On Creating Overlay Routing Topologies between Heterogeneous Experimental Facilities

Christian Henke[1], Robert Wuttke[2], Tanja Zseby[2], and Konrad Campowsky[2]

[1] Technical University Berlin
Chair Next Generation Networks
Str. des 17. Juni 135, 10623 Berlin, Germany
c.henke@tu-berlin.de
http://www.av.tu-berlin.de
[2] Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
surname.lastname@fokus.fraunhofer.de

Abstract. Numerous Future Internet initiatives around the world establish experimental facilities that enable researchers to run their experiments in real world conditions. Through virtualization technologies, researchers have access to a large number of resources to run their experiments. The facilities provide different resources (e.g. sensors, end-hosts, routers), virtualization methods and access policies (private, community-based shared, federated) to accommodate a wide range of experiments. Nevertheless, for some experiments it is necessary to use resources across testbeds. Today, the support for integrating resources in one common routing topology is missing. In this paper we discuss use cases where a routing overlay over different heterogeneous testbeds is needed and present an implementation of a routing overlay mechanism to integrate nodes from Planetlab, VINI, and G-Lab. We identified the need for common resource federation mechanisms and tools that ease the setup of experimental facility resources across administrative domains and across different facilities.

Keywords: experimental facility, routing overlay, routing experiments, packet tracking, future internet, federation.

1 Introduction

Several national and international Future Internet initiatives (e.g. GENI, FIRE, APAN, G-Lab, AKARI) build experimental facilities for researchers to deploy and test novel algorithms, protocols, applications, and network architectures. The goal of such initiatives is to bring innovative and radical Future Internet research approaches from theory to practice. One of the key technologies that enables affordable experimental research facilities is resource virtualization, allowing researchers to run separated experiments on the same physical substrate in parallel.

T. Korakis et al. (Eds.): TridentCom 2011, LNICST 90, pp. 162–171, 2012.

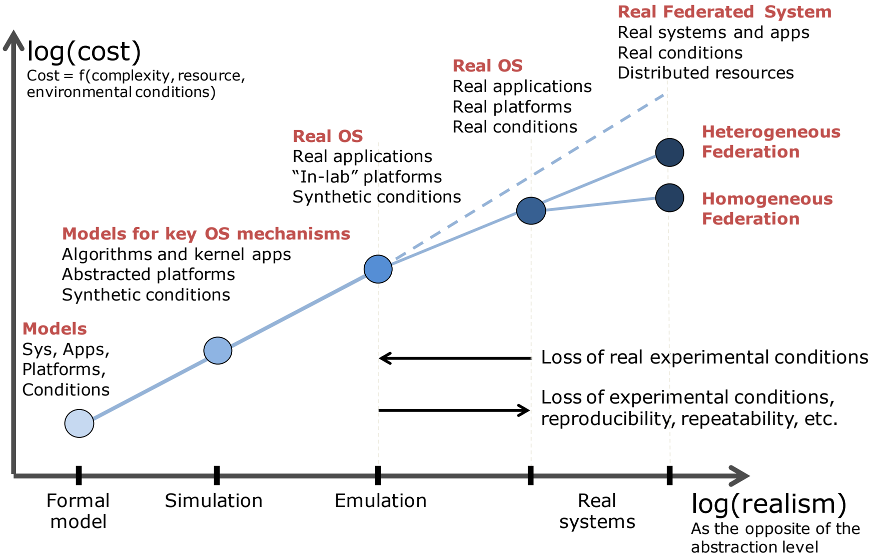


Fig. 1. Experimental Research and the Motivation for Federation [3]

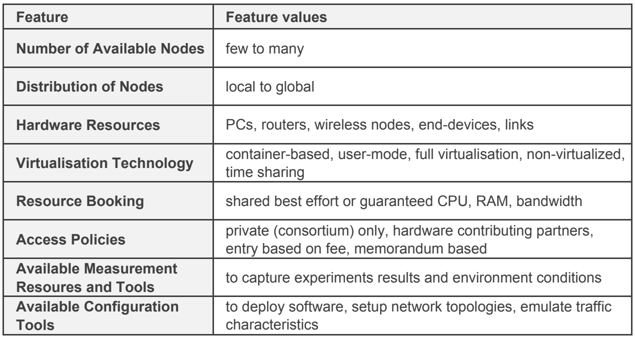
By closing the gap between simulation and real world experiments, experimental facilities provide an important step in the natural evolution of ideas from a formal model to a tested and validated solution. Figure 1 shows this concept. Unfortunately, by increasing the realism of experiments, the associated costs increase as well. Federation aims at re-using existing infrastructure and thus decreasing the overall costs for large scale experiments.

Through several projects funded by Future Internet initiatives, a number of experimental facilities have been set up. However, such facilities have different characteristics in terms of number and distribution of nodes, available hardware, virtualization technology, access policies, deployment frameworks, and measurement/monitoring capabilities (see Table 1).

This paper contributes in the following areas:

1. We show that the current facilities and resource federation frameworks lack the possibility to configure routing topologies required by our experiments.
2. We demonstrate our routing topology mechanism that spans multiple heterogeneous experimental facilities and share experiences while setting up the topology.
3. We show the feasibility and usefulness of routing overlays for a broad spectrum of experiments and motivate the support of such mechanisms by large scale federated facilities.

Table 1. Characteristics of Experimental Facilities



Feature	Feature values
Number of Available Nodes	few to many
Distribution of Nodes	local to global
Hardware Resources	PCs, routers, wireless nodes, end-devices, links
Virtualisation Technology	container-based, user-mode, full virtualisation, non-virtualized, time sharing
Resource Booking	shared best effort or guaranteed CPU, RAM, bandwidth
Access Policies	private (consortium) only, hardware contributing partners, entry based on fee, memorandum based
Available Measurement Resoures and Tools	to capture experiments results and environment conditions
Available Configuration Tools	to deploy software, setup network topologies, emulate traffic characteristics

2 Motivation

Although we may have a good understanding from formal models and simulations, we are still unsure what interdependencies occur within a real network. This requires an experimental setup and the ability to make precise observations of the experiment. Precise monitoring and measurements are not only required to capture the experiment outcome but also to capture the experiment's environmental conditions so that parameters influencing the experiment can either be reduced or introduced into the model.

As one important measurement service we introduced Multi-Hop Packet Tracking [11] [17] which allows the researcher to passively observe the path of packets throughout the network domain. Our simplified packet tracking architecture consists of 1) multiple observation points (passive probes) deployed in the network, 2) a packet matcher that correlates the probe measurements, 3) a visualization tool to facilitate analysis of processed data. The probes export at least a packet ID and either the Time To Live (TTL) or an arrival timestamp for each observed packet to the collector. Based on the packet ID the packet matcher can correlate the observations and determine the packet's direction by the TTL or timestamp. Further packet tracking measurements can capture the experienced transmission quality (in terms of loss, one-way delay, and jitter) of packets between single routing hops which enables a more precise view on the network. Currently, we use the packet tracking architecture in different usage areas:

1. **Evaluation of Routing Protocols:** The evaluation of routing protocols is challenging because one requires a large amount of nodes and heterogeneous resources to test scalability and feasibility. Therefore, researchers often use simulations to evaluate the routing protocol performance because experiments and metrics can be easily implemented. Nevertheless, simulations for mobile adhoc and wireless networks are only of limited use in the real world, because influencing factors are yet not clear and omitted. With packet tracking measurements one can identify influencing factors, locate routing loops,

evaluate the fairness of the protocol and make assertions about the correctness and convergence of the routing path in a real world setting. In the EuroNF project Multinext we use packet tracking one way delay measurements to validate a model for the buffer occupancy in a multipath routing setup [1]. With this model, the buffer occupancy at the receiver caused by the out-of-order packet delivery over the different paths can be calculated. The model requires as an input the delay distribution of the packets on the different paths. We verified the model with active and passive (packet tracking) measured path delays. Furthermore, we use packet tracking for the evaluation of a new multipath routing approach [6] where groups of routers on the packet path exchange their buffer occupancy level information based on the synchronization of pulse coupled oscillators. With the use of packet tracking, we can analyze the impact of transmission delays for synchronization messages, convergence speed and stability in case of queue filling level variations.

2. **Functional Composition:** There is quite a large number of projects that deal with the functional decomposition of the current network stack into network functional blocks which will then be composed on application specific demands (a review can be found in [5]). Functional composition is similar to the Service Oriented Architecture approach used for business and Web services, but the services origin in the network domain (e.g. forward error correction, fragmentation, reliable connection, routing). Through functional composition, the integration of new functionalities can be improved which leads to increasing functionality within the network. In [17] we showed a functional composition approach which is based on a cooperative peer-to-peer system where each peer can offer functionalities (like content caching, transcoding, encoding) to other peers. We then used packet tracking to verify the functionality chain within the peer-to-peer network.
3. **Routing Security and Traceback Systems:** In [4], Goldberg et al. show how a similar packet tracking approach can be used to detect man-in-the-middle attacks based on authenticated packet reports. In case an abnormal amount of packet paths permanently terminate at or after a certain router, one can infer that a router may be misconfigured or compromised by an attacker. Furthermore, packet tracking enables on demand traceback systems [12]. In case network attacks are detected, one can trace the origin and path of single packets.

3 Requirements

Based upon the three usage areas, we have identified the following requirements for our experiments:

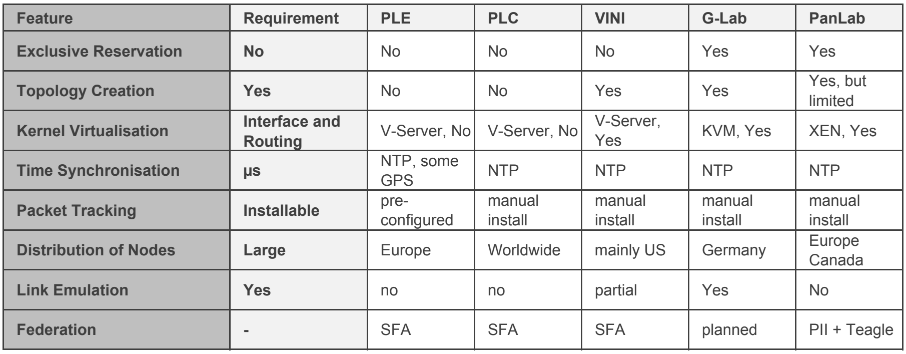
1. **Topology Creation:** All use cases require the configuration of network topologies, i.e. that some of the nodes in the experimental setup serve as routers and others as end hosts.

2. **Kernel virtualization:** For the evaluation of multipath protocols [6] [1] we need to be able to use our own routing algorithm, setup and configure virtual devices, and change flow tables. Some virtualization methods do not allow these changes as they may influence other researchers on the node.
3. **Time Synchronization:** For precise one-way delay measurement the clocks at the different observation points need to be synchronized. Within the Multinext project we require time synchronization accuracy around 1 μ at the end-nodes.
4. **Packet Tracking:** All the nodes should be either pre-configured with the packet tracking software (like the ANME Boxes in Planetlab) or one should be able to deploy the measurement probes on the nodes.
5. **Traffic Characteristics:** The Multipath model [1] and the Functional Composition Approach [17] are strongly influenced by path delay, delay variations, and packet loss. Therefore, reasonable traffic characteristics similar to the current Internet are required for our experiments, either 1) by preferably using large distributed nodes or 2) by using link emulation.

3.1 Available Experimental Facilities

Currently, we have access to the following experimental facilities 1) Planetlab Europe (PLE), 2) Planetlab Central (PLC), 3) VINI through SFA federation with Planetlab, 4) G-Lab, and 5) Panlab. PlanetLab is a global research network that supports the development of new network services. Planetlab Central is PlanetLab's worldwide headquarters based at Princeton University, and most nodes in the U.S. run PLC boot images. PlanetLab Europe is an own administrative domain of Planetlab nodes within Europe with independent slice management and own boot images. VINI is a virtual network infrastructure based on PLC, with nodes mainly situated in the U.S., except one node in Praha. G-Lab is an experimental facility funded by the German Federal Ministry for Education and Research. G-Lab allows the usage of different boot images and virtualization technologies (Planetlab, KVM, OpenVZ) but also the exclusive booking of

Table 2. Comparison of Experiments Requirements and Facility Features



Feature	Requirement	PLE	PLC	VINI	G-Lab	PanLab
Exclusive Reservation	**No**	No	No	No	Yes	Yes
Topology Creation	**Yes**	No	No	Yes	Yes	Yes, but limited
Kernel Virtualisation	**Interface and Routing**	V-Server, No	V-Server, No	V-Server, Yes	KVM, Yes	XEN, Yes
Time Synchronisation	**µs**	NTP, some GPS	NTP	NTP	NTP	NTP
Packet Tracking	**Installable**	pre-configured	manual install	manual install	manual install	manual install
Distribution of Nodes	**Large**	Europe	Worldwide	mainly US	Germany	Europe Canada
Link Emulation	**Yes**	no	no	partial	Yes	No
Federation	**-**	SFA	SFA	SFA	planned	PII + Teagle

resources. Although the Panlab federation is not yet fully operational, we have access to selected resources and the Teagle framework [15].

We tried to match our requirements to the characteristics of the facilities - as depicted in Table 2.

Planetlab Europe and Planetlab Central do not provide means for network topologies configuration. The setup and configuration of virtual interfaces that can be used to run own routing protocols is very limited due to the current restrictions in the Planetlab virtualization. VINI offers the configuration of the network topology using the rspec configuration of SFA (see next section), where one can choose between different links of the real physical network. VINI also allows the configuration of guaranteed bandwidth, but no link characteristics (loss, delay) can be specified. The Tomato web portal of G-Lab [8] offers an easy way to book virtual resources and the configuration of network topologies including the configuration of link characteristics (like guaranteed bandwidth, delay and loss). PlanetLab Europe is the only facility possessing Advanced Network Monitoring Equipment (ANME) that offers precise GPS time synchronization and pre-configured packet tracking. Other facilities rely on the Network Timing Protocol. As a results of our requirement analysis we realized that we cannot use a single experimental facility for our experiments.

3.2 Current State of Federation Frameworks

Several frameworks have emerged around large scale facility federation. Some of them have already been analyzed and compared in [7]. Lately, the Slice Based Federation Architecture (SFA 2.0) [9] as well as the PII federation framework [16] [14] gained considerable momentum and are deployed by a number of projects. In the following, we will introduce those two approaches and compare them in terms of architecture design decisions.

SFA2.0. The basis of this draft specification is the SFA 1.0 draft version [10] which was named Slice-Based Facility Architecture. SFA 2.0 aims to be roughly backward-compatible to SFA 1.0 and is the lowest common denominator in terms of interface and data type definitions. At the time of writing, the SFA 2.0 draft represents a rough consensus among the principals of the GENI control frameworks. However, it leaves many crucial aspects like resource description unspecified.

SFA 2.0 defines several entities, interfaces, and data types that collectively provide a control framework architecture. Among the main entities are owners and operators of a network substrate, researchers and developers, as well as identity anchors that drive authorization by asserting attributes or roles of other entities. Thus, SFA 2.0 defines three principals: a management authority (MA), a slice authority (SA), and a user. The principals interact with two key abstractions: components and slices. Components encapsulate a set of resources (e.g. CPU, memory, bandwidth, ports) and constitute the basic building blocks of the framework. Components are grouped into aggregates which are controlled by an aggregate manager (AM) and are under the authority of an MA governing the aggregate. Via the AM interface, allocation of resources to different users and their experiments can be requested. The MA is in charge of policy definition on how resources are allowed to be assigned to users. Resources are shared among multiple users (e.g. through virtualization). Such a share is called a sliver in SFA terms, while a collection of slivers is named a slice. Slices are requested via the AM interface.

PII Framework. The PII framework aims at provisioning and managing distributed testbeds for carrying out different kinds of testing and experimentation activities. A resource federation model [15] and an according prototype implementation [15] [13] have been developed that allow sharing resources beyond domain boundaries. Testing activities are supported by a Panlab Office, a coordination centre that supports interactions between experimenters and participating testbeds. Several architectural entities have been defined to allow for resource abstraction and management, collectively providing a control framework for distributed resource management. An important architectural component is the domain manager that controls resources of a specific (Panlab partner) domain. Current implementations make use of resource adapters to overcome resource heterogeneity. Resource adapters plug into the domain manager and abstract resource specific communication like device drivers do this on an operating system. Domain managers expose a specified interface that is used by an upper layer framework called Teagle. Teagle acts as a resource broker requesting resources via the domain manger interface from individual domains, relying on a common information model and a central registry. In addition, Teagle provides graphical user interfaces to work with resources (e.g. configure and reserve virtual setups) as well as orchestration capabilities to instantiate abstract virtual environment definitions on physical resources provided by the participating domains. The orchestration also resolves provisioning dependencies and enables parallel deployment and rollback functionality.

With respect to topology creation, the PII framework foresees an interconnection gateway (IGW) that allows to establish a virtual overlay network over public Internet to connect resources from distributed domains. IGWs are ingress-egress points to each site for intra-virtual-testbed-communication via one automatically configured multi-endpoint tunnel per virtual testbed. It is able to act as dynamically configurable hub and allows isolation of local testbed devices. One virtual private network (VPN) per virtual testbed instance is configured between all neighbor IGWs which enforces isolation of local resources by dynamically configuring collision domains. A collision domain is an isolated network segment on a partners physical test site where data packets are sent on a shared channel.

4 Implementation and Experiences

As shown in the analysis in section 3, the use of a single experimental facility cannot meet our experiment requirements and none of the federation frameworks offers topology creation support. Therefore, we used nodes from different facilities in our experimental setup. As the virtualization methods of PLE/PLC do only offer limited configuration of virtual interfaces we used an application level overlay software which places all nodes within a virtual Ethernet network and requires only few administrative privileges. An application level routing overlay creates virtual Ethernet links between different nodes, making the underlying links transparent to the user. In order to conduct precise one-way delay measurements we used Planetlab nodes that are monitored by the ANME box which provides GPS synchronization and pre-configured packet tracking. Furthermore, we used nodes from VINI and G-LAB for software based routers because their interfaces and the routing protocol are freely configurable.

We implemented a routing overlay software based on the freely available virtual distributed Ethernet (VDE) [2]. VDE provides the interconnection between virtual machines with consistent behavior to a real Ethernet network. The VDE network consists of virtual devices similar to the current Ethernet (interface, switch, cable, plug). VDE also allows the configuration of virtual link characteristics like loss, delay, delay variation, packet queue limitation, interface speed restrictions, packet reordering. VDE does not require any administrative privileges to run, only a virtual tap device at the host needs to be configured so that VDE can interact with the system and can set the virtual network addresses. We implemented a solution that deploys VDE on the different machines of the experimental facilities and configures the software according to the experiments network topology. Before setting up the virtual network the researcher needs to provide the credentials for the facilities (location of ssh-keys) and a topology outline similar to .dot representation (see Figure 2). The software will create a virtual switch at each node and set up a virtual tap interface which is connected to the switch. In case nodes belong to different virtual networks, multiple interfaces will be set up and accordingly configured. VDE runs in daemon mode upon the virtual interface and tunnels packets from one node to the other as if there was only one hop between them, adding and removing the IP headers of the real interfaces. The topology that is created in VINI can be integrated into the overlay by viewing the VINI nodes as a separate Ethernet subnet and adding additional interfaces that belong to subnets of the overall topology. It is also possible to create own virtual links in the VINI topology which are not physically connected or configurable by creating additional virtual interfaces that span a virtual overlay over the configured links. The same is also practicable for G-Lab, although we did not use the Tomato tool to configure the Overlay as it was not available by that time. In PLE/PLC the creation of virtual interfaces

```
digraph topology
{
      node1 [label="hostname, 192.168.1.10", ssh="ssh-param1"]
      node2 [label="hostname2, 192.168.1.11", ssh="ssh-param2"]
      node3 [label="hostname3, 192.168.1.12" ssh="ssh-param3"]
      node1 -> node2 [label="loss:0.8, delay:10"]
      node2 -> node3 [label="loss:0.7, delay:10"]
}
```

(a) Simple Topology Configuration File (b) Resulting Routing Overlay Setup

Fig. 2. Topology Creation

is restricted. It has to be ensured that the administrative tags vsys fd_tuntap, vif_up and vsys_vnet are enabled. Due to the interface configuration constraints, the PLE/PLC nodes can only be included into the overlay as end nodes.

5 Conclusion and Future Work

In this paper we presented three different usage areas for packet tracking in experimental facilities: 1) routing, 2) Functional Composition, and 3) network security. We analyzed different requirements for the facilities to support our experiments and identified the need of better topology creation support in experimental facilities. Current federation frameworks like SFA and the Panlab Framework lack the ability to create routing overlays over nodes from different facilities which would have been beneficial for our experiments. Therefore, we implemented a software based on Virtual Distributed Ethernet which integrates heterogeneous nodes from different facilities into a routing overlay. The software is easy configurable and tested in Planetlab Central, Planetlab Europe, VINI and G-Lab. It will be made publicly available under `http://www.free-t-rex.net/`. Based on the feedback that we will get we will also push similar approaches into the Teagle framework.

Acknowledgements. This work is partially funded by the German BMBF as part of the G-Lab Deep project and partially funded through the EU FP7 under grant agreement no 216366 for the NoE project "Euro-NF". We would further like to thank Andy Baview for is support with VINI, and the adminstrative staff of PLE and PLC for their support.

References

1. Zseby, T., et al.: Multipath routing experiments in federated testbeds. In: 3rd EU-Japan Future Internet Symposium (2010)
2. Virtual Distributed Ethernet (2010), `http://wiki.virtualsquare.org/wiki/index.php/Main_Page`
3. Gavras, A. (ed.): Experimentally driven research white paper, version 1. Technical report, ICT Fireworks (April 2010)
4. Goldberg, S., Xiao, D., Tromer, E., Barak, B., Rexford, J.: Path-quality monitoring in the presence of adversaries. SIGMETRICS Perform. Eval. Rev. 36, 193–204 (2008)
5. Henke, C., Siddiqui, A., Khondoker, R.: Network functional composition: State of the art. In: ATNAC 2010: Proceedings of Australasian Telecommunication Netoworks and Application Conference. IEEE (2010)
6. Kleis, M.: Congestion aware path selection for multipath-routing. In: KUVS Workshop Future Internet (2010)
7. Magedanz, T., Wahle, S.: Control Framework Design for Future Internet Testbeds. e & i Elektrotechnik und Informationstechnik 126(07/08), 274–279 (2009) ISSN: 0932-383X (print) ISSN: 1613-7620
8. Tomato Topology management tool (2010), `http://tomato.german-lab.de/`

9. Peterson, L., Ricci, R., Falk, A., Chase, J.: Slice-based federation architecture,version 2.0. Technical report (July 2010)
10. Peterson, L., Sevinc, S., Lepreau, J., Ricci, R., Wroclawski, J., Faber, T., Schwab, S., Baker, S.: Slice-based facility architecture, draft version 1.01. Technical report (August 2008)
11. Santos, T., Henke, C., Schmoll, C., Zseby, T.: Multi-hop packet tracking for experimental facilities. demo. In: SIGCOMM 2010: Proceedings of the ACM SIGCOMM 2010 Conference, pp. 447–448. ACM, New York (2010)
12. Timothy Strayer, W., Jones, C.E., Tchakountio, F., Snoeren, A.C., Schwartz, B., Clements, R.C., Condell, M., Partridge, C.: Traceback of single ip packets using spie. In: DISCEX (2), pp. 266–270 (2003)
13. Wahle, S., Harjoc, B., Campowsky, K., Magedanz, T., Gavras, A.: Pan-European testbed and experimental facility federation - architecture refinement and implementation. International Journal of Communication Networks and Distributed Systems (IJCNDS), Special Issue on Recent Advances in Testbed Driven Networking Research 5(1/2), 67–87 (2010) ISSN: 1754-3924 - ISSN (Print): 1754-3916
14. Wahle, S., Magedanz, T., Campowsky, K.: Interoperability in Heterogeneous Resource Federations. In: 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2010), ICST (May 2010) ISBN: 978-963-9799-92-9
15. Wahle, S., Magedanz, T., Gavras, A.: Conceptual Design and Use Cases for a FIRE Resource Federation Framework. In: Towards the Future Internet - Emerging Trends from European Research, pp. 51–62. IOS Press (2010) ISBN: 978-1-60750-538-9 (print), 978-1-60750-539-6
16. Wahle, S., et al.: Emerging testing trends and the panlab enabling infrastructure. IEEE Communications Magazine 49(3), 167–175 (2011)
17. Zseby, T., Henke, C., Kleis, M.: Packet Tracking in Planet Lab Europe - A use Case. In: Magedanz, T., Gavras, A., Thanh, N.H., Chase, J.S. (eds.) TridentCom 2010. LNICST, vol. 46, pp. 265–274. Springer, Heidelberg (2011)