

Information Management for Dynamic Networks

Kostas Pentikousis
VTT Technical Research Centre of Finland
Kaitoväylä 1
FI-90571 Oulu, Finland
kostas.pentikousis@vtt.fi

Raffaele Giaffreda
British Telecommunications plc
Adastral Park, Martlesham Heath, Ipswich
Suffolk IP5 3RE, U.K.
raffaele.giaffreda@bt.com

Eleanor Hepworth
Roke Manor Research Ltd
Romsey
Hampshire SO51 0ZN, U.K.
eleanor.hepworth@roke.co.uk

Ramón Agüero
Department of Communications Engineering
University of Cantabria
Avda. los Castros s/n
39005 - Santander, Spain
ramon@tmat.unican.es

Alex Galis
Department of Electronic and Electrical Engineering
University College London
Torrington Place
London WC1E 7JE, U.K.
a.galis@ee.ucl.ac.uk

Abstract— The proliferation of mobile devices and the bundling of several network access technologies in a single product have increased considerably the communication environment complexity. There has been a certain drive for allowing end users to be “best-connected” anywhere, anytime, which places an additional set of stringent requirements on the already over-engineered domain of mobile computing. This rapidly increasing computational complexity is reinforced by the mounting number of wirelessly connected mobile devices and, therefore, network solutions are sought after with several self-X properties (self-configuration, -adaptation, and -management, to name a few). Such networks will require services that reliably cater the latest information aiming at simplifying and fostering autonomic decision-making. We introduce the Dynamic Networks Information Service Infrastructure (DNISI), an application of the information service architecture developed in the Ambient Networks project. We present how DNISI supports both enhanced mobility management and context-aware communications in today’s pervasive networking environments by providing the means for gathering, correlating, and managing cross-layer and cross-domain information. Finally, we explain how service platform components and end-users applications can benefit in the near-term.

I. INTRODUCTION

A cornerstone of forthcoming communication systems operating optimally in highly dynamic environments is information gathering and aggregation. For example, the Internet Engineering Task Force (IETF) released a number of RFCs related to collecting and managing information about different parts of the protocol stack, see, for example, [1]–[3]. Leveraging information to support mobility management decisions will be instrumental in optimizing network resources usage, while maximizing the user-perceived quality of communication services and applications. Nonetheless, the value of a comprehensive information service is often counterbalanced by the intrinsic difficulty in collecting data across different layers and locations and the corresponding overhead [4]. We are interested in designing, developing, and evaluating an information service infrastructure that takes advantage of existing data stores; can collect, filter and correlate events; and provision

context to interested system components and applications, facilitating mobility management. Information services will be essential in providing the supplementary context on which to base decisions, such as, whether to handover and to which network, and they can enable dynamic application configuration, support media adaptation, and deliver the best possible performance in the current network environment.

Fig. 1 presents the information services evolution for mobility management and provision of context-aware communications. It is borrowed from [5], which details the steps in this evolution, and is included in this paper mainly for completeness. As we progress in this timeline from GSM cellular networks to the recently proposed IEEE 802.21 base standard [6], we observe the growing need for a more comprehensive information service which can support mobility management. Effectively, we transition from using solely single-technology layer 2 (L2) information to a more information-hungry mobility management system. We argue for an information service infrastructure that pushes the envelope a bit further and encompasses dynamic in addition to the static Information Elements of IEEE 802.21. The main contribution of this paper is the introduction of the Dynamic Networks Information Service Infrastructure (DNISI), which aims at gathering and correlating information from different layers of the protocol stack and across different domains. DNISI is a focused application of the architecture presented in [5] and can be applied in today’s pervasive networking environments.

This paper is organized as follows. Section II overviews previous work related to DNISI and briefly describes the wider Ambient Networks architecture within which our information system is developed and tested. Sections III and IV present and evaluate DNISI, respectively. Finally, Section V concludes this paper and outlines ongoing and future work items.

II. AMBIENT NETWORKS AND RELATED WORK

The Ambient Networks Integrated Project (www.ambient-networks.org) is developing new networking concepts for

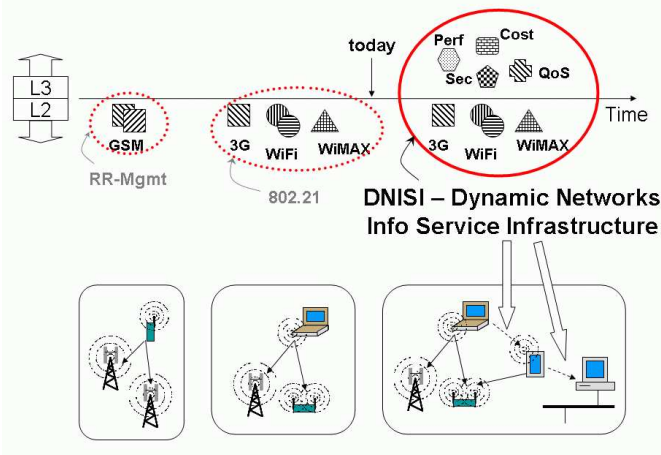


Fig. 1. Temporal evolution for information services needed to support mobility management and context-awareness

future wireless and mobile networks [7]–[10]. A key aspect of the project is to support dynamic composition of networks to establish a common control layer for various network types. This common control layer enables a more “plug and play” style of internetworking introducing sophisticated network features, as well as more dynamic business relationships between network providers. Ambient Networks (AN) aim at being scalable and flexible, able to “compose,” i.e., negotiate and agree on-the-fly across different administration domains (something not possible in today’s networks), and able to reconfigure themselves in a self-managed way. DNISI can assist all these tasks by gathering and correlating information from different layers of the protocol stack, including several components of the Ambient Control Space (ACS), and across different domains. As we will explain in the following section, DNISI supports both enhanced mobility management and context-aware communications in pervasive networking environments.

IEEE 802.21 identified the requirement for better information services to support media independent handover decisions, and is working towards an initial standard to support this functionality. This includes investigating network access interface-related aspects, similar to the Ambient Resource Interface (ARI). In AN, the ARI enables the ACS to deal with subjacent radio access technologies and internetworking procedures in a homogeneous manner. On the other hand, the scope of IEEE 802.21 base standard does not include the higher layer aspects considered by the Ambient Service Interface (ASI), which enables both applications and services to use the functionality provided by the ACS. Furthermore, the current information sets defined by the base specification are static and less diverse than those handled by DNISI.

The benefits enabled by more sophisticated information services have led to a number of work items in this area. The discovery of roaming agreement information pre-authentication has been recognised as a key aspect to supporting seamless user roaming between different networks, and a number of solutions have emerged including those developed within the

IEEE by IEEE 802.21 and IEEE 802.11, and also within the IETF [11]. For example, IEEE 802.11 is working on a number of amendments to its base standard to support discovery of information such what access points (APs) are in the local neighborhood [12], what mobility domain a particular AP is a member of [13], and what roaming agreements are in place between one WLAN access network and a number of service providers [14]. Support for discovery of this information in the IEEE 802.11 standard allows user devices to find out this information up front before authenticating with the network, which in turn allows better selection of which point of attachment should be used for communication. IEEE 802.21 [6] has also extended the information service to include other information to support handover decisions, including L2 events and commands, and the delivery of this information across a network. Nevertheless, the IEEE 802.21 base standard has not been finalized yet and does not specify the intelligence that will take advantage of the information gathered.

Finally, and in addition to the multi-interface aspects that have been mentioned above, it is expected that the use of multi-hop network topologies will play a key role in forthcoming 4G communication scenarios, as an extension to more traditional deployments [15] as in, for instance, the rapid growth of the meshed networking paradigm (IEEE 802.11s). In these topologies, where end-users will assist in forwarding traffic thus allowing others to reach network elements, topological changes are to be anticipated, since users may move freely. Furthermore, the entire network could also be on the move (for example, in a train) rising correlation between consecutive mobility events. DNISI can deal with the challenges that appear in these dynamic environments, bringing about the possibility to quickly react upon changes, while decoupling this functionality from the routing protocols per se, which are always seeking the best connectivity options.

III. DNISI – DYNAMIC NETWORKS INFORMATION SERVICE INFRASTRUCTURE

DNISI supports network attachment and handover decisions with a diverse information set about network characteristics and trigger events, described in §III-B, below. Through its collection, management, and maintenance of up-to-date information, DNISI can also be used when configuring applications or adjusting media delivery. DNISI relies on context managers and a distributed storage system called Context Information Base (CIB) to make ACS network functional entities, services, and applications using the ASI, context-aware. Context managers are distributed processes that can be dynamically created, based on context client requirements, providing aggregation, translation, and inference capabilities. The CIB is managed using data distribution algorithms that, through specialized context managers, enforce optimal dissemination of information between context sources (entities providing one or more context objects) and clients through usage of a distributed set of stores. The objective of this optimization is not only to guarantee timely delivery, or update of context information, but also to minimize the traffic generated by sources updating

context and clients requesting it and, thus, the overall system overhead. This is achieved by accounting for the rates at which those updates and requests occur when deciding the best place to store a particular piece of context information.

At the center of DNISI lies ConCoord, dealing with context coordination; it is the first port of call for both sources willing to register the information they want to publish and clients willing to retrieve context information. ConCoord primarily manages lower-churn information originating from the higher layers of the protocol stack. It corresponds to a distributed registry that maps Universal Context Identifiers (UCIs) to the location of context information objects. UCIs are a new type of Uniform Resource Identifiers (URI) [16], which uniquely identify a given context object, but not its location within the network. UCIs act as conceptual rendezvous points between context clients and sources; they define a new URI scheme (“ctx”) with the following format

ctx://host.example.com/path?options

Context sources send UCI REGISTER requests to ConCoord, thus actively disseminating pointers to their context objects via ConCoord. Context managers, once created, also REGISTER their output type and capabilities with the ConCoord. Context clients, on the other hand, need to send UCI RESOLVE requests to ConCoord, in order to be able to access context information; each request may contain more than one UCIs. ConCoord is responsible for maintaining UCI-location mappings for all context sources and managers. This enables recursive multi-pipe establishment in a distributed way for capability reuse. In fact, upon receiving a request for a particular UCI, the ConCoord locates the final context manager but if the UCI refers to aggregated context, the resolving process might also involve the context managers for its input, which in turn locate the managers, and so on, until the inputs are all initial objects (i.e. basic context sources). After checking that the client is authorized to issue such a request (see §III-A, below), ConCoord responds by returning the locations of the corresponding context objects.

After receiving the RESOLVE response, context clients contact the located context sources directly to GET the information, or to SUSBScribe to context change events by receiving NOTIFICATIONS. The rationale for this design is to leave context information, which might change frequently, at the source, until it is actually requested by some client. A source REGISTERs an object only once, and a client RESOLVEs each object UCI only once. Any further interaction is then done between clients and sources directly, as we illustrate in §III-B. Of course, it is also possible to cache context information on behalf of sources and at most appropriate locations in the CIB to address scalability, performance optimization, and minimize retrieval time from clients and update time from sources, as well as overhead traffic. The exact performance gains from such an optimization are under investigation.

To sum up, ConCoord implements a distributed registry where context sources register their UCIs along with their contact information, and which can be used by clients locate con-

text objects. Next, we introduce how ConCoord can support authentication and authorization for both source registrations and client access to context objects.

A. Security, Trust, and Policies

DNISI supports secure source/consumer authentication and context information management including caching, aggregation and dissemination. Once an information source (or consumer) has been admitted to the DNISI trust domain, it gains access to all services rendered by it. Recall that context sources and clients may be running (a) on the same node, (b) within the same domain, or (c) in different domains. In the first case, we assume that system components, services, and applications running in the same process space are by default “authenticated.” There is no need to authenticate, for example, applications already running in a laptop; a malicious application can do more harm than providing or accessing context information via DNISI. Authorization, on the other hand, is established using policies described in XACML [17]. The hierarchical policies defined for a DNISI trust domain permit users and operators to allow certain clients to access context content while securely denying access to others. XACML-based authorization is also used in cases (b) and (c) above. Authentication of context sources and clients in cases (b) and (c) is part of our ongoing work in Ambient Networks, as it depends to a large extent to the wider AN system requirements. Currently, we are investigating the benefits of using host cryptographic tags (for example, in a similar fashion as in the Host Identity Protocol [18]).

B. Mobility Management Triggers

As mentioned earlier, DNISI plays a fundamental role in assisting mobility management. Mobility management is mostly interested in changes of state in different components, as opposed to what is the current state. In many cases, see, for example [19], it is important to be notified when a network access becomes available. Conventional event sources include, for example, the radio interface (reporting events associated with radio access characteristics, such as, current or average network capacity load, signal-to-noise ratio (SNR), dropped frames ratio, received signal strength indication (RSSI), and so on); the battery state of charge, when not powered from the grid; and even processor loads and storage quotas. Other notable events can occur at higher layers of the protocol stack, for example, due to policy violations and security alerts, breaches in privacy agreements, changes in charging, and mobility protocol [20], [21] state transitions, and can all be reported by designated sources.

This section introduces the TRG context manager, which receives status changes from other entities, referred to as event sources. Fig. 2 illustrates the messages exchanged between TRG, ConCoord, and a typical event source/trigger consumer pair. Event sources must first authenticate themselves, in order to become part of the DNISI trust domain, and register with TRG (Fig. 2-1:) before they can start sending event notifications. Whenever a producer observes an event worth

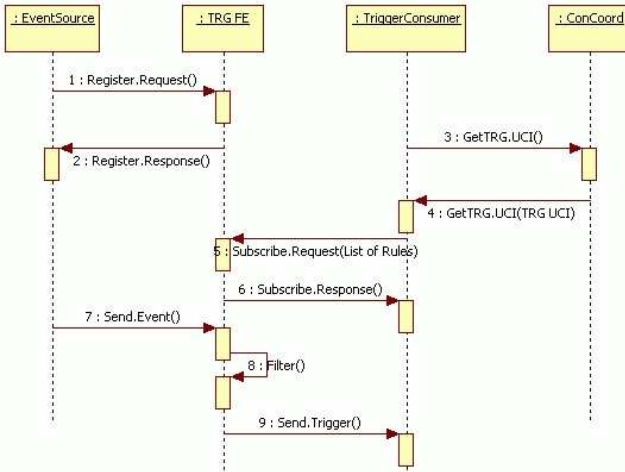


Fig. 2. DNISI TRG/ConCoord message exchange diagram

reporting, it will send it towards TRG using this interface (7:); the sampling rate is something that the source needs to determine and is beyond the scope of TRG per se, but can be configured. TRG processes the information received from the event sources based on a set of rules and policies, and generates “triggers.” The term trigger refers to a NOTIFICATION sent by TRG to a particular consumer (9:) based on the latter’s specified preferences and filtering rules (8:). Triggers are sent in a standardised format, as explained in [5], [22].

Context clients, i.e. trigger consumers, include firstly the handover decision-making process, but also user applications, protocols, and components interested in optimizing their performance in a mobile, multi-access network environment. When clients use ConCoord to RESOLVE UCIs REGISTERed by TRG (3:), they receive the location of TRG (4:) and can then proceed with subscribing (5:) and start (6:) receiving DNISI NOTIFICATIONs as standardized triggers (9:). Along with their subscription, clients can inform TRG to apply certain filtering rules. For example, a mobility protocol can take advantage of triggers about the activation of a particular link only, about crossing a threshold in the battery state of charge, the received signal strength, or any combination of these. In a wireless sensor network, the current gateway nodes running DNISI may decide to kick start the process of electing alternative nodes to serve gateways if their traffic load is too high and their battery state of charge is too low.

The consumer filtering rules are settled according to the specific needs of each consumer. Policies (§III-A) are a critical component of DNISI and take precedence over the filtering rules specified by any particular consumer. For instance, security aspects, such as, which sources and consumers can be considered as trustworthy (and for which types of triggers) and what types of information should not be delivered to certain consumers (for example, different operators may not allow that the current load is delivered to another operator) are taken care by introducing the proper XACML policy.

Although TRG falls under the general category of notification systems, certain characteristics make it unique. TRG deals explicitly with mobility-related events, and any other information that can assist handover (HO) decision making, typically runs in resource-limited devices under stringent time constraints and is required to deliver triggers expeditiously, using standardized APIs based on well-defined and versatile, yet compact data structures suitable for handover management, and is policy-driven. To sum up, the TRG context manager comprises three main primitives: (a) event sources, which feed TRG with relatively fast-changing information; (b) trigger consumers, which receive notifications in the form of standardised triggers about events they are interested about; and (c) the associated data stores and internal logic.

C. Discussion

DNISI is designed with a particular aim of being flexible enough to address a number of basic requirements for a generic information service, scalable, and able to cope with high dynamics as well as a more AN-specific requirement related to composition of Ambient Networks. Flexibility is needed to accommodate the wide variety of information that can be classified as network context (including the events mentioned in III-B), whereas scalability is meant to address the huge numbers of individual items from most disparate sources that potentially need to be managed for the development of comprehensive context-aware services and applications. Furthermore, network composition introduces another dimension to the problem, which consists of properly merging and de-merging information bases.

To account for scalability goals, distribution is being considered for the building blocks briefly introduced above. In particular, to scale up ConCoord, Distributed Hash Tables (DHTs) [23] are used, whereas other features mentioned above (such as the recursive use of context managers) also address scalability. Moreover, as explained earlier, after the ConCoord lookup, context clients communicate directly with context managers in a peer-to-peer way since clients can be dynamically equipped with the right protocol needed. By removing ConCoord from the client-source path we increase the scalability of DNISI.

With respect to flexibility, DNISI can take advantage of the solution proposed in [5] where dynamically created ad-hoc context managers can, for instance, sanitize, reformat, or aggregate information bringing it into compliance with the standardized TRG APIs, thereby enhancing the quality of context delivery when additional sources and consumers are introduced.

IV. EVALUATION AND FUTURE WORK

Our approach in developing and rolling out DNISI follows a step-by-step process. One of the original third party concerns was the feasibility of implementing TRG in resource-limited mobile devices. We built a prototype using PDAs and laptops and successfully demonstrated [24] that TRG can be used to initiate a session handover based on triggers originating from a sensor box providing position and orientation information.

The demo comprises different types of devices, as it may be the case in forthcoming personal communications and allows the user to transfer a video stream from her PDA to some wide screen at his home. In particular, filtering rules were provided by the video client application (based on VLC, available from www.videolan.org/vlc/), and according to the position of the PDA, the on-going session is transferred from/to the PDA to/from a laptop (which has a larger screen). In this illustrative example, changes in orientation measured by the sensor box triggered the HO, but other triggers, such as, a new access point becoming available, RSSI crossing a predefined threshold, to name a few, could have been used to initiate such mobility-related actions. Interested readers can find more details and testbed measurements in [22].

On the other hand, the ConCoord implementation is based on a Distributed Hash Table (DHT). DHTs are scalable, decentralized registries that map a set of keys among participating nodes to certain stored values. The protocols in DHTs can efficiently route messages to the unique owner of any given key to find the corresponding value. DHTs are typically designed to scale to large numbers of nodes and to handle continual node arrivals and failures [23]. They have been proposed as a generic building block for many large-scale distributed applications, such as [25], [26]. In short, DHTs form a routing overlay on which requests for data entries are routed toward the nodes currently managing them [23]. Each node in a DHT is assigned a unique node ID from a defined keyspace (usually 2160 keys large). Each piece of data that needs to be stored in a DHT is assigned a key. A key is a fixed length hash. In the Bamboo DHT implementation (available from www.bamboo-dht.org) the data values are stored on the node whose node ID is closest to the data value's corresponding key and also replicated at a number of other nodes. Requests for access to the data stored in a DHT are routed to the node ID closest to its key and will be routed in $O(\log N)$ hops, where N is the number of nodes in the DHT [25].

The performance of the individual DNISI primitives (REGISTER, RESOLVE, GET, SUBSCRIBE, NOTIFY, etc.) ultimately depend on the performance of the *get* and *put* primitives in the underlying DHT, given their operational mapping as discussed in the previous section. Both *get* and *put* operations in the DHT require the requesting node to contact a known node in the overlay and have the overlay internally route the requested operation to the node that owns the zone to which the hashed key of the $\langle \text{key}, \text{value} \rangle$ pair maps. This is done by message passing on a hop-by-hop basis within the overlay, using its own routing mechanism.

To evaluate the performance of DNISI over a larger number of nodes than cannot be supported by the physical testbed, and to facilitate the collection of statistics, we used a version of our software that can simulate the operation of a large number of virtual nodes on a much smaller number of actual physical nodes. Due to space restrictions we present in Fig. 3 results about the average and worst-case latency of the ConCoord RESOLVE request in terms of the number of application-level hops within the DHT overlay. As with the basic DHT

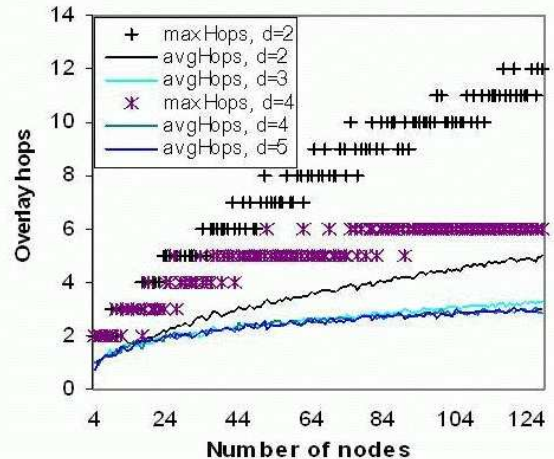


Fig. 3. Effect of overlay size and dimensions on RESOLVE latency

design, the routing path length is influenced by the number of dimensions (d) used in setting up the key space, since for a given number of nodes (N) in an overlay, a key space set up with more dimensions would result in a larger number of neighbors per node than a space having fewer dimensions. A larger number of neighbors per node would result in a shorter path length on average, but would require each node to maintain more states. In our case, for a moderately-sized AN with 128 nodes, there seems to be a significant improvement both in the average and worst-case RESOLVE latency with an increase in the number of dimensions from $d = 2$ to $d = 4$.

To mitigate the effects of DHT overlay routing delay, we are evaluating various caching mechanisms and strategies. Since our current DHT model stores either context addresses or context client addresses rather than the context information itself, the relatively static nature of the pair $\langle \text{key}, \text{value} \rangle$ lends itself well to caching, even in the face of numerous topological changes. Caching recently-obtained $\langle \text{key}, \text{value} \rangle$ pairs by overlay nodes has been suggested in [25] resulting in what might be analogous to “non-authoritative replies,” and local caching may also be done by context clients themselves. A time-to-live (TTL) parameter could be specified by context sources and clients during registration and subscription, respectively, to provide a metric for the expiration of cache entries. We are also studying the impact of simultaneously routing a get within the overlay, even in the event of a cache hit, for the purpose of validating or invalidating a cached entry. We plan to evaluate the use of various cache coherency protocols as we further develop our caching strategy.

Another approach we are evaluating involves the use of cached source addresses of overlay nodes obtained from transaction replies or acknowledgments sent from the overlay to a client. For example, a successful RESOLVE request should be followed by a RESOLVEACK reply from the overlay node to the requesting client. If the client needs to contact the overlay again because its cached data has already expired, it could now

use the cached address of the overlay node as the initial known node. Our hypothesis is that even in the event that the address of a cached overlay node no longer validly maps to a key (say, because that portion of its zone has already been assigned to another node), that node may, with high probability, be located in within the immediate vicinity of the new owner of the target zone. Performance improvement is a major part of our ongoing effort.

V. CONCLUSION

Communication environments are becoming increasingly more complex due to the diversity of available network technologies and the proliferation of multi-function devices and non-conventional network deployments. It will be in such scenarios that creating consistent and up-to-date information services will enable more informed and intelligent decisions to be made automatically.

We introduced DNISI, an information service infrastructure designed to provide services and applications at different layers with support for network information gathering, correlation and intelligent decision-making in support of enhanced mobility management and context-aware communications. Building on the design principles of Ambient Networks, DNISI features include the capability to gather information spanning different administrative domains, the ability to deliver triggers for advanced mobility management, and the opportunity to provide clients with relevant and up-to-date contextual information. To support applications and services at various layers with a very large and diverse knowledge base, the challenge lies in creating information services that can flexibly address such diversity and scalability issues. We are currently working on extensions of the existing prototype implementations, in order to assess the performance of DNISI in more diverse settings. Furthermore, we aim at capitalizing on the AN general framework, especially the interface of the ACS towards applications, addressing migration issues in detail, and going after standardization opportunities.

ACKNOWLEDGEMENT

This work has been carried out in the framework of the Ambient Networks project, which is partially funded by the Commission of the European Union. The enlightening and fruitful discussions with our colleagues in the AN project were instrumental in furthering our work. The views expressed in this paper are solely those of the authors and do not necessarily represent the views of their employers, the Ambient Networks project, or the Commission of the European Union.

REFERENCES

- [1] J. Case, R. Mundy, D. Partain, and B. Stewart. Introduction and Applicability Statements for Internet Standard Management Framework. IETF RFC 3410 (Informational), December 2002.
- [2] S. Waldbusser and P. Grillo. Host Resources MIB. IETF RFC 2790 (Standards Track), March 2000.
- [3] R. Raghunathan (Ed.). Management Information Base for the Transmission Control Protocol (TCP). IETF RFC 4022 (Standards Track), March 2005.

- [4] T. Buchholz and C. Linnhoff-Popien. Towards realizing global scalability in context-aware systems. In *Proc. of International Workshop on Location- and Context-Awareness (LoCA 2005)*, LNCS 3479, pages 26–39, May 2005.
- [5] R. Giaffreda, K. Pentikousis, E. Hepworth, R. Agüero, and A. Galis. An information service infrastructure for Ambient Networks. In *Proc. IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN 2007)*, Innsbruck, Austria, February 2007.
- [6] IEEE P802.21/D02.00, Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services, September 2006.
- [7] N. Niebert, A. Schieder, H. Abramowicz, G. Malmgren, J. Sachs, U. Horn, C. Prehofer, and H. Karl. Ambient Networks – an architecture for communication networks beyond 3G. *IEEE Wireless Communications (Special Issue on 4G Mobile Communications – Towards Open Wireless Architecture)*, 11(2):14–21, April 2004.
- [8] B. Ahlgren, J. Arkko, L. Eggert, and J. Rajahalme. A node identity internetworking architecture. In *Proc. IEEE Global Internet Symposium*, Barcelona, Spain, April 2006.
- [9] F. Hartung, N. Niebert, A. Schieder, R. Rembarz, S. Schmid, and L. Eggert. Advances in network-supported media delivery in next-generation mobile systems. *IEEE Communications Magazine*, 44(8):82–89, August 2006.
- [10] L. Ho, J. Markendahl, and M. Berg. Business aspects of advertising and discovery concepts in Ambient Networks. In *Proc. IEEE 17th International Symposium on Personal, Indoor and Mobile Radio Communications (PMRC 2006)*, Helsinki, Finland, September 2006.
- [11] F. Adrangi, V. Lortz, F. Bari, and P. Eronen. Identity Selection Hints for the Extensible Authentication Protocol (EAP). IETF RFC 4284 (Informational), January 2006.
- [12] IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification; Amendment 9: Radio Resource Measurement (IEEE 802.11k) D4.0, March 2006.
- [13] IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification; Amendment 2: Fast BSS Transition (IEEE 802.11r) D2.1, May 2006.
- [14] E. Hepworth. Network selection problem statement, 11-06-0542r1, April 2006.
- [15] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic. *Mobile Ad Hoc Networking*. Wiley-IEEE Press, August 2004.
- [16] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. IETF RFC 3986 (Standards Track), January 2005.
- [17] T. Moses (Ed.). OASIS Standard, eXtensible Access Control Markup Language (XACML) Version 2.0. February 2005.
- [18] P. Nikander, J. Ylitalo, and J. Wall. Integrating Security, Mobility, and Multi-homing in a HIP Way. In *Proc. Network and Distributed Systems Security Symposium (NDSS03)*, San Diego, CA, USA, February 2003.
- [19] S. Schütz, L. Eggert, S. Schmid, and M. Brunner. Protocol enhancements for intermittently connected hosts. *SIGCOMM Computer Communications Review*, 35(3):5–18, 2005.
- [20] C. E. Perkins. Mobile IP. *IEEE Communications Magazine*, 40(5):66–82, May 2002.
- [21] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. IETF RFC 4423, May 2006.
- [22] J. Mäkelä and K. Pentikousis. Trigger management mechanisms. In *Proc. International Symposium on Wireless Pervasive Computing (ISWPC 2007)*, San Juan, Puerto Rico, February 2007.
- [23] D. Ratajczak and J. Hellerstein. Deconstructing dhds. Technical report, Intel Research Berkley, IRB-TR-03-042, Nov. 2003.
- [24] J. Mäkelä, R. Agüero, J. Tenhunen, V. Kyllönen, J. Choque, and L. Munoz. Paving the way for future mobility mechanisms: A testbed for mobility triggering and moving network support. In *Proc. 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom 2006)*, Barcelona, Spain, 2006.
- [25] A. Rowstron and P. Druschel. Pastry: Scalable, decentralised object location, and routing for large-scale peer-to-peer systems. In *Proc. of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, LNCS 2218, pages 329–350, Heidelberg, Germany, November 2001. Springer.
- [26] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proc. of ACM SIGCOMM Conference*, San Diego, California, USA, August 2001.