Spontaneous Virtual Networks: On the Road Towards the Internet's Next Generation

Spontane Virtuelle Netze: Auf dem Weg zur nächsten Generation des Internets

Oliver P. Waldhorst, University of Karlsruhe (TH),
Christian Blankenhorn, University of Stuttgart,
Dirk Haage, Ralph Holz, University of Tübingen,
Gerald G. Koch, Boris Koldehofe, University of Stuttgart,
Fleming Lampi, University of Mannheim,
Christoph P. Mayer, Sebastian Mies, University of Karlsruhe (TH)

Summary Novel Internet applications demand global availability of complex services that can adapt dynamically to application requirements. At the same time, pervasive Internet usage and heterogeneous access technologies impose new challenges for service deployment. We present Spontaneous Virtual Networks (SpoVNet), a methodology that enables easy development of new services with transparent support for mobility, multi-homing, and heterogeneous environments. This article presents the overlay-based architecture of SpoVNet that supports the spontaneous deployment of new services as well as a seamless transition towards future networks. SpoVNet's architecture offers support for the underlay aware adaptation of overlays by the use of cross-layer information. In the context of two exemplary services like a group communication service and an event service as well as two demanding applications – a realtime online game and a video streaming application – we illustrate how SpoVNet is of value in establishing services and applications for the Next Generation Internet. ►►► **Zusammenfassung** Moderne Internetanwendungen setzen die Existenz weltweit verfügbarer, fortgeschrittener Dienste voraus, die sich zur Laufzeit an Anwendungsanforderungen anpassen können. Gleichzeitig werden die Anbieter solcher Dienste durch die allgegenwärtige Benutzung des Internet mit verschiedensten Zugangstechnologien vor neue Herausforderungen gestellt. Wir stellen mit Spontaneous Virtual Networks (SpoVNet) eine Methodik vor, welche die Entwicklung neuer Dienste erleichtert und die Heterogenität von Netzen und Geräten sowie Mobilität und Multi-Homing vor ihnen verbirgt. Dieser Artikel geht auf die Architektur von SpoVNet ein. Sie basiert auf Overlays und ermöglicht die spontane Bereitstellung neuer Dienste ebenso wie den Übergang in zukünftige Netze, in die solche Dienste bereits integriert sind. Durch das Angebot schichtenübergreifender Information unterstützt die SpoVNet-Architektur auch die Selbstorganisation von Overlays mit Rücksicht auf das Underlay. Anhand zweier Beispieldienste (Gruppenkommunikationsdienst und Ereignisdienst) und zweier anspruchsvoller Anwendungen (Realzeitspiel und Videostreaming) veranschaulichen wir den Nutzen von SpoVNet für die Bereitstellung von Diensten und Anwendungen im Internet der nächsten Generation.

KEYWORDS C.2.1 [Computer Systems Organization: Computer-Communication Networks: Network Architecture and Design]; C.2.4 [Computer Systems Organization: Computer-Communication Networks: Distributed Systems]; C.2.2 [Computer Systems Organization: Computer-Communication Networks: Network Protocols]; C.2.6 [Computer Systems Organization: Computer-Communication Networks: Internetworking]

1 Introduction

The Internet has undergone many changes with respect to end-user behavior, application requirements, and network technology. End-users expect, e. g., stable connectivity in spite of mobility, multi-homing, and use of heterogeneous network access. Furthermore, Internet applications aim to provide end-users with instant and permanent interaction.

The original Internet was not designed to support such requirements. As a consequence, network-oriented services for, e. g., Quality of Service (QoS), group communication, or user mobility have been integrated into the Internet architecture as patches and half-layers.

In most cases such services lack global deployment, like QoS in UMTS or IEEE 802.11e-based access networks, or multicast in the core networks of Internet service providers. Consequently, innovative applications have had to implement end-to-end support for their requirements themselves.

In this article we present *Spontaneous Virtual Networks* (SpoVNet), a methodology that enables flexible, adaptive, and spontaneous provisioning of application-oriented network services on top of heterogeneous networks. SpoVNet copes with changing and diverse network technologies and transparently supports end-user mobility. Moreover, it provides sophisticated communication interfaces along with end-to-end support for QoS and security. In this paper we highlight the main concepts of the SpoVNet architecture and illustrate how SpoVNet is of value in the development of new demanding services and applications.

2 Challenges and Key Concepts

A Future Internet architecture has to provide robust communication in an environment characterized by mobility, multi-homing and heterogeneity. Next Generation Internet applications also expect end-to-end provision of QoS and security. Significant amount of research has been done to meet these challenges in a completely re-designed Internet. However, the establishment of such a 'clean-slate' approach seems not to be feasible in the near future. *Overlay networks* (or for short *overlays*) provide an appropriate solution for the time in between. Overlays are composed of logical application layer connections and, thus, can be deployed without support of underlying networks. Additionally, overlays can provide self-organization and robust communication over a variety of heterogeneous technologies.

Within SpoVNet, overlays are used in three different ways: First, they provide basic connectivity between the individual devices taking part in a distributed application. Second, they provide an infrastructure for the spontaneous deployment of novel network- and application-oriented services between the devices. Third, they are used for the implementation of those services themselves.

In addition, SpoVNet enables an efficient adaptation of overlay nodes and connections to the underlying networks by provision of *cross-layer information*. This is the key to additional features of the SpoVNet approach like the establishment of probabilistic end-to-end QoS guarantees.

A number of related approaches also use overlays as a general concept. The *SATO* [15] subproject of Ambient Networks, for example, enables the flexible creation of service aware transport overlays. The *Autonomic Network Architecture* (ANA) [7] employs generic building blocks for the creation and optimization of overlays.

In comparison to SATO, SpoVNet provides structural building blocks (i. e., transport links) to compose overlay-based services instead of functional building blocks (e. g., caching, transcoding) for the composition of new overlay services. Thus, the concepts developed by SATO and SpoVNet can complement each other. Compared to ANA, which establishes an overlay-based architecture with explicit support of the 'core' of the network (e. g., of core routers), the SpoVNet architecture is implemented at the edges of the network (i. e., in the end systems) with optional core support.

SpoVNet's key concepts – overlay networks and cross-layer information – are employed in the layers and components of the SpoVNet architecture which is described in the following section.

3 SpoVNet Architecture

The design of the SpoVNet architecture has been driven by two major objectives: (1) to support flexible, underlay-aware, secure, and spontaneous provisioning of application-oriented and network-oriented services on top of heterogeneous networks, and (2) to enable a seamless transition from current to future generation networks by introducing suitable abstraction layers.

3.1 Overview

SpoVNet may run on arbitrary devices like personal computers, notebooks, PDAs, or smartphones. Such a device is called a *SpoVNet Device*. All devices participating in the same distributed application constitute a *SpoVNet Instance*. Within a SpoVNet Instance, each SpoVNet Device is logically represented by a *SpoVNet Node*. All nodes within an instance are connected by an overlay structure denoted as *Base Overlay*. This is built on top of the physical network and may span multiple domains, transport and network layer protocols, and access technologies as shown in Fig. 1.

The SpoVNet architecture containing the main components *SpoVNet Core* and *SpoVNet Node* is shown in Fig. 2. A SpoVNet Core runs once per device and offers basic communication functionality. Since a device may participate in multiple SpoVNet Instances at the same time, one SpoVNet Node is run per SpoVNet Instance.

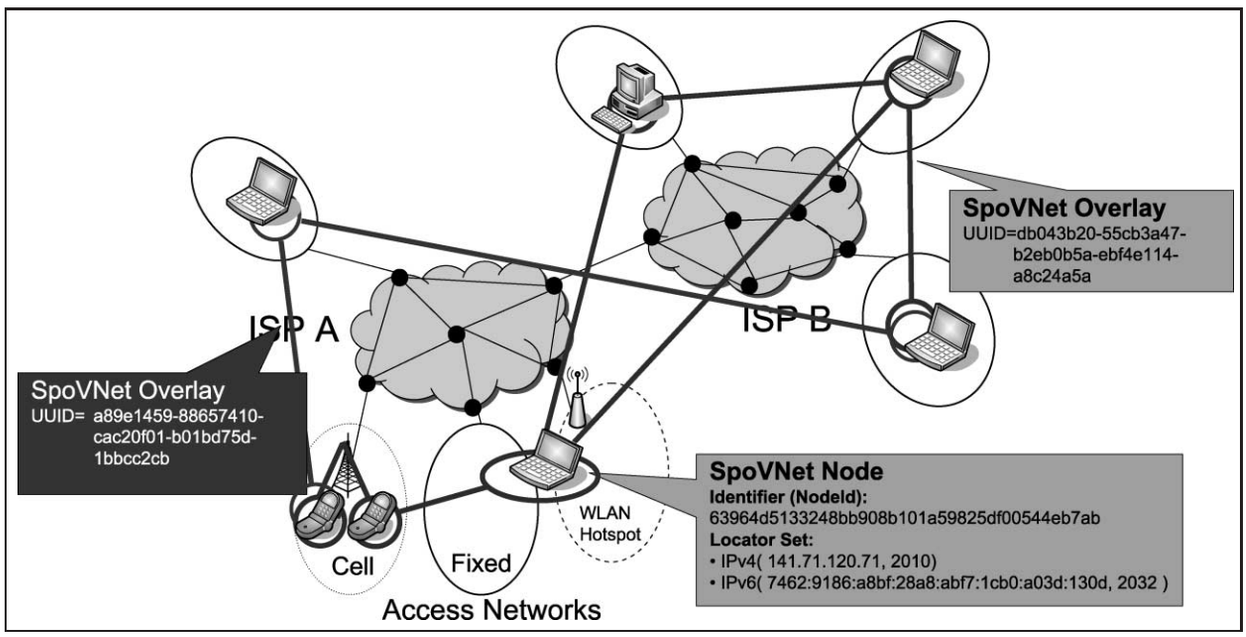

Figure 1 Two SpoVNet Instances in a heterogeneous underlay.

SpoVNet fosters the development of future Internet services and applications using overlays. To this end, the SpoVNet architecture provides two abstraction layers, as shown in Fig. 2. The lower abstraction layer – denoted as *Underlay Abstraction* – eases overlay construction by providing persistent transport links in spite of mobility, multi-homing, and heterogeneous network technologies. The upper abstraction layer – the *Service Abstraction* – is made up by interfaces provided by the services. Adaptation of overlay services is supported by providing cross-layer information of the underlying network. If the underlay lacks QoS support, SpoVNet offers probabilistic QoS guarantees by monitoring transport links. Furthermore, SpoVNet persues an integrated security approach by considering security as an integral part of future networks.

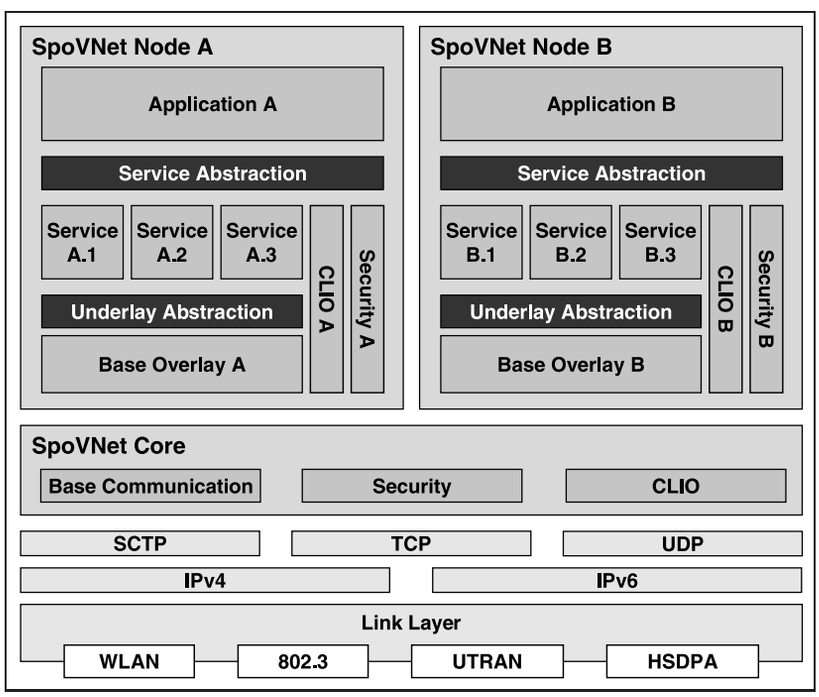

Figure 2 The SpoVNet architecture.

3.2 Abstraction Layers

SpoVNet's *Underlay Abstraction* is the key concept to address SpoVNet's objective (1). It eases the creation of overlay-based services by hiding tasks like connection maintenance – in spite of mobility and multi-homing – from higher layers.

The Underlay Abstraction comprises two components working together to provide basic transport connectivity: the *Base Overlay* and the *Base Communication*, as shown in Fig. 2.

The *Base Overlay* is implemented in each SpoVNet Node component and connects a node to all other nodes in a SpoVNet Instance. The SpoVNet Underlay Abstraction implements an identifier-locator split: Higher layer components such as SpoVNet applications and services only use *Base Overlay Identifiers* that uniquely identify a node within a SpoVNet Instance. Therefore, higher layer com-

ponents never get in touch with actual network locators (i. e., IP address and port). The construction of Base Overlay Identifiers is based on the *Cryptographically Generated Addresses* (CGA) [1] scheme. Using CGA, a cryptographically provable binding of identifier to node is accomplished.

When a service or an application wants to establish a link to a remote SpoVNet node, it requests link establishment from the Base Overlay. Subsequently, the Base Overlay maps the identifier of the remote node to one of possibly multiple locators using identifier-based routing. It requests link establishment from the Base Communication using the locators and returns a link handle which is used for the actual communication.

The *Base Communication* is implemented in the SpoVNet Core component, i. e., it runs once per SpoVNet device. The main task of the Base Communication is to create and maintain direct transport links between SpoVNet nodes, based on higher layer requirements like, e. g., QoS, and connection properties like reliability and security. Requirements are passed from the Base Overlay to the Base Communication that chooses a proper transport protocol (e. g., TCP, UDP), appropriate network locators if communicating nodes are multi-homed, and uses QoS signaling (e. g., NSIS [10]) if available. If the underlay does not support QoS reservation, the Base Communication provides a best-effort service and returns probabilistic QoS guarantees based on cross-layer information as feedback to higher layer components.

In order to handle connectivity between heterogeneous layer 3 technologies (e. g., IPv4, IPv6) and cope with NATs, the Base Communication provides transparent relaying to ensure connectivity. If link maintenance fails, e. g., when QoS guarantees can no longer be provided due to mobility, higher layers will be notified. More details on the Underlay Abstraction are given in [4].

SpoVNet's second abstraction is the *Service Abstration*, which adresses objective (2) by the use of persistent interfaces. These allow the exchange of SpoVNet services by native services in the underlay, and therefore enable a seemless transition towards future networks.

In order to support future services, the Service Abstraction must be extensible. To this end, once the demand for a new service is identified, the service interface is fixed in a first step and service functionality is implemented by an overlay-approach within the SpoVNet Service Layer on top of the Underlay Abstraction.

Ideally, the service proves of value and becomes part of the core functionality of future generation networks. Then, service functionality on the SpoVNet Service Layer can be reduced to a stub that transparently employs network functionality without changes to the applications.

3.3 Providing Cross-Layer Information

SpoVNet applications and services create their own, tailor-made overlays on demand. Their requirements may differ widely. Multicast video streaming, for instance, has different QoS requirements than massively distributed online games. The overlays thus need to employ different optimization techniques to meet specific needs. SpoVNet addresses this with a special component called *Cross-Layer Information for Overlays* (CLIO). CLIO is decentralized, i. e., there is no central point where information is collected and presented (as, e. g., in Ganglia [11]). Instead, this component runs on each SpoVNet Device, and aims to provide a view of the SpoVNet Instance and its nodes from this perspective.

Information about the network means static and dynamic properties of the underlay. Multicast/broadcast capabilities or connection types (e. g., WLAN, UMTS) are examples of static properties. Bandwidth, loss rate, and number of nodes in a SpoVNet Instance are examples of dynamic properties. Information about nodes means node-specific properties like latency (to or between nodes), available bandwidth, CPU load, available memory, or number of CPUs.

Information on performance parameters like achievable bandwidth and latency provided by CLIO may be expressed as a discrete distribution function. This way an overlay service can derive the mean value, the variance, or higher order moments as needed. In order to derive statistically significant distribution functions, large numbers of measurements are necessary. Since CLIO is aware of the type of underlay, the number of measurements can be reduced by using a priori information of the measured values through an underlay model. Simulations of Wireless LAN, e. g., show that latency can be approximated by a family of similarly shaped distribution functions that can be fitted to the measured values. The same method can be applied to estimate the throughput distribution. Since the throughput also depends on the packet sizes this needs additional measurements to estimate the average packet size.

CLIO attempts to minimize its impact on network and device resources. To this end, data is reused whenever possible. Furthermore, measurement queries are executed in an aggregated manner: If two queries from different applications or services can be subsumed into one (e. g., because one result-set is a superset of another), this is recognized and the measurement executed only once.

Furthermore, CLIO also pursues privacy-related goals. For example, it ensures that information does not leak between SpoVNet Instances, and access to critical data is controlled by policies.

CLIO has to know the properties of the underlay on the SpoVNet device for measurement. At the same time, it must be able to process queries by services and applications

that use node identifiers specific to a SpoVNet Instance. This suggests one component that runs within each SpoVNet Node – the *CLIO Node* – and a second component that is part of the SpoVNet Core – called *CLIO Core Component*, as shown in Fig. 2. Queries are passed from a CLIO Node Component to the CLIO Core, and node identifiers must be mapped to locators by the Base Overlay. The CLIO Core is then responsible for the actual execution of the query (e. g., measurements, monitoring, and pre-processing of data).

CLIO enables reaction to changes in the underlay (e. g., handovers from LAN to UMTS). It provides the necessary information for the Base Communication to handle such situations transparently for applications or services. As a further benefit, bootstrapping and initial optimization of overlays can be speeded up: due to the re-use of data across several overlays, it is possible that the required information is already available due to prior measurements.

Although data is stored in a database, the interface presents it as a Data Tree (conceptionally similar to MIB trees in SNMP). Queries are either executed for key-value pairs or for whole subtrees (using XPath-like expressions). They can be either one-time, with periodical iteration or with a trigger that notifies subscribing services or applications of changes.

CLIO supports a pluggable architecture for data pre-processing. Example plug-ins constitute modules for the prediction of bandwidth and load, or statistics of connectivity. The latter are useful for QoS support: CLIO can generate network statistics and evaluate these to provide probabilistic guarantees for applications and services that depend on QoS.

3.4 Security Component

To prevent the security patchwork approach that has evolved in the current Internet architecture, SpoVNet pursues *integrated security*. This implies that security is an inherent part of the SpoVNet architecture, including SpoVNet services. The advantage of this approach is threefold: First, security is implemented as a common component in the SpoVNet architecture, removing code and data redundancy, and hence leaving less opportunities for errors and failures. Second, security mechanisms can easily and safely be accessed by all SpoVNet components in a sound manner. And third, node-specific security policies can be implemented more easily.

The *Security Component* copes with two major issues: First, it maintains security state that is used for securing communication between nodes. For example, if a secure confidential link is needed between SpoVNet nodes, the Security Component uses the cryptographic Base Overlay identifiers to perform an authenticated key exchange. The exchanged keys are stored in the Security Component and are used to secure connections maintained by the Base Communication. If, e. g., TLS is available on the device, it is used for transparent connection security. Otherwise, the Security Component provides its own connection security, possibly building upon other security mechanisms available on the device. Due to the fact that all sensitive information is stored inside the security component, it is possible to source this information out into a safe storage device, e. g., a Trusted Platform Module, or smartcard.

Second, the Security Component manages policies for authorization, authentication, and confidentiality. Policies are determined by the node that creates the SpoVNet Instance. To enforce policies, the Security Component provides an interface to control communication and link establishment between nodes in different scenarios, e. g., bootstrap, join, maintenance, and leave phases of SpoVNet nodes. Services and applications can use this interface when upcoming operations are sensitive to SpoVNet security.

It should be noted that SpoVNet services and applications may need security features that go beyond what the Security Component can offer. In this case, the required functionality must be provided by the service or application itself, using the functionality offered by the Security Component.

4 Services and Applications

The Service Abstraction provides an extensible set of services which serve as basic building blocks for applications – beyond basic end-to-end communication as already provided by the Underlay Abstraction. We show how SpoVNet's architecture is of value in the design of complex services which can cope with highly demanding application requirements. Moreover, we highlight how demanding applications can benefit by relying on these services.

4.1 SpoVNet Services

Currently, two SpoVNet services provide advanced communication facilities to applications:

First, the *Multicast/Multipeer-Overlay* (MCP-O) provides efficient and scalable group communication mechanisms. MCP-O can, e. g., be used for efficient video streaming to groups of users. The goal of MCP-O is to reduce network load and to speed up content delivery.

Second, the SpoVNet *Event Service* (ES) provides efficient event notification and detection of complex events. The *Event Service* allows single application instances of a distributed application to individually express their need for plain as well as correlated information.

4.1.1 Group Communication Service

The MCP-O service component in SpoVNet provides scalable group communication through the use of *Application Layer Multicast* (ALM). Furthermore, QoS and security mechanisms are employed to enable

global deployment of group communication.

Efficient and scalable distribution of multicast data is achieved by a hierarchical clustering approach that is based upon the primary idea of NICE [2]: the grouping of multicast members into hierarchical clusters. This approach requires low maintenance overhead as management traffic is mainly exchanged in a cluster and not group-wide. MCP-O extends the NICE approach in several points: First, it provides more flexibility by the use of network and infrastructure information from CLIO. Therefore, better clustering can be achieved and heterogeneity of the underlay taken into account. Second, grouping members that are capable of native IP Multicast into the same cluster can be used to exploit IP Multicast data distribution in the respective cluster.

An exemplary clustering of group members is shown in Fig. 3. This clustering is implemented by a metric that is based on different weighted properties p_i:

$$d : (x, y) \mapsto \sum_i a_i p_i(x, y),$$

where p_i represents properties like, e. g., attachment of the member to a broadcast-capable medium. Through this function d each group member x is assigned to a cluster with current member y that represents the best cluster for this node at the lowest level with respect to the properties p_i. Since the size of a cluster is bounded, clusters have to be split when the group grows and new levels are created. Therefore, in each cluster a cluster leader is identified that becomes member of the cluster at the next higher level. For the actual data distribution, the cluster leader hierarchy is used.

Depending on the intended use of MCP-O, different security mechanisms have to be employed. For instance, video data distribution needs confidentiality, authenticity and integrity in the majority of cases. Confidentiality is provided by encryption using cluster keys, whereby the rekeying scheme is designed to integrate seamlessly with the clustering approach of MCP-O. Special schemes like TESLA [12] handle the issue of authenticity and integrity in multicast scenarios efficiently and are deployed within MCP-O.

MCP-O highly benefits from the Underlay Abstraction that provides easy to use connection establishment and maintenance, as well as probabilistic QoS support. Finally, the SpoVNet Service Abstraction hides the underlying clustering and data distribution mechanisms and provides for seamless service usability.

The use of ALM in MCP-O introduces increased network load in comparison to *Network Layer Multicast* (NLM). Therefore, usage of NLM in dedicated MCP-O clusters can improve overall service performance. We now describe the utilization of NLM in Wireless LAN for special MCP-O clusters through modelling of the MCP-O service.

Wireless LAN (WLAN) multicast provides efficient data distribution through the use of radio broadcast. As multicast transmissions are not acknowledged, radio collisions are not detected and the reliability of the service is therefore decreased. As many applications require reliable data transport, a protocol to detect and recover transmission loss must be employed. This, on the other hand, leads to protocol overhead caused by acknowledgment packets sent by receivers. Therefore, a decision function must be employed to decide whether acknowledged NLM using WLAN radio or ALM is more efficient for a specific MCP-O cluster.

In case of NLM in WLAN, the optimal protocol parameters for acknowledgment timeout and transmission window size must be determined. Our simulations show that the decision whether to use WLAN multicast or ALM mainly depends on the number of receiving stations. As a result, MCP-O can select a multicast distribution mechanism based on the cluster size. The same holds true for the transmission window since a range of values were found to lead to good performance. Yet, setting the right acknowledgment timeout requires knowledge of the wireless medium utilization and the used WLAN standard, both of which are provided by CLIO. As the timeout value can be expressed as a function of before-mentioned properties, it can be adapted on-the-fly.

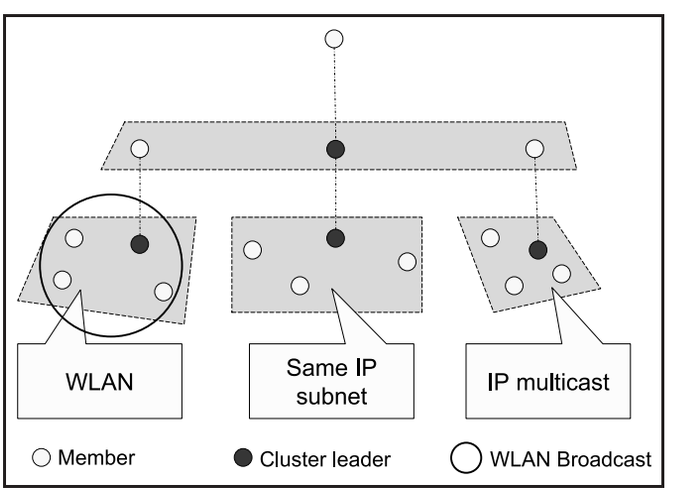

Figure 3 Hierarchical clustering in MCP-O.

4.1.2 Event Service

SpoVNet's Event Service decouples consumers and producers of information, where consumers express their individual interest via subscriptions similar to traditional content-based publish/subscribe sys-

tems [5]. The routing and filtering of event messages according to individual subscriptions and the event message content allows for a significant reduction of the overall load of information dissemination. A great support for applications is the in-network detection and composition of complex events. Instances of a distributed application can dynamically change the set of correlation rules, which trigger complex events once a correlation between events is detected.

In comparison to related approaches for distributed event notification and correlation (see [9; 13]), the SpoVNet ES addresses two main additional aspects: (1) the desired message transmission behavior as well as the data quality of the event messages can be dynamically managed by the application, (2) ES considers current device and link properties for the maintenance of the overlay structure and the placement of correlation functionality.

A SpoVNet node can interact with or contribute to ES by acting as a *broker*, *subscriber*, *publisher* or *correlator*. A node may dynamically change its role or act with respect to multiple roles.

Subscribers register their interest in form of subscriptions by a broker and are notified about corresponding publications from publishers. The set of brokers form a *broker overlay* which filters and forwards messages and ensures connectivity between subscribers and publishers. The broker overlay is maintained by a *publication-centric* and a *subscription-centric* adaption algorithm in order to comply with QoS requirements of subscribers, like those mentioned in [3]. Publication-centric adaptation determines a meaningful number and position of *cluster heads*, which partition the event space and serve as roots of content-based distribution trees embedded in the broker overlay. For instance, the algorithm could increase the number of cluster heads in order to achieve flatter distribution trees and thus reduce latency.

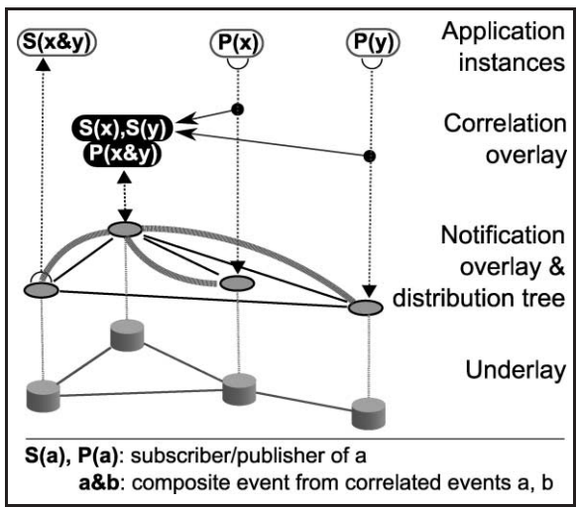

Figure 4 ES architecture. P: publisher, S: subscriber.

The distribution trees themselves are maintained by a subscription-centric adaptation algorithm. The algorithm ensures that brokers meet their individual QoS requirements and optimizes the capability to fulfill other brokers' QoS requirements by moving them bottom-up until they are sufficiently close to the cluster head.

The correlators form a *correlator overlay* by acting as subscribers to (basic) events and as publishers of composite events at the same time (see Fig. 4). They execute correlation detection on constantly updated and individually managed local event histories. Applications can define correlation rules which are decomposed and then placed in a self-organizing manner on suitable correlators considering the properties of links as well as the capabilities of nodes. The correlation functionality of a correlator may dynamically migrate in the presence of changing application requirements or changes to available resources of the network. SpoVNet's ES assures a dependable correlation behavior and determines the reliability of composed events (see [8]).

In general, ES provides a unified notion of probabilistic confidence that QoS requirements are met for message transmission behavior as well as for data quality.

The adaptation algorithms used in ES rely much on the advanced properties and services of SpoVNet. They access information about node and link characteristics provided by CLIO and use the SpoVNet Underlay Abstraction in order to route messages or negotiate QoS agreements. Moreover, they use the SpoVNet Security Component to implement fine-grained access control mechanisms that apply to each of the roles. Apart from that, the ES is also of benefit for other SpoVNet components like CLIO that collect or correlate changes in data.

4.2 SpoVNet Applications

Applications play a significant role in the development of services provided within SpoVNet. Currently two demanding applications are considered and developed, which take very different important real-time requirements into account: (1) A real-time *Peer-to-Peer based Massively Multiplayer Online First Person Shooter* (P2P-MMOFPS) that has high scalability and low latency requirements, and (2) a video streaming application that is adaptive to, e. g., network and QoS changes induced by mobility.

4.2.1 Real-time Game

In contrast to existing MMOFPS games, the game application developed for SpoVNet is peer-to-peer based. Several decentralized ap-

proaches exist [6], but most of them concentrate only on a selection of the most important challenges for such games: massive scalability, handling of game-dynamics, and consistency preservation.

Our approach uses the SpoVNet services ES and MCP-O in order to address these challenges. The game application defines an *Area of Interest* (AoI) for each player, and it subscribes to ES in order to detect which peers enter or leave an AoI. Information from corresponding event notifications is used for clustering multicast groups. In addition, many other criteria can be used to define multicast groups, e. g., friend or foe membership. Using these groups, the distribution of messages sent by the game application is limited to the AoI.

A specific feature concerns the quality of event transmission of the game. It is necessary to keep the state of the game consistent for all peers. Therefore, occurring events – like shots, hits or movements – have to be evaluated and delivered in at most 150 milliseconds to achieve an acceptable gaming experience. The game application is capable of complying to these requirements, because it can impose them on the SpoVNet services MCP-O and ES. These, in turn, rely on the Underlay Abstraction to enforce the corresponding network behavior, but they are also capable of achieving probabilistic service guarantees through independent adaptation of their overlays transparently to the application.

4.2.2 Video Streaming

The SpoVNet video streaming application uses MCP-O for scalability reasons, ES for adaptivity to underlay constraints, and mobility support provided by the Underlay Abstraction. To support simple transcoding during data dissemination the *Scalable Video Coding* (SVC) [14] is used for video encoding.

Consider multiple mobile devices, capable of high-quality video decoding. When using the SpoVNet multicast service MCP-O for data dissemination, all mobility is handled transparently by the Underlay Abstraction, unless the bandwidth changes significantly. In this case one option for the affected mobile devices is to send an event using ES to relevant peers along the path in direction to the video source in order to create a new multicast group with appropriate data rate. Filtering and aggregation provided by ES help to reduce the message load, in particular for the video source.

As another option when bandwidth degrades, MCP-O can support clustering of nodes with equal bandwidth capabilities by the use of an appropriate metric. This allows to transcode the video stream at the cluster leaders by simple truncation of video data, as supported by SVC.

Furthermore, the video streaming application can make use of the QoS information when available, concerning upper bounds for latency and jitter. It therefore allows to use small buffers in SpoVNet nodes.

5 Conclusions

Deployment of new services and applications is challenging in the current Internet architecture and has led to the introduction of half-layers and patchwork design. SpoVNet enables evolution of the underlying network architecture and protocols without changing services and applications. Several services and applications have been designed to prove the applicability of SpoVNet's concepts and its additional benefit on the way to the Future Internet. The integrated security approach and use of cross-layer information complete the SpoVNet architecture.

We are currently implementing the SpoVNet architecture and the introduced services and applications. The SpoVNet source code will be released under an Open Source model to contribute to the goals of the Future Internet community.

Acknowledgements

The SpoVNet Project is funded by Landesstiftung Baden-Württemberg under the initiative BW-FIT. The ideas and concepts presented in this paper are the results from work and discussion of all members of the SpoVNet project.

References

[1] T. Aura. Cryptographically Generated Addresses (CGA). RFC 3972 (Proposed Standard), Mar. 2005. Updated by RFCs 4581, 4982.

[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable Application Layer Multicast. In: *Proc. of SIGCOMM'02*, pp. 205–217, Pittsburgh, Pennsylvania, USA, Oct. 2002.

[3] S. Behnel, L. Fiege, and G. Mühl. On Quality-of-Service and Publish-Subscribe. In: *Proc. 26th IEEE Int'l Conf. Workshops on Distributed Computing Systems*, pp. 20–25, July 2006.

[4] R. Bless, C. Hübsch, S. Mies, and O. Waldhorst. The Underlay Abstraction in the Spontaneous Virtual Networks (SpoVNet) Architecture. In: *Proc. 4th EuroNGI Conf. on Next Generation Internet Networks (NGI 2008)*, Apr. 2008. CD-ROM.

[5] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and Evaluation of a Wide-area Event Notification Service. In: *ACM Trans. Comput. Syst.* 19(3):332–383, Aug. 2001.

[6] S. S. Chang, T. H. Chen, S. Y. Hu, and G. Liao. P2P-based Virtual Environment Research. http://vast.sourceforge.net/relatedwork.php.

[7] C. Jelger, C. Tschudin, S. Schmid, and G. Leduc. Basic abstractions for an autonomic network architecture. In: *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE Int'l Symposium*, pp. 1–6, June 2007.

[8] G. G. Koch, B. Koldehofe, and K. Rothermel. Higher Confidence In Event Correlation Using Uncertainty Restrictions. In: *Proc. 28th IEEE Int'l Conf. on Distributed Computing Systems Workshops (ICDCSW '08)*. IEEE Computer Society, June 2008.

[9] G. Li and H.-A. Jacobsen. Composite Subscriptions in Content-Based

Publish/Subscribe Systems. In: *Proc. of 6th Int'l Middleware Conf.*, LNCS, no. 3970, pp. 249–269. Springer, Nov. 2005.

[10] J. Manner, G. Karagiannis, and A. McDonald. NSLP for Quality-of-Service Signaling. Internet-Draft, Feb. 2008. draft-ietf-nsis-qos-nslp-16.txt.

[11] M. L. Massie, B. N. Chun, and D. E. Culler. The Ganglia Distributed Monitoring System: Design, Implementation, and Experience. Berkeley technical report, University of California at Berkeley, Feb. 2003.

[12] A. Perrig, D. Song, R. Canetti, J. D. Tygar, and B. Briscoe. Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction. RFC 4082 (Informational), June 2005.

[13] P. R. Pietzuch, B. Shand, and J. Bacon. Composite Event Detection as a Generic Middleware Extension. In: *Network, IEEE* 18(1):44–55, Jan. 2004.

[14] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H. 264/AVC Standard. In: *IEEE Trans. on Circuits and Systems for Video Technology* 17(9):1103–1129, Sep. 2007.

[15] M. Stiemerling et al. System Design of SATO and ASI. Deliverable D12-F.1, Ambient Networks Project, 2006.

Dr. Oliver P. Waldhorst received both a Dipl.-Inform. and Dr. rer. nat. degree from Universität Dortmund, Germany. He is currently leader of a young investigator group at the Institute of Telematics, Universität Karlsruhe (TH), Germany. His research interests include overlay, peer-to-peer and grid systems in future networks as well as performance modeling and analysis of self-organizing systems.
Address: Institute of Telematics, University of Karlsruhe (TH), Zirkel 2, Karlsruhe, 76131, Germany, Tel.: +49-721-6086403, Fax: +49-721-6086789,
E-Mail: waldhorst@tm.uka.de

Dipl.-Ing. Christian M. Blankenhorn received his degree from Universität Stuttgart in 2006. He is currently working as a research assistant and PhD candidate at the Institute of Communication Network and Computer Engineering of Universität Stuttgart. His research interests include application layer coding, wireless networks and performance evaluation.
Address: Institute of Communication Networks and Computer Engineering, University of Stuttgart, Pfaffenwaldring 47, Stuttgart, 70569, Germany, E-Mail: christian.blankenhorn@ikr.uni-stuttgart.de

Dipl.-Ing. Dirk Haage, M.Sc. received a Dipl.-Ing. degree from Technische Universität Berlin, Germany and an M.Sc. in telecommunication from Universidad Carlos III. de Madrid, Spain. He is currently a research assistant and PhD candidate at Universität Tübingen and a guest scientist at Technische Universität München. His research interests include measurement and monitoring in computer networks and distributed analyis of measurement data.
Address: Department for Computer Networks and Internet, University of Tübingen, Sand 13, Tübingen, 72076, Germany, Tel.: +49-7071-2970507,
E-Mail: haage@informatik.uni-tuebingen.de

Dipl.-Inform. Ralph Holz received his degree from Universität Tübingen, Germany. He is currently a research assistant and PhD candidate at Universität Tübingen, and a guest scientist at Technische Universität München. His research interests include security in massively distributed systems (in particular security protocols) and cryptography.
Address: Department for Computer Networks and Internet, University of Tübingen, Sand 13, Tübingen, 72076, Germany, Tel.: +49-7071-2970579,
E-Mail: holz@informatik.uni-tuebingen.de

Dipl.-Inform. Gerald Georg Koch received his degree from Universität Stuttgart. He is currently a research assistant and PhD candidate at Universität Stuttgart.
Address: Institute of Parallel and Distributed Systems, University of Stuttgart, Universitätsstr. 38, Stuttgart, 70569, Germany, Tel.: +49-711-7816-296,
E-Mail: gerald.koch@ipvs.uni-stuttgart.de

Dr. Boris Koldehofe graduated at Universität des Saarlandes, Germany, and obtained his PhD at Chalmers University of Technology, Sweden. He is working as a lecturer and senior researcher at the IPVS, Universität Stuttgart. His research interests include peer-to-peer computing and complex event processing.
Address: Institute of Parallel and Distributed Systems, University of Stuttgart, Universitätsstr. 38, Stuttgart, 70569, Germany, Tel.: +49-711-7816-357, E-Mail: boris.koldehofe@ipvs.uni-stuttgart.de

Fleming Lampi, M.Sc. received his degree in computer science and multimedia from the University of Applied Sciences in Karlsruhe, Germany. Currently, he is a PhD student and research assistant at the Department of Computer Science IV, Universität Mannheim, Germany. His research interests include video recording, processing, and transcoding.
Address: Department of Computer Science IV, University of Mannheim, A5, 6, Mannheim, 68159, Germany, Tel.: +49-621-181-2411, Fax: +49-621-181-2601,
E-Mail: lampi@informatik.uni-mannheim.de

Dipl.-Inform. Christoph P. Mayer graduated from Universität Karlsruhe in 2007 with a degree in computer science. In 2008 he joined the Institute of Telematics at Universität Karlsruhe (TH), Germany, as scientific staff. He is working on security for future networks in the BW-FIT project SpoVNet. He is the author of the PktAnon packet trace anonymization tool and the Distack framework for distributed attack detection.
Address: Institute of Telematics, University of Karlsruhe (TH), Zirkel 2, Karlsruhe, 76131, Germany, Tel.: +49-721-6086415, Fax: +49-721-6086789,
E-Mail: mayer@tm.uka.de

Dipl.-Inform. Sebastian Mies received his degree in computer science from Universität Karlsruhe (TH), Germany, in 2006. He is working at the Institute of Telematics at Universität Karlsruhe. His research interests are in the field of overlay networks, routing stategies and security mechanisms.
Address: Institute of Telematics, University of Karlsruhe (TH), Zirkel 2, Karlsruhe, 76131, Germany, Tel.: +49-721-6086402, Fax: +49-721-6086789,
E-Mail: mies@tm.uka.de