Authenticated Setup of Virtual Links with Quality-of-Service Guarantees

Roland Bless, Martin Röhricht, Christoph Werle
Institute of Telematics, Karlsruhe Institute of Technology (KIT)
Motivation

- Network Virtualization is an enabling technology
- Easier deployment of global networks and services
  - Homogeneity across provider domain boundaries
- Parallel operation of different network architectures
  - deploy novel network architectures and E2E services without requiring Internet-wide consensus
- Increased flexibility
  - On-Demand creation and modification of virtual network topology and resources, esp. nodes and links
  - Resource migration as Traffic Engineering mechanism
  - More efficient use of resources (exploit statistical multiplexing gain)
Network Virtualization

Virtual Network (VNet)

- Set of (virtual) nodes directly connected by (virtual) links (realized on top of a set of physical resources, the “substrate”)
- „Naked“ topology at layer 3
- No assumptions about the network protocols or architecture running inside the VNet, i.e., not necessarily IP
- May use various substrate techniques to create virtual links, e.g., IP Tunnels, MPLS, Ethernet VLANs,…
- We assume an IP-based substrate

Partitioning or aggregation of resources possible
Network Virtualization Business Model

Virtual Network Operator (VNO)

Virtual Network Provider (VNP)

Virtual Network

Substrate Networks

Infrastructure Provider A

Infrastructure Provider B

Infrastructure Provider C

Setup of Virtual Links
Setup of Virtual Links with QoS

- Isolation and **QoS guarantees** required
- Need to reserve resources along a substrate path
- Combine **resource reservation** with virtual link setup

![Diagram of Virtual Node Architecture]

- Virtual node
- Substrate node
- Virtual Link
- Physical Link

Infrastructures Provider InP₁ and InP₂

A

QoS signaling

R. Bless "Authenticated Setup of Virtual Links w/ QoS guarantees"  
ICCCN 2011, Maui, Hawaii
**Approach**

- Use existing **QoS resource reservation protocol** of the NSIS framework **QoS NSLP**
- Need interoperable solution for link setup across provider (InP) domains
- Add information object for setup of virtual links
- Add **security** object
  - Authentication (Pre-Shared Key)
  - Integrity protection for NSLP msgs (HMAC)
Step by Step Example

Shows unidirectional resource reservation VM1 → 2
Bidirectional reservation is possible
**Detailed Message Sequence with GIST**

Router A

<table>
<thead>
<tr>
<th>GIST Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIST Confirm</td>
</tr>
<tr>
<td>GIST Data [RESERVE + VLSP object]</td>
</tr>
</tbody>
</table>

1. Reserve Resources
2. Install virtual link

Router X

<table>
<thead>
<tr>
<th>GIST Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIST Response</td>
</tr>
<tr>
<td>GIST Confirm</td>
</tr>
<tr>
<td>GIST Data [RESERVE + VLSP object]</td>
</tr>
</tbody>
</table>

1. Perform Resource Admission Control
2. Pre-reserve Resources
3. Forward RESERVE

Router B

<table>
<thead>
<tr>
<th>GIST Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>GIST Response</td>
</tr>
<tr>
<td>GIST Confirm</td>
</tr>
<tr>
<td>GIST Data [RESPONSE + VLSP object]</td>
</tr>
</tbody>
</table>

1. Commit Resources
2. Forward RESPONSE

1. Reserve Resources
2. Install virtual link

**GIST 3-way handshake**

Send a RESERVE

Install GIST state

Perform Resource Admission Control
Evaluation Setup

- How long does it take to setup a virtual link, incl. QoS guarantees?
- Used freely available NSIS implementation (C++) [http://nsis-ka.org/] → evaluation code is available!
- Linux, KVM-based VM, Xeon X3430 Quad-core@2.4GHz, GRE Tunnel

![Diagram of network setup]

VLink₁ (Ethernet over GRE tunnel)
Measurement Methodology

- Measurement points in the code
- tcpdump packet capture on all nodes tb1 – tb4

GIST Query → GIST Response → GIST Confirm → RESERVE

Execute script for GRE tunnel setup

RESPONSE
Round-trip time \(tb1 \rightarrow tb4\): 0.7ms

- External program triggers virtual link setup
  - Includes inter-process communication
- Script execution for virtual link setup dominates
Pure NSIS Signaling

Intermediate node processing <1ms
Teardown Duration

- Link teardown takes much longer than setup, presumably due to “still in-use” checks
- Teardown not so critical (compared to setup)
Signaling Authentication Overhead

- Subtracted script execution for virtual link setup
- No significant overhead if security is used
Authentication Overhead

- Additional SessionAuthorization object [RFC5981]
  - Protects RESERVE and RESPONSE messages
  - Added 104 bytes to message (VLSP object: 80 bytes)
- HMAC calculation is negligible
Conclusion and Summary

- Combining QoS reservation and virtual link setup is useful and efficient.
- Extension of an existing NSIS signaling protocol was easy.
  - Additional VLSP object is ignored by intermediate nodes, but will perform QoS resource reservation.
  - Local link setup within nodes is much more costly than pure signaling and admission control processes.
- Securing the signaling is important and can be done without significant overhead.
- Currently: extend approach by node setup.
Thank you!