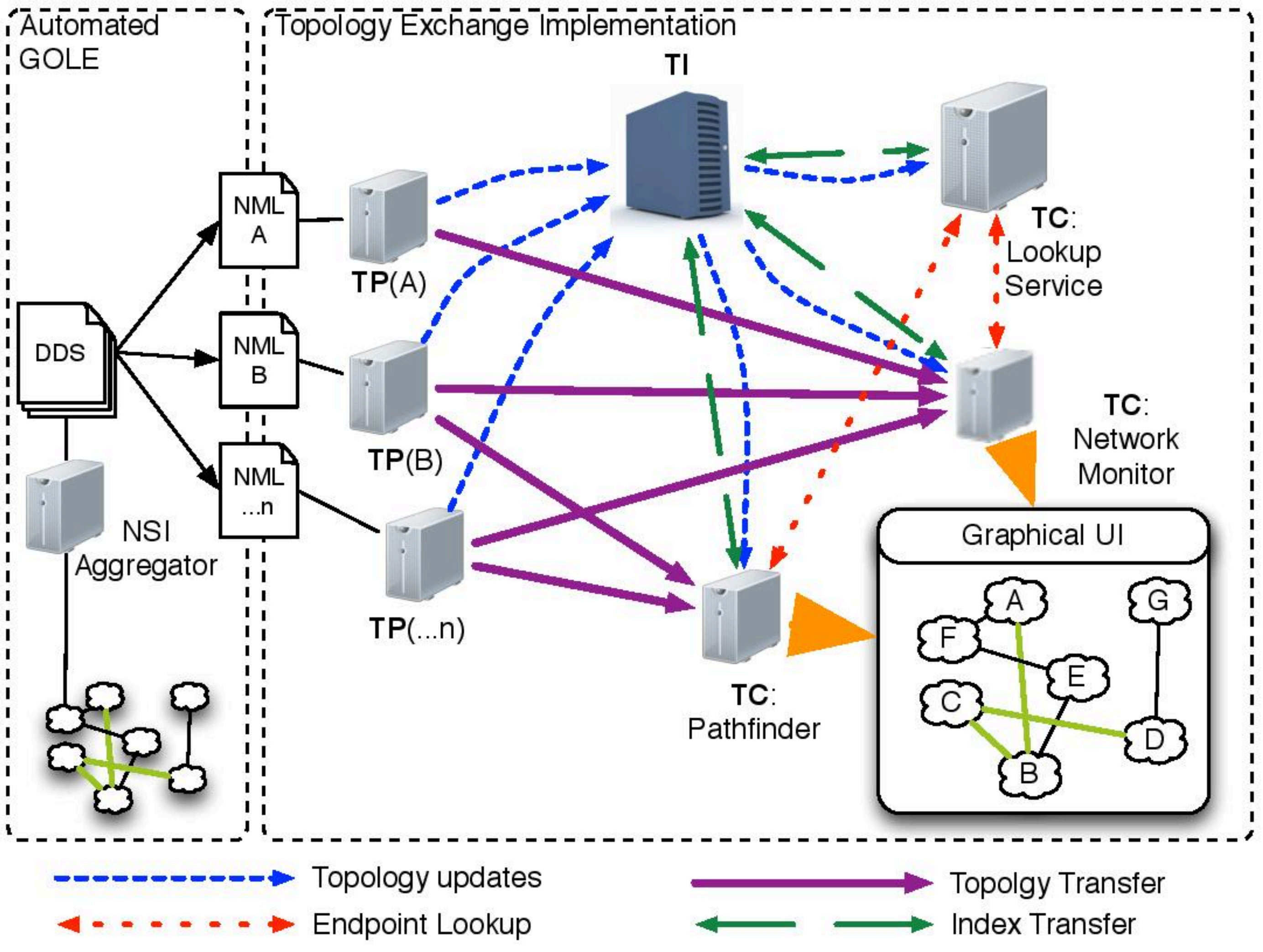
The Automated GOLE was moved into production on Friday September 12, 2014.
We do expect further necessary improvements throughout the fabric and need user

Architecture implementation (SC14)

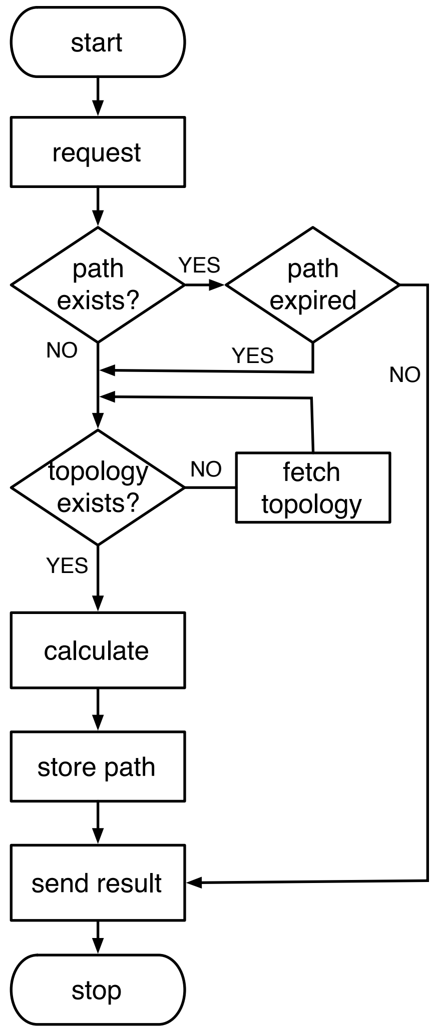


*https://agg.netherlight.net/dds/

Architecture implementation (SC14)



Path finding details (SC 14)



Advantages and drawbacks

- Advantages
  - Simple, easy to re-use components (do one thing and do it well)
  - Uses real topology information from the Automated GOLE
  - NSA implementation independent
- Drawbacks
  - Security has not been implemented yet
  - A full API needs to be defined

Future work

- Topology Index replication methods
- Using foreign domains information
- Inclusion of dynamic link information (e.g. actual bandwidth, utilization, availability)

Thank you