Demonstrating Distributed Virtual Networks

Alexandru Paler, Andreas Fischer, and Hermann de Meer

University of Passau, Passau 94032, Germany,
{alexandru.paler, andreas.fischer, hermann.demeer}@uni-passau.de

Keywords: future internet, virtual networks, virtual network deployment

1 Introduction

The current internet is commonly perceived as being too inflexible. Network Virtualization has been recognized lately as a method to overcome these limitations. Previous implementations of Virtual Network (VN) deployment software either provided limited access to the network layer (like PlanetLab), or focused only on specific application scenarios (like VNUML or VLAN). A first step toward the expected flexibility, with a solution that is both universal and thorough, is the implementation of network virtualization using system virtualization approaches [2].

We present a system that is constructing an underlying VN in a universal, scalable and flexible way. This was developed as part of the Autonomic Internet project[3]. A video demonstrating the software can be found at [1].

2 Controlling Distributed Virtual Networks

This demonstration features a software for the on-demand deployment and management of VNs distributed on physical infrastructures. Applications running on top of it are supported transparently and have full access to the VN layer. From a hardware perspective, a VN is composed of virtual routers (VRs) and virtual links (VLs). The routing service (RS) supported by the VN can also be viewed as an intrinsic component.

The software architecture is based on two loosely coupled components. The first component translates the description of a VN topology into a set of commands. The commands are then forwarded to and processed by the second component, and as a result, a VN is constructed or modified. The description of a VN contains names for both VRs and VLs. Moreover, the VLs can be configured, allowing the user to assign network addresses. The software is designed to be protocol agnostic (e.g. IPv4, IPv6, ATM), making future extensions possible.

The chosen physical test bed consists of four physical hosts interconnected with Gigabit Ethernet. The current version is using XEN for the VRs, OpenSSH for supporting the VLs and Quagga as a basis for the RS. The functionality of the software is demonstrated by constructing a VN with a linear topology distributed across all the physical hosts of the test bed. We use this particular virtual topology because it eases the visual presentation, but this fact does not limit the generality of our software.

To construct a VN, the following steps are executed on each physical host:

1. Receive a complete set of commands
2. For each VR command start the VR_i and the corresponding RS_i
3. For each VL command (between VR_1 and VR_2)
   (a) Add a virtual network interface to the VR_1
   (b) Connect to the remote physical host to create a link-segment
   (c) Add a virtual network interface to the VR_2
   (d) Create a link (tunnel) between VR_1 and VR_2
   (e) Instantiate VL_i and configure the RS in VR_1 and VR_2

A VL is constructed between two existing VRs and consists of three segments: one is connecting the physical hosts, and the other two are connecting each VR with its host. All three segments are aggregated to a VL by two software bridges. If the linked VRs are hosted on the same host, then Step 3b is optional.

Once a VN is built, third-party applications can monitor it by using a request/response mechanism implemented in the software. Collecting monitoring data about the infrastructure helps taking management decisions like the migration of VRs (together with corresponding VLs) or the change of topology. Such decisions can increase the resilience of the VN in the face of faults or attacks. The modification of topology is transparently affecting the RS, thus leading to an automatic reconfiguration.

3 Conclusion

A major goal of the Future Internet is to enable very flexible and scalable topologies. The presented software proves this to be possible. Future work will be targeted on improving the performance and the responsivenes to live-migration scenarios.

Acknowledgements The research leading to these results has received funding from the European Community's Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) in the context of the AutoI project, grant agreement no. 216404.

References

1. Distributed Virtual Networks - Video Demonstration (August 2010), `http://www.net.fim.uni-passau.de/demo_ghent`
2. Fischer, A., Berl, A., De Meer, H.: Virtualized networks based on system virtualization. In: Proc. of the 2nd GI/ITG KuVS Workshop on The Future Internet (2008)
3. Rubio-Loyola, J., Astorga, A., Serrat, J., Chai, W.K., Mamatas, L., Galis, A., Clayman, S., Cheniour, A., Lefevre, L., Mornard, O., Fischer, A., Paler, A., de Meer, H.: Platforms and Software Systems for an Autonomic Internet. In: Proc. IEEE Global Communications Conference (GLOBECOM 2010). Miami, USA (December 2010)