AN INFORMATION SERVICE INFRASTRUCTURE FOR AMBIENT NETWORKS

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

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

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

Ramón Agüero
University of Cantabria
Avda. los Castros s/n
39005 – Santander
Spain
ramon@tlmat.unican.es

Alex Galis
University College London
Torrington Place
London WC1E 7JE
U.K.
a.galis@ee.ucl.ac.uk

ABSTRACT

Communication environments are becoming increasingly more complex due to the diversity of available network technologies in terms of spatial coverage and design characteristics, and the proliferation of multi-function devices. In order to take full advantage of such technology capital, there is a growing need to reduce complexity for both end-users and network operators delivering services over these ubiquitous communication environments. Recent research efforts have moved in the direction of creating solutions that facilitate self- properties (i.e. self-configuring, -adaptation -management, -optimisation, -organisation) in future networks. An important enabler underpinning such solutions is the availability of a reliable and up-to-date knowledge base to simplify and foster autonomic decision-making. We introduce the Ambient Networks Information Service Infrastructure (ANISI), which aims at gathering and correlating information from different layers of the protocol stack and across different domains. We show how ANISI supports both enhanced mobility management and context-aware communications in pervasive networking environments.

KEY WORDS

Ambient Networks, Context Management, Mobility Management

1. Introduction

Collecting information from a number of heterogeneous networks is proving more and more useful for supporting mobility management decisions that optimise the use of network resources, while maximising the perceived quality of communication services and applications. However, the usefulness of a comprehensive information service is often counterbalanced by the intrinsic difficulty in collecting data across different layers of the protocol stack and from different locations in the network [1].

Figure 1 illustrates the evolution of information services for mobility management and provision of context-aware communications during the last two decades. The first relevant example on our timeline is a cellular mobile telephony network based on the Global System for Mobile Communications (GSM) standard. In GSM, the Radio Resource Management (RR-Mgmt) layer collects information about radio links in contiguous cells in order to support seamless handover of voice communications for mobile users [2]. Note that RR-Mgmt gathers information only from link layer (L2) entities and does not consider collecting information from higher layer protocols (L3 and above).

With the advent of other access technologies that support both data and voice services, such as IEEE 802.11 (WiFi) and IEEE 802.16 (WiMax), however, the challenge is no

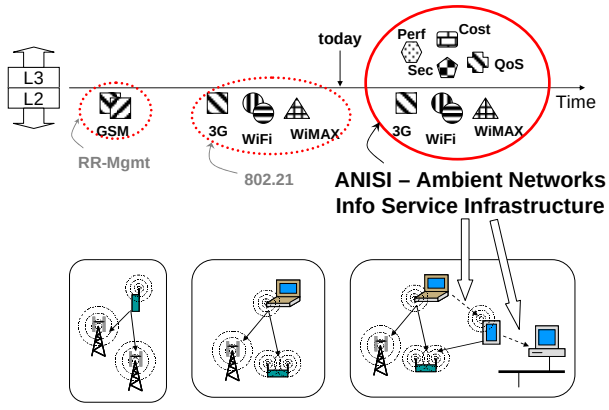

Figure 1. Information services evolution over time.

longer how to support mobility for voice services within the same technology (GSM, for instance), but how to develop enhanced mobility management that serves multimedia applications, often considering the opportunity of spanning a number of different access technologies.

As shown in the centre of Figure 1, for example, recent work within the IEEE 802.21 standards group has been investigating what information should be made available in a "media independent" way. Although the standard does not specify the intelligence that will take advantage of the information gathered, it is clear that the group is addressing the demand for a more comprehensive information service particularly, useful for the exploitation of all radio resources available. Nonetheless, the enlarged knowledge space specified by IEEE 802.21 is populated by predominantly static Information Elements gathered from different access technologies (spanning WiFi, WiMax, 3GPP, etc.) [3], and does not entirely meet the requirements of advanced services and applications at different layers, which could be enhanced and made context-aware if, in addition to the information from several radio link layers, dynamic and continuously refreshed information was available from the entire protocol stack.

Moving rightward on our progressing timeline (Figure 1) it is therefore straightforward to extrapolate how the cross-technology information gathering challenge we face today will be quickly be superseded by one of further enlarging network knowledge based on information gathered from different layers of the protocol stack and taking into consideration, for example, end-to-end Quality of Service (*QoS*) and performance (*Perf*) metrics, network costs (*Cost*) and compensation, and complying to security requirements (*Sec*).

2. The Ambient Networks Information Service Infrastructure

We believe that an enlarged information base is useful across multiple layers of the protocol stack, from network management applications to service components within service platforms as well as end-user applications (see Figure 2). Below we focus on how the Ambient Networks Information Service Infrastructure (ANISI) can enhance mobility management through the delivery of triggers and information exchange via context-aware communications that can account for *network* context too.

2.1 Requirements

As networks become more heterogeneous in nature, there is a trend towards more autonomic decision-making distributed across different nodes in the network. In order to make this more powerful and effective, especially from an end-user perspective, it is essential to provide a wide range of content information collected from multiple administrative domains. Moreover, to satisfy the requirements of these clients, the data contributing to such enlarged information base will exhibit a wide degree of temporal variability: there will be need to manage relatively static data as well as frequently changing pieces of information. The timely delivery of data in the higher part of the variability spectrum will demand a more distributed approach to information management than it is currently the case with existing information services.

Data dissemination techniques towards the above-illustrated clients will also impose different requirements on the information service infrastructure. Data will have to be pushed according to client specific filtering rules or pulled at client will with minimised retrieval delay. In light of such considerations and requirements, we propose ANISI, which provides support for distributed management of both, triggers for enhanced mobility management as well as more generic network context information aimed at ensuring longevity and a future-proof validity of our service. Apart from storing data in a way that addresses its dynamicity and heterogeneity, ANISI also provides interested network or service clients with the means for subscribing to receive specific information, event filtering, and processing before having the relevant data delivered. In particular, the ANISI ability to relate through inference, filter relevant data and notify changes based on clients' instructions inherently enables clients to become context-aware.

ANISI is being developed as a component within the wider Ambient Networks (AN) [4] architecture and as such, it benefits from some of its features. To be more precise, the AN architecture supports the additional functionality required by our information service in a number of ways. AN provide a modular environment within which functional entities can discover and communicate with each other, even for devices in different administrative domains. This inherently supports distributed data management, with devices able to locate the source of information they require in a straightforward manner. Deployment architectures are not constrained, and device failures are easier to handle. Moreover, AN provide a set of standard interfaces across which information from different layers of the protocol stack can be retrieved and exchanged between functions in different networks.

AN support for this functionality is briefly described in the following section, which presents a use case for

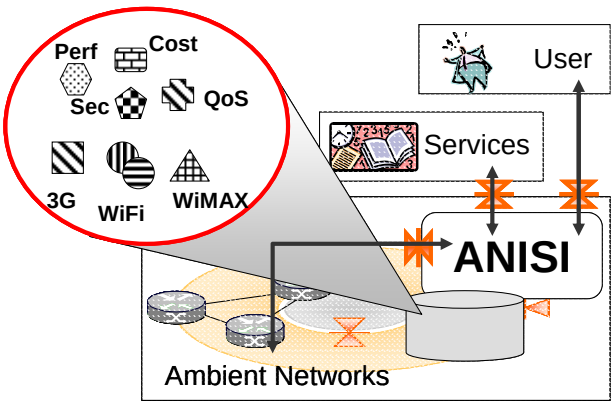

Figure 2. ANISI in the AN architecture.

ANISI illustrating how it can be used to improve the end-user experience and endow Ambient Network operators with the means to monitor usage of resources and dynamically instantiate features such as load balancing amongst own and available "friendly" networks. The use case aims at conveying *what* types of new opportunities become available with ANISI. *How* can this be achieved will be explained later with more use cases described after the ANISI features are introduced.

2.2 ANISI in Practice

The main actors in our scenario are Bob, an end-user, and a telecommunications operator called TelcoX. Bob is a subscriber of TelcoX, which operates its own network, but it can also rely on the availability of several other access networks to deliver its value-added services provided each of these networks is compliant with the features of an AN [4]. That is, TelcoX provides services across a *composite* network infrastructure partly self-owned and partly owned by other network providers. As soon as Bob powers-up his device(s), his presence on the respective networks is recorded and correlated via ANISI. TelcoX's value-added services rely on ANISI to gather up-to-date information on the capabilities and performance of the networks covering the area where Bob is located.

The ability to create dynamic roaming agreements via the composition procedures [4] amongst ANs allows correlation of information about networks on which Bob is present. This provides an indication of his location, which is then mapped onto the knowledge TelcoX has or can get through AN features about various networks within Bob's reach. Note that these are a superset of the networks he is actively connected to. The association *Bob's location* → *available networks* → *performance* becomes our key enabler for some of the features illustrated in the use case.

After logon, Bob's multi-homed laptop is made aware of the features of the Ambient Networks he has just attached to. In particular, this allows each of the applications he launches to receive customised information from ANISI about surrounding networks. As Bob moves through a number of Ambient Networks, his applications get notified via ANISI features that some networks suit more his preferences than others. As those events are notified, some of his running applications may get suspended and temporarily work offline, and then be resumed when suitable networks become available. As Bob gets on with his work, the network he is using starts to experience increasing traffic levels, which can either prompt Bob to handover to a different network, as we have seen, or it can trigger the network operator to discover potential networks for composition or to enforce load-balancing increasing availability of backhaul bandwidth, which in turn results in improved delivery quality for some of Bob's running applications.

In this example, although ANISI provides the necessary information it neither decides nor executes the handover. Instead, it is the handover decision logic on the device that makes the decision. ANISI enhances not only end-users applications but the operator's resource management system as well. Next we describe how this functionality is provided by ANISI by introducing the two fundamental building blocks: ContextWare and TRG.

3. ANISI: The ContextWare Building Block

ANISI defines a generic network context information management system, known as ContextWare, which collects, manages and keeps up-to-date information that might be used by other network functions, services or applications to make decisions. ContextWare comprises two major FEs as shown in Figure 3. The first is called Context Coordination Functional Entity (ConCoord FE) and 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. Rather than storing only pointers, as a directory service would do, ContextWare also provides support for authentication and context information management including caching, aggregation and dissemination. This is achieved through a number of context managers forming a Context Management Functional Entity (CM FE) and by a distributed storage system called Context Information Base (CIB). It is important to highlight that the CIB not only provides resources to those context sources willing to delegate the distribution and management of the context information they can provide, but also works as support storage for the operations of the various context managers.

3.1 ConCoord Functional Entity

ConCoord corresponds to a distributed registry that maps Universal Context Identifiers (UCIs) to the location of context information objects. ConCoord maintains this registry by receiving REGISTER requests from context sources (entities providing one or more context objects). Hence, sources actively disseminate pointers to their context objects to the ConCoord, which is the first point-of contact for context clients. When clients want to access context information, they send RESOLVE requests, which contain one or more UCIs to the ConCoord. After checking that the client is authorised to issue such a request [5], ConCoord responds by returning the locations of the corresponding context objects. UCIs are a new type of Uniform Resource Identifiers (URI) [6], which uniquely identify a context object, but not its network location. Then clients contact the located context sources

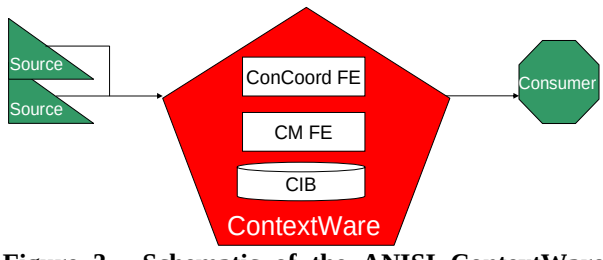

Figure 3. Schematic of the ANISI ContextWare building block.

directly to GET the information, or to SUSBSCRIBE to context change events by receiving NOTIFICATIONs. This design lets context information, which may change frequently, at the source, until it is actually requested. Sources register an object only once, and a client resolves each UCI only once. Any further interaction is then done between clients and sources directly.

The registry of the ConCoord is itself a context object, the meta-context object of all other context objects. The context source for this object is the ConCoord itself, and the meta-object is essentially the set of registered UCIs. This meta-object should also be accessible like any other context object by the ContextWare protocol primitives, as this enables context clients to subscribe to events like "notify me whenever a new object of this type registers". This is important information for context clients to detect new sources of context information, or to learn that currently used context sources are no longer available.

In short, ConCord is a distributed registry where a context source registers the UCIs of its context objects with its contact information, with a function to authenticate and authorise source registrations, and client access to context objects; and a function to resolve the UCIs required by context clients to the stored locations of context objects.

3.2 Context Management Functional Entity

As opposed to the ConCoord FE that maps UCIs to location of context information, the Context Management (CM) FE manages context within and across domains. More precisely, this is achieved through a number of specialised Context Managers, each of them dynamically created as the need arises for carrying out a particular task. TRG per se, for example, can be seen as a special-purpose network context manager. The Context Management FE therefore represents the service provided by a number of distributed processes that can be dynamically created based on context client requirements. It provides aggregation, translation, inference capabilities allowing reuse of those capabilities by many clients. The CM FE can also cache context information on behalf of context sources and at most appropriate locations in the CIB to address scalability, performance optimisation and minimise retrieval time from clients and update time from sources as well as overhead traffic.

Context managers, once created, register their output type and capabilities with ConCoord. ConCoord's registry therefore maintains mappings to location of context sources and of context managers. This also 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 involve also the context managers for its input, which in turn locate the managers for their input, and so on, until the inputs are all initial objects (i.e. basic context sources).

3.3 ContextWare Design Features

To explicitly meet some of the objectives of the Ambient Networks vision, this architecture was designed with a particular aim of being flexible enough to address a number of basic requirements for a generic information service, such as flexibility, scalability, ability 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*, 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 meet the goal of flexibility, context managers not only assume fixed roles as in the case of the ANISI TRG, but they can also play many different roles based on the need of clients. Therefore, not only they are dynamically reconfigurable, but also they can be recursively used as components to meet different clients' needs avoiding duplication of functionality in different stovepipes. This is achieved is by having each context manager publish the results of context information processing in the same way as any context source would do. Recursion allows context managers to become sources as well as clients for context information themselves. Likewise, to allow flexible interactions with these context managers in the CM FE, it is envisaged that the ConCoord lookup service might return the UCI-to-location mapping *and* any further information needed to communicate with the source associated with that UCI (say, for a SNMP object, to "equip" the client SNMP if the source of context is a MIB).

To account for scalability goals, distribution is being considered for the building blocks briefly introduced above. In particular, to scale up the ConCoord functionality, Distributed Hash Tables (DHTs) [7] are being used whereas for the CM FE, some of the flexibility features mentioned (like recursive use of context managers) are also addressing scalability. Also, where applicable, after the ConCoord lookup, context clients can directly communicate with context sources in a peer-to-peer way, due to the fact that clients can be dynamically equipped with the right protocol needed. This removes ContextWare from the client-source path therefore increasing the scalability of the whole context-provisioning framework.

To cope with high dynamics of some context information, the CIB is managed through data distribution algorithms that, through specialised context managers, enforce optimal information dissemination between context sources and clients using a distributed set of context stores. The objective of this optimisation is to guarantee timely delivery or update of context information and minimise overhead traffic generated by sources updating context and clients requesting it. This is achieved by accounting for the rates at which those updates and requests occur when

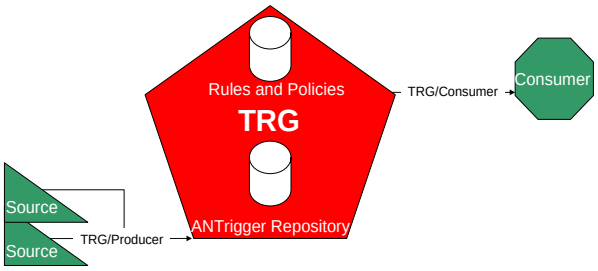

Figure 4. Schematic of ANISI TRG building block.

deciding the best place to store a particular piece of context information. Finally, to support network composition, ContextWare uses techniques for merging DHTs that compose ConCoord registries consistently.

4. ANISI: The TRG Building Block

Triggering (dubbed TRG) is the AN Functional Entity (FE), which receives "events" from other FEs, typically referred to as event sources. TRG processes the information received from the event sources based on a set of Rules and Policies, and generates "triggers". TRG is mainly concerned with mobility-related events, and any other information that can assist handover (HO) decision making. Given the stringent time constraints that such processes place, TRG is required to deliver triggers quickly, using standardised APIs based on well-defined and versatile, yet compact, data structures suitable for handover management. TRG is focussed on the processing of highly dynamic information, lower layer information, and can be considered to be a specialised instantiation of a Content Management functional entity.

TRG is an important building block of ANISI and plays a central role in assisting mobility management in the ACS, based on a "push" method: as soon as events are received, processed, and stored in TRG, the corresponding triggers are sent to their respective recipients in a single format.

The schematic shown in Figure 4 illustrates the main parts of the ANISI TRG building block: (a) the event sources, which feed TRG with relatively fast-changing information; (b) the trigger consumers, which receive notifications in the form of standardised triggers about events they are interested about; and (c) TRG itself with its associated data stores and internal logic.

4.1 TRG Producers

Event sources, also referred to as TRG Producers, must first authenticate themselves, in order to become part of the ANISI trust domain, and register with TRG before they can start sending events. Both of these procedures along with the actual sending of event information are implemented using the TRG/Producer interface. When a producer observes an event worth reporting, it will send it towards TRG using this interface as well. Conventional event sources include, for example, the radio interface (reporting events associated with radio access characteristics, such as, current or average network capacity load, SNR, dropped frames ratio, RSSI, and so on), and the battery state of charge (for mobile devices). Other sources can provide information on CPU load and storage quotas. Notable events occurring 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 (for example, MIP [8] and HIP [9][10]) state transitions, can all be reported by designated producers. The nature of the information from TRG producers is dynamic and should be "pushed" to recipients immediately. There are several scenarios (see Section 2.2) where information collected from these and other sources, often correlated, may be particularly helpful in the handover decision-making process.

4.2 TRG Consumers

Trigger consumers include firstly the handover decision-making process, but also user applications, mobility protocols, and FEs interested in optimising their performance in a mobile, multi-access network environment. Consumers can locate sources that provide useful information for their operation using ConCoord, described in Section 3.1. They have to contact TRG and authenticate, and then proceed to declare their interest in a particular set of triggers. The term *trigger* refers to a notification sent by TRG to a particular consumer based on the latter's specified preferences and filtering rules. For example, a mobility protocol can take advantage of triggers that inform it about not only the activation of a given link ("link up"), but also about crossing a threshold in the battery state of charge or the received signal strength indication (RSSI), or any combination of two or more events. For example, in a wireless sensor network gateway nodes may decide to kick start the process to elect a different set of gateway nodes if their traffic load is too high *and* their battery state of charge is too low. Triggers are sent in a standardised format, known as ANTrigger, to consumers. This is done using the TRG/Consumer interface, which also allows consumers to specify certain filtering rules to be applied before ANTriggers are sent to them.

4.3 TRG Internals

TRG implements the ANTrigger Repository and the Rules and Policies store, as well as the logic required to manage them. When different consumers register with TRG, they can set filtering rules for the events they are interested in. For example, a consumer may specify that it wants to receive triggers about the battery state of charge (type of trigger) only if it goes below 30% (filtering rule). Consumer-submitted filtering rules are stored in the ANTrigger Repository, and are used jointly with the general system policies (Rules and Policies), to classify incoming events, before delivering them to the registered consumers. These filtering rules are settled according to the specific needs of each consumer. Rules and Policies are an important component of ANISI and

apply system-wide taking precedence over the filtering rules specified by any particular consumer. For instance, security aspects, such as, which producers and consumers can be considered by TRG 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 a competing operator).

When events are posted to TRG by different producers, they are (a) classified according to the set of Rules and Policies described above; (b) transformed to the aforementioned standardised ANTrigger format, if so is required; (c) stored in the corresponding repository; and (d) delivered to the consumers that have already registered for the corresponding ANTrigger if their filtering rules apply. Note that TRG does not act as an intermediate agent in the negotiation between the producer and consumer, and remains agnostic to the end-to-end semantics of the ANTrigger values. This implies that although TRG will expeditiously push ANTriggers to consumers, it cannot itself guarantee their generation; this is the responsibility of the event sources. Guarantees about the generation of event information are given by the ANISI source authentication procedure, which does not allow malicious producers to enter the ANISI trust domain.

5. Discussion

Sections 3 and 4 described the main functional components that form the ANISI. ContextWare provides the framework within which functional entities can locate, authenticate and retrieve the information they require to support decisions. TRG uses this functionality to provide high speed event processing and notification services to other functions within the network. For example, when a trigger producer registers with TRG, TRG can subsequently contact ConCoord and register a unique UCI on behalf of the producer. This way, consumers can locate event sources using the standard ANISI mechanisms.

In addition, information in the information database can be used by TRG consumers to decide whether they wish to receive certain types of ANTriggers, *if* the networking context dictates so. Take for example an email application: when Bob starts it, the email client can query ANISI and receive user preferences/policy and temporal/spatial information in addition to networking context information. Thus, the email client can correlate context information such as “business day”, “at the office”, “connected with office 1 Gb/s LAN” and decide to skip registering with TRG altogether because neither network access cost nor performance/security is an issue. However, if the email client is started in a different networking context, our ANISI-compliant IMAP application will opt for registering with TRG.

The synergies from this choice in ANISI are several, and reflect two fundamental architectural principles: “do no harm” and “assist whenever possible”. In non-mobility scenarios, ANISI consumers can take advantage of the context information maintained by ContextWare. No performance penalties are introduced for consumers as the closed loop between TRG and producers is isolated. In mobility scenarios, consumers can use ANISI and their own policies and user preferences to determine if they need to receive the mobility-related information provided by TRG, and if so, register with it providing the appropriate filtering rules.

6. Related Work

As presented in the previous sections, information services are useful for providing additional context on which to base decisions such as whether to handover; and to which network. It can also provide valuable information to support application configuration and media adaptation to ensure the best possible application performance in the current network environment. The obvious benefits enabled by more sophisticated information services have lead to a number of work items in this area; 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 (AP) are in the local neighbourhood [11], what mobility domain a particular AP is a member of [12], and what roaming agreements are in place between one WLAN access network and a number of service providers [13]. 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.

The discovery of roaming agreement information preauthentication has been recognised as a key aspect to supporting seamless user roaming between different networks based on different, and a number of solutions have emerged including those developed by IEEE 802.21 and IEEE 802.11, and also within the IETF [14].

IEEE 802.21 [3] has also extended the information service to include other information to support handover decisions, which includes trigger events and commands, and the delivery of this information across a network. IEEE 802.21 defines three services:

- The event service is similar to TRG; this service is responsible for delivering events such as loss of connectivity to interested functions both in the network and in the user device.
- The command service allows configuration and handover initiation commands to be exchanged between network entities and user devices.
- The information service provides the information model and information repository to support handover decisions. This is accessed via the current point of attachment to the network, and has parallels with the ContextWare building block.

IEEE 802.21 has therefore 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 interface related aspects similar to the Ambient Resource Interface [4], although the scope of IEEE 802.21 does not include the higher layer aspects considered by the Ambient Service Interface definition [4]. Therefore, the current information sets defined by the base specification are static and not as diverse as though that can be handled by the ANISI.

In addition, existing solutions have to rely on pre-established relationships between networks in order to allow retrieval and distribution of information cross administrative boundaries. The support for dynamic roaming agreement establishment provided by Ambient Networks enables information to be shared in a much more flexible way, supporting a much richer set of context information on which decisions can be based. Although IEEE 802.21 and other activities are establishing information services paradigms, the focus of that work is on tightly scoped use cases where particular pieces of information are useful. Currently, these solutions merely address network attachment and handover scenarios, and even then in most cases the available information is fairly limited.

ANISI provides a way to not only support the network attachment and handover decisions with a diverse set of information about network characteristics and trigger events, but can also be applied to other scenarios where the information can be used to configure applications and adapt media delivery based on the current network characteristics.

7. Conclusions

We introduced ANISI, an information service infrastructure designed to provide services and applications at different protocol stack 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, ANISI 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.

Communication environments are becoming increasingly more complex due to the diversity of available network technologies and the proliferation of multi-function devices. 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. 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 a diversity and scalability issues. They need, therefore, to be capable of providing rapid data delivery for those applications where timely delivery is crucial. At the same time, these information services need to be reliable and robust to accommodate requests from less time-constrained applications, flexible enough to account for diversity of information served and rely on a distributed structure to address scalability concerns. The current work on ANISI is positioned within this wider research background and attempts to make a contribution towards the achievements of these goals.

Acknowledgements

This work has been carried out in the framework of the Ambient Networks project (IST 507134), which is partially funded by the Commission of the European Union. 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] T. Buchholz and C. Linnhoff-Popien, Towards realising global scalability in context-aware systems, *Proc. International Workshop on Location- and Context-Awareness (LoCA)*, LNCS 3479, Oberpfaffenhofen, Germany, 2005, 26-39.

[2] M. Mouly and M.-B. Pautet, *The GSM system for mobile communications* (Palaiseau, Cell & Sys, 1992).

[3] P802.21/D10.00, Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services, September 2006.

[4] N. Niebert, *et al.*, Ambient Networks: A framework for future wireless internetworking, *Proc. 61st IEEE VTC-Spring*, Vol. 5, Stockholm, Sweden, 2005, 2969 - 2973.

[5] R. Giaffreda, *et al.*, An authorization and privacy framework for context-aware networks, *Proc. of MATA05 Int'l W'shop*, Montreal, Canada, 2005.

[6] T. Berners-Lee, R. Fielding, and L. Masinter, Uniform Resource Identifier (URI): Generic Syntax (IETF, RFC 3986, 2005.

[7] D. Ratajczak and J. Hellerstein, Deconstructing DHTs, *IRB-TR-03-042, Intel Research*, Berkley, CA, Nov. 2003.

[8] C. E. Perkins, "Mobile IP", IEEE Communications Magazine, vol.40, no.5, pp.66-82, May 2002.

[9] R. Moskowitz and P. Nikander, Host Identity Protocol (HIP) Architecture (IETF, RFC 4423, 2006).

[10] T.R. Henderson, Host mobility for IP networks: a comparison, *IEEE Network*, 17(5), 18-26.

[11] IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification; Amend.9: Radio Res. Meas. (IEEE 802.11k), D4.0 March 2006.

[12] IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification; Amend.2: Fast BSS Transition (IEEE 802.11r), D2.1 May 2006.

[13] E. Hepworth, "Network Selection Problem Statement", 11-06-0542r1, April 2006.

[14] F. Adrangi et al., Identity Selection Hints for the Extensible Authentication Protocol (EAP) (IETF, RFC 4284, 2006).