Dynamic Virtual Overlay Networks for Large Scale Resource Federation Frameworks

Sebastian Wahle, André Steinbach, Thomas Magedanz, and Konrad Campowsky

Fraunhofer FOKUS, Germany
{sebastian.wahle,konrad.campowsky}@fokus.fraunhofer.de
Deutsche Telekom Laboratories, Berlin, Germany
andre.steinbach@telekom.de
Technische Universität Berlin
tm@cs.tu-berlin.de

Abstract. Resource federations provide access to distributed resources committed by participating organizations. This concept is currently applied to provide large scale experimental facilities serving Future Internet research and development. We have developed a model and framework for generic resource federation and have implemented an according prototype system that allows federating heterogeneous resources on a pan-European scale. With this paper we show how heterogeneous federated resources can be interconnected by means of meshed domain border gateways that provide separated Layer 2 tunnels for resource groupings following our federation model. This enables to dynamically build virtual overlay networks over the public Internet to support various experimentation purposes.

Keywords: Resource Federation, Future Internet, Panlab, Teagle, Overlay Networking.

1 Introduction

Resource Federation allows sharing and re-using Information and Communication Technology (ICT) resources across independent administrative domains that are usually governed by different organizations. This approach is applied in several fields such as Grid and Cloud Computing, as well as federated identity management (e.g. eduroam[1]) for several reasons. Most prominently, as today's societies are concerned about ICT energy consumption, re-using infrastructure and services across the silos of individual organizations is seen as a promising way to reduce the overall energy consumption and over provisioning in the ICT field.

Furthermore, the pace of network convergence and technology evolution has dramatically decreased infrastructure lifetime – the time an infrastructure remains at the technology's cutting edge. This makes investments in specialized expensive infrastructures more risky than they were already [1] and particularly applies to

[1] http://www.eduroam.org

T. Korakis et al. (Eds.): TridentCom 2011, LNICST 90, pp. 180–192, 2012.

complex cross-layer and cross-technology infrastructures such as Future Internet (FI) research testbeds.

Here, federation is expected to deliver a number of benefits [2]:

- Federation enables access to additional resources increasing the scale of potential experiments.
- Federation can considerably cut down the associated cost of large scale FI experiments. [3]
- Federation enables access to resources with unique properties to enrich experiments.
- Combining resources from different communities promotes the collaboration between these and the related research groups (e.g. Telco and Internet).
- A collection of testbeds that share or feature similar properties or technologies might eventually evolve into the backbone of the Future Internet itself.

Today, numerous research programs build upon a federation approach. Examples are the NSF programs GENI [4] and FIND [5] as well as the European FIRE initiative [6] [7]. In Asia similar programs have been launched such as AKARI [8] in Japan. An in-depth discussion and comparison between the different federation and resource control framework approaches for experimental facilities has been published earlier [9].

Many aspects of distributed computing and the management of distributed resources have been investigated in the past. Computing power federation has been looked at in the Grid domain since years. Lately, Cloud Computing federation has been recognized as an interesting and important field due to considerable industrial impact. For example, live virtual machine migration across clouds and therefore across the boundaries of administrative domains, holds unexplored industrial potential once the numerous challenges (data privacy, multitenancy, etc.) have been addressed sufficiently.

Despite previous efforts, generically federating heterogeneous resources across multiple administrative domains and on multiple federation levels, involves so many technical, operational, and legal issues that it can be considered a valid research field with many yet unsolved issues. In order to realize the vision of fully federated ICT resources that can be used transparently and seamlessly, the following fields have to be addressed: resources description, resource interconnectivity, resource access control, service level agreements, resource usage policies, resource management, resource life cycle, operational procedures, legal frameworks, provider/user incentives, business frameworks, market platforms, etc. [2]

In this paper we describe how to provide cross-domain resource interconnectivity building upon virtual overlay network technology. This allows provisioning separated orchestrated infrastructure services and maximizing resource utilization due to shared access. The current prototypes are integrated in to our federated platform operating on a pan-European scale.

Our main point in addressing challenges as seen in the discussion[2] on conforming GENI aggregates through existing control frameworks is to pre-orchestrate

[2] http://groups.geni.net/geni/attachment/wiki/Gec7ControlFrameworkAgenda/falk-cfwg-gec7-stitching-summary.pptx?

interconnection parameters as well as name- and addressing-spaces of used protocols in local testbeds. Federation with other environments is therefore possible, since resources are limited to certain rules of setup and communication by Teagle (see Fig. 1) and gateway controlled interconnection.

The next section introduces our federation model and approach to resource management. Section 3 outlines the virtual overlay networking concept and the design of our implementation. Section 4 explains further details and lessons learned in terms of a use case while section 5 concludes the paper.

2 Federation and Domain Level Resource Management

We have developed a Resource Federation Model [1] and an according prototype implementation [1] [11] that allows sharing resources beyond domain boundaries. As this has been discussed in previous publications as cited above, we will not go into details regarding our framework and prototypes but rather provide a broad overview for the convenience of the reader.

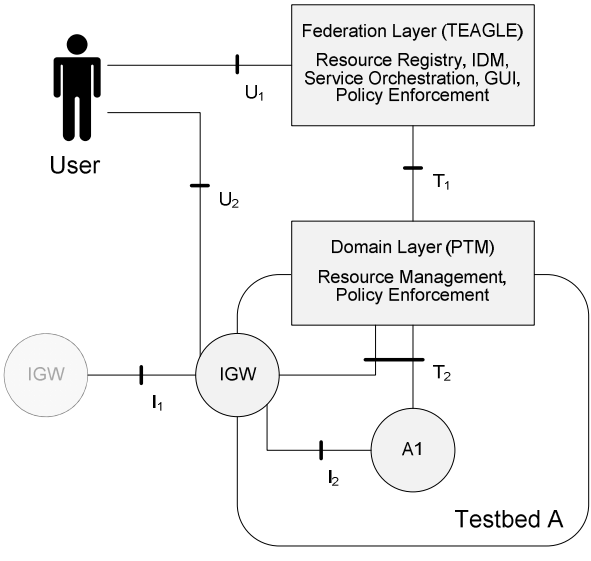

Fig. 1. Federation & domain level resource management & domain interconnection

In the following, we will outline how resources are controlled inside domains and how this relates to the federation level. Relying on the federation framework, users can get access to distributed resources and group them into a virtual environment which is called a VCT (virtual customer testbed). Teagle [12] as a collection of central federation services helps the user in configuring and reserving a desired set of resources. Resources are offered by participating organizations across Europe [10] [3].

Fig. 1 shows how resources (A1) are controlled and which interfaces are involved. Between the federation layer and the domain layer resides reference point T1 which is defined in subsection 2.1.

The T1 reference point represents a domain manager which is responsible for handling provisioning requests on behalf of its domain.

The T2 reference point, representing the control interface of actual resources, is not specified. Our domain manager implementation (PTM) handles this by providing a framework for resource adaptors (RA) to plug into. RAs implement resource specific interfaces on reference point T2, like a device driver controls a device plugging into an operating system.

On the federation layer, our Teagle framework implementation offers several services to the user and other framework entities. Among those central services are a registry, a common information model, identity management, orchestration of distributed services, policy handling, and graphical user interfaces for resource registration, configuration, deployment, etc. However, the main focus of this article is on the domain level resource handling and how resources provided by different domains can be interconnected on reference points I1 and I2 which are explained in detail in section 3. To ease the understanding of those parts, we give some more insights into the T1 interface exposed by domain managers.

2.1 T1 Reference Point

Resources in Teagle exist as types and instances. An instance can be instantiated from a type. Example: a resource type might be a virtual machine type, with configuration parameters (e.g. CPU, memory, storage) to be set, while a deployed virtual machine instance with defined parameters is an instance of type virtual machine. Create commands are typically executed upon types resulting in the creation of new instances, whereas delete, update, and read commands are typically executed upon instances.

Resource instances that reside inside a domain are typically organized in hierarchies where a resource "lives" inside another resource. This might for example apply for a piece of software that is installed on a computer. Every resource instance can be contained in at most one parent instance. The containment relation for a resource can in most cases be omitted, leaving it to domain managers to choose an appropriate parent for a newly created instance.

Every resource instance is uniquely identified by an identifier. Domain managers are responsible for assigning an identifier to each resource instance under their control. They must also ensure the uniqueness of these identifiers within their own domain.

Teagle does not impose strict limitations regarding the form or semantics of an identifier. An identifier is not restricted in length and may consist of any printable ASCII[3] character. The only requirement for identifiers is that they consist of a prefix and a local name separated by a dash character.

[3] American Standard Code for Information Interchange.

Example

```
/node-0/mysql-1
/node-1/mysql-1
```

The two examples identifiers represent two different instances of a MySQL software package, although they have the same local name "1". However, they have different pre-fixes `/node-0/mysql` and `/node-1/mysql`. The pre-fixes also illustrate our concept of resource hierarchy where both MySQL instances are hosted by a different machine (`node-0/1`).

On the federation layer an additional pre-fix is needed in order to map resource instances to domains. Here, a prefix per domain is used, e.g. "`fokus.`" for the Fraunhofer FOKUS domain and its managed resources.

```
fokus./node-0/mysql-1
```

In the following, we discuss the operations exposed by domain managers on the T1 interface which basically provides the CRUD operations (create, read, update, delete). The specification of the XML format used for configuration data can also be found at http://www.fire-teagle.org/T1.

2.1.1 Create

```
add_resource(parent_id: Identifier, typename: TypeName,
[name: LocalName,] config: Configuration, vct: VCTName) :
Identifier
```

The `add_resource` operation requests the instantiation of a given resource type with a given configuration as a child of the existing resource instance denoted by `parent_id`, optionally specifying a local name. The `vct` parameter indicates which VCT this instance will be part of. Upon success, an identifier of an existing resource instance is returned. This can be either an instance that was created in response to the request or an instance that had existed before and might hence be used by others. Likewise, the domain manager can choose to return an identifier of an instance that is not a child of the instance given in `parent_id`.

2.1.2 Read

```
list_resource(parent_id: Identifier, typename: TypeName):
{ Identifier }
```

The `list_resources` operation retrieves a list of all resource instances that are regarded as children of the instance denoted by `parent_id` and that are of the type given in `typename`. If `parent_id` is omitted, all instances at the root of the resource hierarchy must be listed. If `typename` is omitted, instances if all types must be listed.

```
get_resource(identifier: Identifier): Configuration
```

The `get_resource` operation retrieves configuration information for the existing resource instance denoted by `identifier`.

2.1.3 Update

```
update_resource (id: Identifier, config: Configuration):
  Configuration
```

The `update_resource` operation requests the reconfiguration of an existing resource instance denoted by `id` with the configuration specified in `config`. This configuration does not have to include all parameters of the resource instance. It is sufficient to include only the parameters that are to be changed. Upon success, the full configuration of the resource instance is returned.

2.1.4 Delete

```
delete_resource(identifier: Identifier): None
```

The `delete_resource` operation requests the deletion of the existing resource instance given by `identifier`. It is up to the domain manager to decide if the instance is actually deleted, so this can rather be viewed as an indication that a certain instance is not needed by the federation layer anymore.

In this section we showed how the generic T1 commands have been defined. The mapping of these commands to actual resource control is up to the domain manager implementation.

Configuration data of resources is defined by a common information model. All resources controlled by our system need to be described in terms of this model. Although this introduces initial overhead at resource registration/publication time, it allows for a fine grained resource management. For several resource types (e.g. virtual machines, etc.) existing type models can be re-used. The same is true for resource adaptor implementations.

The next section deals with the reference points I1 und I2 explaining how we interconnect resources across multiple sites which is the main focus of this paper.

3 Dynamic Virtual Overlay Networks

There are a number of problems to be solved for interconnecting resources provided by testbeds of heterogeneous buildup in order to establish inter-testbed connectivity. To securely connect test sites using public Internet and in order to allow for transport of experiment traffic, dynamic virtual overlay networks are established. This hides the complexities of the physical testbed infrastructure and allows the dynamic provisioning of virtual customer testbeds (VCTs) involving resources provided by distributed sites.

We designed and implemented an interconnection gateway (IGW) resource (and the associated resource adaptor for the management framework outlined in section 2) that functions as a border gateway and connects physical testbeds with each other in a peer-to-peer fashion.

IGWs are ingress-egress points to each site for intra-VCT-communication via one automatically configured multi-endpoint tunnel per virtual testbed. It is able to act as a dynamically configurable hub and allows isolation of local testbed devices. One virtual private network (VPN) [14] per VCT instance is configured between all neighbor IGWs and enforces isolation of local resources by dynamically configured collision domain isolation. A collision domain is an isolated network segment on a partner´s physical test site where data packets are sent on a shared channel.

Fig. 2 shows how IGWs interconnect two physical testbeds using one VPN tunnel between partner testbed B and partner testbed E. However, two different VCTs are established containing partner resources B1/E1 and B2/E2.

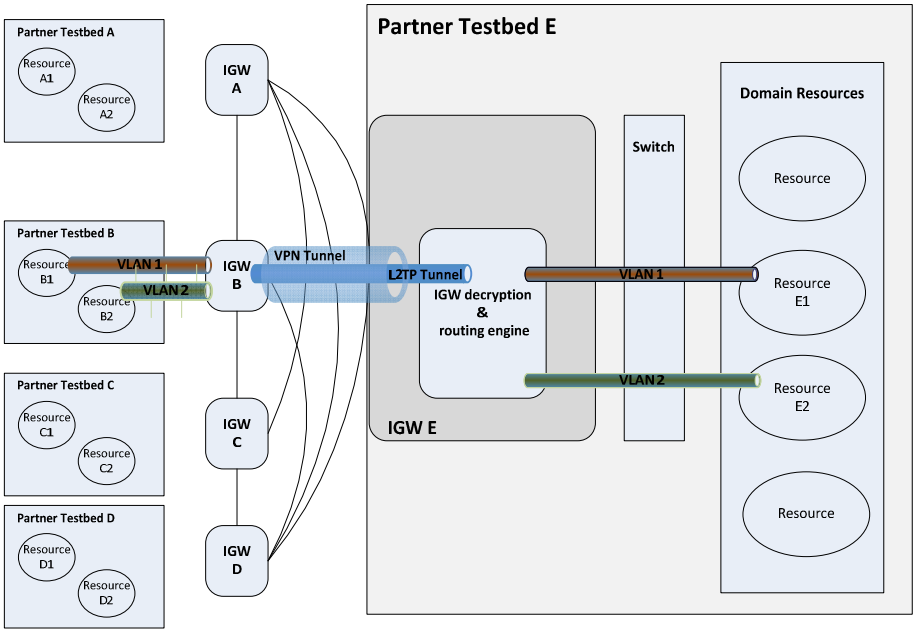

Fig. 2. Interconnected physical testbeds with VPN/VLAN based VCT overlay

IGWs are foreseen to "mesh" automatically with each other and therefore establish connections to other peer IGWs. An important design criterion was to make them as self-configuring as possible.

For such meshing of all IGWs that are part of a specific VCT, a stateless low-overhead tunneling was chosen. The IGW resource might be completely transparent to the customer using VCT planning tools provided by Teagle. However, this depends on the experiment. The IGWs can be exposed as any other resources in Teagle or not, depending on the level of configuration granularity that is demanded by the experimenter.

On the IGWs internal connection state machine, active VCTs are lists of tuples consisting of the other IGW´s external address and the collision domain(s) associated with the specific VCT behind it.

Each interconnection state can be expanded by adding more interconnections. New interconnection states do not interfere with existing states. They use the same VPN tunnel but are separated during the routing and filtering process. This guarantees an on-demand automatic IGW-to-IGW meshing of all test sites with stateless low-overhead tunneling without using proprietary inter-IGW protocols.

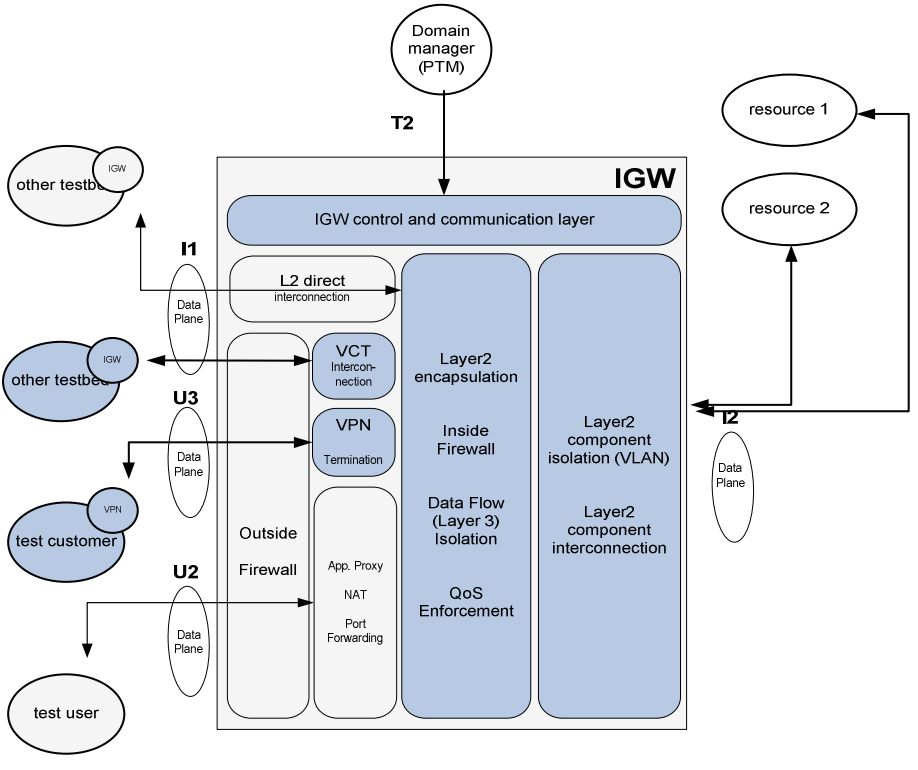

Fig. 3. Interfaces and internal building blocks of an IGW testbed border gateway

Fig. 3 shows the external interfaces and internal building blocks of an IGW. The darker blocks mark the functionalities used for the testbed example shown in Fig. 2.

An experimenter is able to connect single devices (e.g. test clients) to his VCT, using a “customer dial in” feature. This Layer Two Tunneling Protocol (L2TP) based on-demand tunnel [13] delivers direct access to a specific VCT as if the experimenter would work from within a partner domain that had a local IGW.

The main functionalities provided by an IGW are to interconnect, keep, and protect the mapping of local collision domain communication to external VPN interconnection. Therefore, it functions like an IP-based trunking device for testbed components communicating on data planes separated by collision domains on the internal side and VPN based access on the external side. Routing of data plane packets in-between these secure channels is done by the interconnection engine.

Furthermore, if demanded by a request via the domain manager, QoS rules may be enforced on routing decisions, for instance limiting connection of one VCT to another testbed to a certain maximum throughput rate. In front and in the back of the interconnection engine, the secure channels are being de-encapsulated/decrypted and filtered by a stateful IP-based firewall. This makes sure that access to specific resources can be restricted as defined by the resource provider (the authority governing the domain).

On the external side of the IGW there may be also generic collision domains bridged to other testbeds that are not publicly accessible. In this way it is possible to perform real QoS reservations such as ATM or fibre optic links.

The north side of any IGW is the control and communication layer facing the domain manager/resource adaptor that uses simple command/reply communication (e.g. for activation of a QoS rule) but also subscription based event updates (e.g. some security rule was violated) to communicate on reference point T2 (see Fig. 1).

As explained earlier, besides the IGW as default gateway, test sites usually provide additional resources like physical servers, virtualized resources, or dedicated testing equipment like radio base stations, protocol testers, network equipment (routers, switches, etc.). Such resources are exposed to the experimenter via the domain manger and the Teagle framework building several abstraction layers to provide a large pool of federated resources.

For connecting and providing such resources in separated VCTs, collision domain channel isolation is required. Therefore, IEEE 802.1Q VLANs [15] based systems have been added as a mandatory requirement and a prerequisite for conducting different test sessions in parallel that are fully separated. If this is not supported by the chain of resources used to interconnect resources from different sites, no full isolation can be guaranteed. This might impact some types of experiments while for others it might not be an important aspect depending on the focus of the experiment.

Fig. 4 shows our isolation concept. Since several VLANs may be used as a shared medium to connect multiple resources in a single test site, the experimenter has full control over the network topology to be deployed. A virtualized host resource may act as a software router within a VCT. However, this flexibility comes with a significant complexity in configuring the network layer.

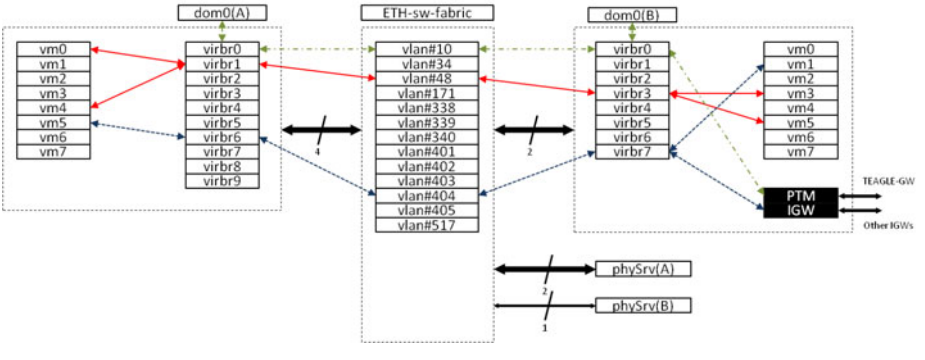

Fig. 4. VLAN based local testbed collision domain isolation between parallel VCTs

A VCT virtual link mapped on a local site VLAN is capable of connecting virtual host resources and physical systems. For physical components directly attached to the virtual link, the responsible port on the Ethernet switching domain is added to the VLAN in untagged mode. For virtualized server resources running on the hosting resource, VLAN tags can be assigned transparently.

4 Use Case

This section outlines how we used the platform described in the previous sections for the execution of a set of experiments. The experiments have been set up and executed to gain insight into:

- the behavior of different mesh routing protocols.
- multipath traffic distribution on real routing nodes.
- the impact of changing routing conditions on different kind of traffic and transmission protocols (download, streaming, etc.).
- real time simulation of fault resistant meshed networks with failing nodes.
- scalability, quality of service, and load balancing aspects.

Fig. 5 shows the VCT setup for our experiments. Via the Teagle framework, several resources have been reserved including physical servers that host virtual machine appliances simulating a mesh network. As simulation resources have been used, the level of experiment realism is not yet satisfying. However, the intention was to keep the costs as low as possible. Therefore, only resources from two individual test sites have been used to host on the one hand the server and client resources and on the other hand the mesh network simulation resource. The interconnection between the different resources was enabled by IGWs on both sides.

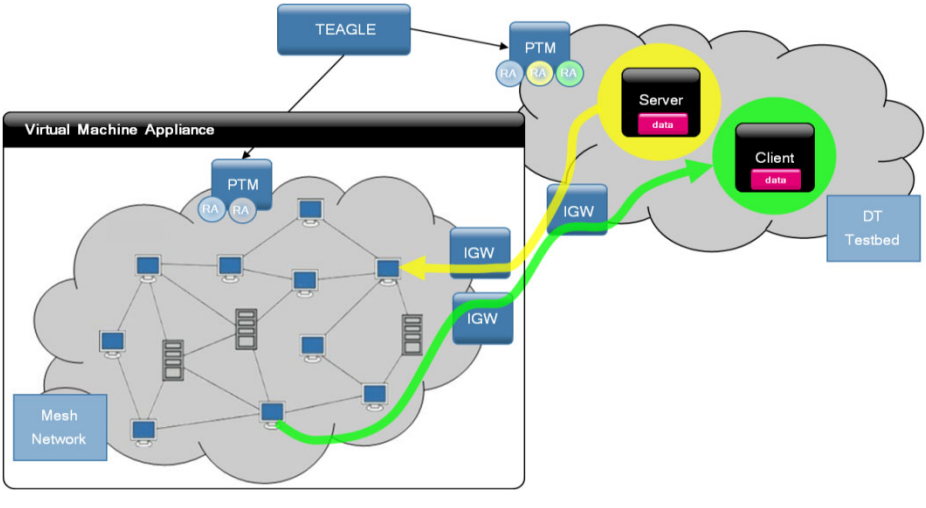

Fig. 5. Mesh networking experiment involving interconnection gateways for routing of experiment traffic across test sites

This setup allowed us to interconnect client and server resources across different test sites (interconnected via public Internet) using a Layer 2 network which was important for our analysis of mesh routing protocols and multipath traffic distribution.

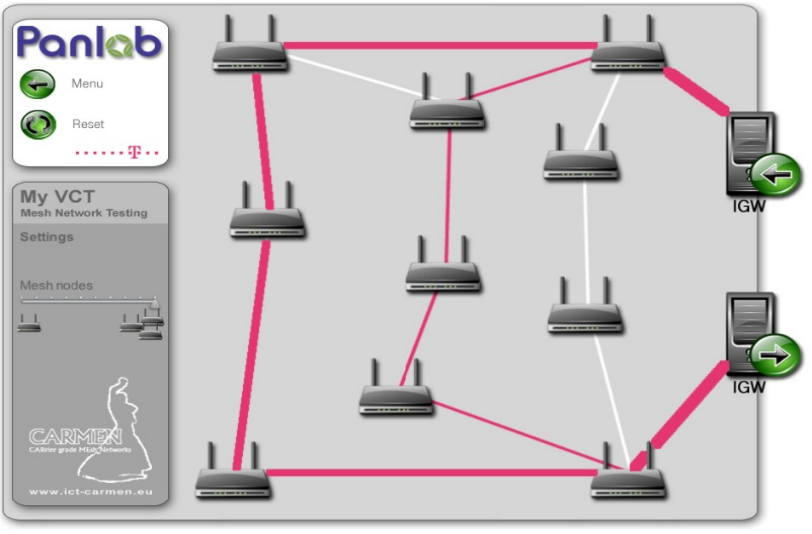

Fig. 6. Application for experiment tuning at experiment run-time

Fig. 6 shows a specific tool that has been developed for demonstration and experiment fine tuning purposes. It allows modifying traffic routes by creating and cutting links between nodes. This works intra-domain as well as cross-domain.

The virtual machine appliance may be modified and monitored during experiments using this front-end communicating to the PTM. The number and position of mesh nodes, connectivity paths, and its maximum data rate can be changed on demand. Exact data are delivered by a detailed log file for further processing.

The overlay is established dynamically as needed. However, so far, the overlay parameters cannot be modified at experiment run time, e.g. to restrict traffic throughput across domains dynamically. The major point of interest was on route selection capabilities and stability of different mesh protocols. The line width in Fig. 6 shows the relative intra-domain link utilization.

Further aspects that have been investigated using this experimental setup are: effectiveness of network route priority change strategies in case of failing nodes and the impact of multipath traffic distribution on standard stream-based protocols.

Resource virtualization, inter-domain delays, different bandwidth, and other phenomena of multi site experiments, heavily influence the experiment results. Therefore, we share our lessons learned from the experiments described above and give insight to some best practices that were developed over time to ensure usability and efficiency:

- Many setups created by the Teagle framework and the underlying resource management environment encapsulate stand-alone functions, such as a mesh network simulation, and can be tweaked in advance on a local basis. This allows holding them on standby to avoid a full re-orchestration at experimentation run time. However, this limits the generic utility of resources and might require additional domain specific knowledge at experiment design time.
- If needed, mentioned functions can be duplicated for scalability reasons in a pre-configured and optimized way on other sites. Teagle supports easy setup and interconnectivity of such heterogeneous, replicated capabilities relying on a common control framework.
- Resources that are predictable or less performance-critical can be allocated to remote locations, as seen with the data source and sink in the use case described above. However, the experimenter needs to keep in mind that the hop count of remote resources may be noticed and taken in to account.
- Using IGWs is essential to enforce communication rules orchestrated by Teagle but might result in a bottle neck in case of high-speed or high-bandwidth experiments. A possible workaround would be to interconnect specific IGWs with more dedicated connections or using more than one IGW with separate connections per test site.

5 Conclusion and Future Work

Resource Federation is an important aspect for Future Internet experimentation as the Internet itself is based upon federation mechanisms. The facilities that provide support for experiments today provide a great heterogeneity of resources in order to facilitate different experiments and serve multiple communities.

However, it can be observed that the setup of experiments that include many heterogeneous resources across distributed sites and across different federation frameworks is still very difficult and time consuming making large scale experiments cost intensive.

More work in the area of heterogeneous resource federation is expected and needed in order to enable the full chain of Future Internet experimentation starting from an abstract idea/model, moving to a simulation/emulation, and result in a large scale real system deployment.

Our virtual overlay mechanism that allows to flexibly interconnect resources across different sites seeks to enable the important step from a simulation/emulation to a real system taking into account that most test sites rely on public Internet to connect to other sites. Whenever more advanced equipment is in place such as optical links between sites, another class of experiments is possible. We will continue to work into this direction. The ultimate goal is to enable experimenters to choose from several test site interconnection technologies whenever this is supported by the site. This would allow for a level of experiment realism that is very hard (and costly) to achieve today.

From our perspective, resource federation mechanisms should allow for flexible heterogeneous resource provisioning across federations where a resource can be anything, including cross domain interconnection devices. This vision is driving our work.

Acknowledgments. Parts of this work received funding by the European Commission's Sixth Framework Programme under grant agreement no.: 224119. Also, we would like to thank the PII/Panlab consortia for the good collaboration as well as Prof. Dr. Paul Müller (TU Kaiserslautern/G-Lab), for his support on our federation ideas.

References

[1] 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)

[2] Wahle, S., Magedanz, T., Fox, S., Power, E.: Heterogeous resource description and management in generic resource federation frameworks. In: Proceedings of the 1st IFIP/IEEE Workshop on Managing Federations and Cooperative Management (May 2011)

[3] Wahle, S., et al.: Emerging testing trends and the panlab enabling infrastructure. IEEE Communications Magazine 49(3), 167–175 (2011)

[4] National Science Foundation, GENI website, `http://www.geni.net`

[5] National Science Foundation, FIND website, `http://www.nets-find.net`

[6] European Commission, FIRE website, `http://cordis.europa.eu/fp7/ict/fire`

[7] Gavras, A., Karila, A., Fdida, S., May, M., Potts, M.: Future internet research and experimentation: the FIRE initiative. SIGCOMM Comput. Commun. Rev. 37(3), 89–92 (2007)

[8] AKARI project website, `http://akari-project.nict.go.jp/eng/index2.htm`

[9] Magedanz, T., Wahle, S.: Control Framework Design for Future Internet Testbeds. e & i Elektrotechnik und Informationstechnik 126(07/08), 274–279 (2009)

[10] Website of Panlab and PII European projects, supported by the European Commission in its both framework programmes FP6 (2001-2006) and FP7 (2007-2013), `http://www.panlab.net`

[11] 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)

[12] Teagle website, `http://www.fire-teagle.org`

[13] RFC 2661: Layer Two Tunneling Protocol "L2TP". The Internet Society (August 1999)

[14] RFC 4026: Provider Provisioned Virtual Private Network (VPN) Terminology. The Internet Society (March 2005)

[15] IEEE Standard for Local and metropolitan area networks: Virtual Bridged Local Area Networks, 802.1Q. IEEE Computer Society, New York (2006)