

Self-Management of Context-Aware Overlay Ambient Networks

Bertrand Mathieu, Meng Song

France Telecom R&D

2 Av. Pierre Marzin, Lannion, France

{bertrand2.mathieu; meng.song}@orange-ft.com

Alex Galis, Lawrence Cheng, Kerry Jean, Roel

Ocampo

University College London, Electrical Engineering Dept.

Torrington Place, London, WC1E 7JE, United Kingdom

{a.galis, l.cheng, k.jean, r.ocampo}@ee.ucl.ac.uk

Marcus Brunner, Martin Stiemerling, Marco Cassini

Network Laboratories,

NEC Europe Ltd

Heidelberg, Germany

{brunner, stiemerling, cassini}@netlab.nec.de

Abstract

Ambient Networks (ANs) are dynamically changing and heterogeneous as they consist of potentially large numbers of independent, heterogeneous mobile nodes, with spontaneous topologies that can logically interact with each other to share a common control space, known as the Ambient Control Space. ANs are also flexible i.e. they can compose and decompose dynamically and automatically, for supporting the deployment of cross-domain (new) services. Thus, the AN architecture must be sophisticatedly designed to support such high level of dynamicity, heterogeneity and flexibility. We advocate the use of service specific overlay networks in ANs, that are created on-demand according to specific service requirements, to deliver, and to automatically adapt services to the dynamically changing user and network context. This paper presents a self-management approach to create, configure, adapt, contextualise, and finally teardown service specific overlay networks.

Keywords: overlay network, self-management, context-aware, ambient network

I. INTRODUCTION

The concept of Ambient Networks (ANs) introduces a new kind of mobile networks, formed by heterogeneous user/network devices and using different networking technologies. ANs can range from small size ad-hoc networks to much larger wireless networks that cover various administrative boundaries or domains. The major novelty of ANs is that they can *compose* and *decompose* with other ANs dynamically. (De)composition of ANs therefore extends a network's capability, to provide different types of (new) services to end users, by composing different (service) modules depending on the user context.

We advocate the use of overlay networks to enable the composition of services and to support the subsequent delivery of services to end-users in ANs. In this paper, we present the concept of Service-aware Adaptive Transport Overlays (SATO) for ANs, which are created for delivering a given

requested service. Given that ANs are very dynamic, heterogeneous, and flexible, a self-management approach is a natural way to manage SATOs. The SATO self-management system collects distributed user and network context in an AN, and automatically assigns dedicated nodes to analyse these information, in order to support the setting up, and subsequently the adaptation of SATOs.

This paper briefly introduces the main concepts of ANs, before focusing on SATO, its lifecycle and explaining its self-management. The following section sums up some evaluation results and section 5 concludes this paper.

II. SATO IN AMBIENT NETWORKS

A. Description of Ambient Networks

The Ambient Networks (AN) project [1] seeks to study ambient networks taking into consideration aspects like multi-radio interfaces, mobility management, security issues, composition of ANs, context management and service delivery [2]. The main task is to design an overall architecture enabling the user-centred delivery of service, any time, everywhere, whatever the device and the network are. The entity that gathers all the information and links them is called the Ambient Control Space (ACS) (Figure 1: The ACS, OM FE and SATO. It can be seen as a control framework that manages all characteristics of ANs, provides abstraction of the resources and enables the service delivery for ANs.

A service interface, the Ambient Service Interface (ASI) has been defined as an "upper layer" interface of the ACS that is accessible to applications to define their requirements and specify how the service should be delivered (in terms of QoS, security, connectivity). The management of this request is performed by the *Overlay Management (OM) FE* [4][5]. The OM FE will then create and maintain a service-specific overlay network to fulfil the service provider requirements and to manage the service delivery to end-users while adapting to user and network context. This specific overlay is called a

Service-aware Adaptive Transport Overlay (SATO) network. The *Service Context (SC) FE* [3] is also very important in this work since the SATO should adapt automatically to context.

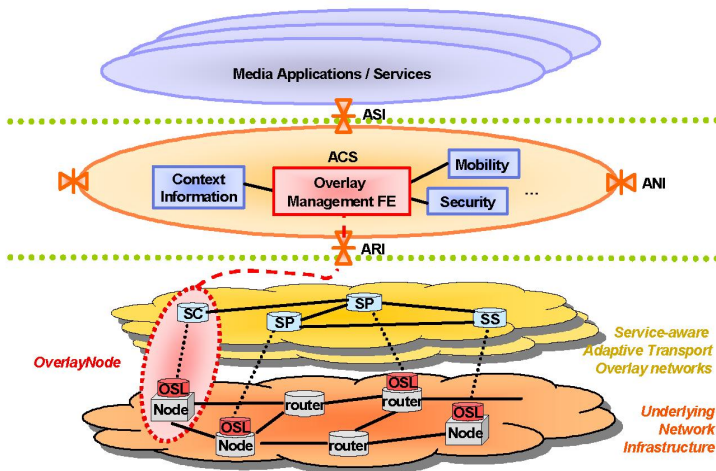


Figure 1: The ACS, OM FE and SATO

B. SATO Overlay Network

SATOs are service specific, as they take into account specific requirements, specific resources (and other supportive services) in the network that are needed to support/operate/manage the delivery of a particular service to end-users. The services are assumed to be designed in a modular and fine-grained way. This leads to elementary services modules that are distributed in the overlay network and composed by OM FE for providing the overall service. The composition is performed by configuring the overlay routing tables in such a way that the data passes through the modules in the right order. Those services modules are called SatoPorts (SPs) and can then be viewed as the necessary resources or supportive services or service computation resources that are needed for deploying a (new) end user service (Figure 1: The ACS, OM FE and SATO [6]. Examples of SPs are application processing modules such as transcoders, cache, virus scanning, security services or application control data such as a P2P SIP (Peer-to-Peer Session Initiation Protocol) network [7] but can also be networking service such as QoS-control services. SPs are hosted on nodes that should be included in the SATO to ensure the good delivery of the service. Those nodes are called SATO Overlay Nodes (SON).

The OM FE is a distributed entity. Each SATO node hosts a part of it, the SATO manager. The SATO manager is in charge of the management of the SATO (setup, complex adaptation, teardown) but specific tasks are delegated to other modules. For instance, the SATO routing algorithm responds to simple context changes which can be resolved by simple data re-routing within the same SATO overlay network topology. Finally, the SON Local Resource Manager and context sensors (bandwidth sensors, Flow sensors...) are designed in the SON to monitor and collect specific resources and to provide information related to the current context to the OM FE.

III. SELF-MANAGED SATOS

This section presents how a SATO is self-managed, how context information is managed and the solution for the three steps of the lifecycle of the SATO, i.e. the establishment, the adaptation and the teardown.

A. Characteristics of ANs to take into account

The setup and maintenance of simple overlay networks is typically not seen as a big problem in fixed networks. However, in wireless and dynamic networks, topologies and network characteristics dynamically change all the time, making the setup and maintenance a tricky problem. Furthermore for Ambient Network, where ANs can dynamically (de)compose. The SATO overlay networks are tailored towards the specific requirements of a service delivery, and with it, the topology of the overlay network is also dependent on the service running within the SATO overlay network.

B. A DHT-based approach for managing Context Co-ordinators

Bearing in mind all those characteristics, it clearly appears that to setup SATOs, there is a need for a monitoring system to be aware of the current context (by collecting and processing related real-time network information, known as aggregation in this paper) and a management system to use the processed information to setup (and subsequently to manage) SATOs.

The challenge of enabling aggregation in ANs is that we need to dynamically select a (or a set of) node(s) to become responsible for collecting and subsequently analysing the collected information. In other words, we need some "co-ordinators" in the network, each responsible for managing (i.e. aggregating) a type of distributed data, and serves clients upon requests. Since the broadcasting of information to AN nodes is not scalable and since static assignment of Co-ordinators is not reliable in wireless and changing environments, we suggest the solution of the dynamic election of "co-ordinators" by utilising the dynamic and scalability advantage of DHTs.

C. Context Management : How to aggregate context information in Context Coordinators?

We define the different types of aggregated information as *context*, and assign each with a Universal Context Identifier (UCI) [8]. A UCI is a String identifier similar to a Uniform Resource Identifier (URI) and uniquely identifies a type of context. We assume that each AN node has a set of sensors (not necessary a complete set of sensors) that are capable of monitoring different types of context.

- Start of the Overlay Sensors

During bootstrapping of an AN, the distributed sensors in the network will start collecting the type of context that they are responsible for (step 0 in Figure 2). The collected context will be stored locally. The sensors register themselves with the local Context Co-ordinator (ConCoord) (step 1 in Figure 2). The sensor registration process involves registering the UCI of the type of context that the sensor is responsible for, and optionally a network location identifier that the sensor wishes

to be identified by other nodes in subsequent communications (i.e. IP).

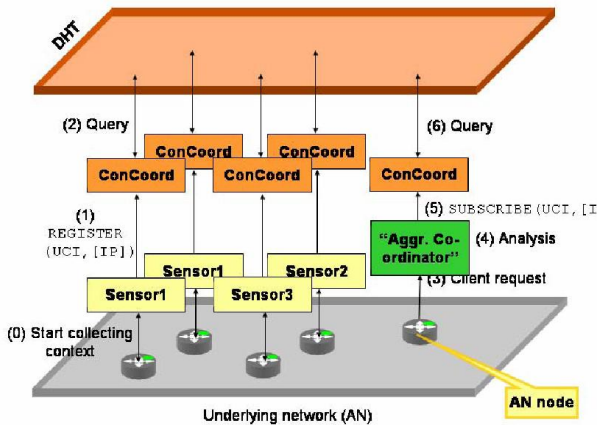


Figure 2 A SATO establishment process

- Dynamic Aggregation Node Determination through ConCoord

The ConCoord then begins a distributed query process (step 2 in Figure 2) to identify which nodes in the AN to be responsible for aggregating, and subsequently disseminating the type of (aggregated) context that the sensor has just registered with (identified through the UCI provided by the local sensor). A matching of the UCI and the DHT's keyspace enables AN nodes to know if they are responsible for aggregating the type of context information represented by the UCI.

For reliability, it is worth mentioning that the aggregation responsibility is automatically transferred to another node in the AN, should the original responsible node fails. Thanks to DHT principles, automated aggregation responsibility recovery is achieved, *without* any form of real-time re-negotiation between the remaining nodes.

D. Context Resolution

When receiving a setup request (step 3 in Figure 2), the OM FE first analyses the request to know the application requirements (e.g. required QoS or security policies) and the needed intermediate services modules... (step 4 in Figure 2). The result of this analysis is a list of context criteria and a service chain, composed of the required SPs to deliver the service. The Aggregation Co-ordinator (a.k.a. the SATO Manager) will then query its local ConCoord through a subscription process (step 5 in Figure 2). The SATO manager subscribes to the ConCoord with the UCI(s) of the type(s) of aggregated context that it needs to establish a SATO. The ConCoord then begins a distributed query process (step 6 in Figure 2) to identify which nodes in the AN are responsible for aggregating the type of context that the SATO Manager has just subscribed to. The ConCoord on the same node as the SATO manager will retrieve the aggregated context from the aggregation node. The SATO Manager will then trigger a distributed aggregation process to establish a corresponding SATO (see later).

E. SATO Establishment

Once the SATO Manager has received the aggregated context that it has previously subscribed for, it can begin to establish a SATO by communicating with the nodes on which the located SPs are hosted (i.e. negotiation on overlay IDs, security parameters... etc.). Figure 3 shows different establishment of SATOs in an AN.

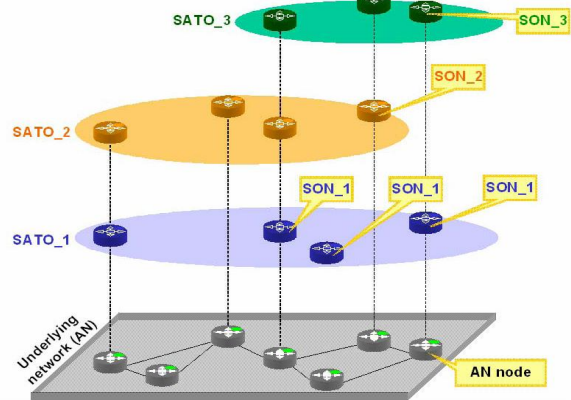


Figure 3 Different SATOs in an AN

F. SATO Adaptation

The SATO must be self-adaptive, which means that when changes in network context have taken place, the SATO should be capable of carrying out dynamic re-configuration. The change of context can be detected by the components of the SATO, like the QoS sensor, the flow sensor or triggered by external entities of the ACS, like the context or the mobility Functional Entities. Changes in the context may just imply a re-calculation of the route between SPs, done by the SATO QoS routing algorithm when the QoS is no longer ensured on the current path (overloaded link, failure of nodes...). More important context changes (user's mobility, user's change of device or when Ambient Networks compose) can lead to the need to add, remove SPs in the overlay in case the current available SPs do not offer such functions. In this case, the adaptation process is similar to the establishment process including the discovery of the needed SP.

G. SATO Teardown

When it is no longer needed, the SATO should be torn down, in order to release unused resources in the overlay nodes. Three different cases can be considered: 1) when SATO server endpoints cause the SATO to be destroyed; 2) when the duration for which the SATO was set up is over (if not extended); 3) When endpoints did not leave properly or failed

IV. EVALUATION OF THE SOLUTION

In this section, an evaluation of the efficiency for looking up context aggregators and of the reliability and scalability of the solution is presented. Using the CAN protocol [9] for the DHT, we have developed a CAN-DHT simulator that creates 2-dimensional CAN-DHTs of different network sizes, and generates a graphical display of the DHT keyspace ownership. We have also developed a "SATO Manager" management

simulator, which takes different values of the neighbourhood scale N (see later), then simulates the activation of SATO Manager(s) on different number of overlay nodes in the DHT accordingly, and returns the total number of a particular type of SATO Manager that are currently running.

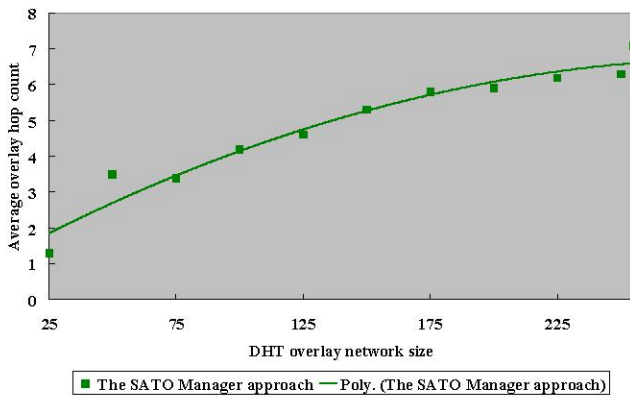


Figure 4 Average overlay hop count Vs. DHT overlay network size

The results in Figure 4 suggests that the SATO Manager approach is *efficient*: only few overlay hops are needed for a ConCoord to reach an aggregation node of interest. Note that the slope of the curve in Figure 4 gradually decreases as the DHT overlay network size increases. This suggests that the approach is *scalable*.

The SATO Manager approach should be robust to handle heterogeneity and dynamicity in ANs: that means that in the case of a failing node, the node is “backup” by its immediate overlay neighbours. The evaluation is carried out over a network of 100 overlay nodes. We assume that an administrator would like to have 10% of the nodes in the network to be the hosts of a particular type of SATO Manager. The failure rate ranges from 0% to 80%.

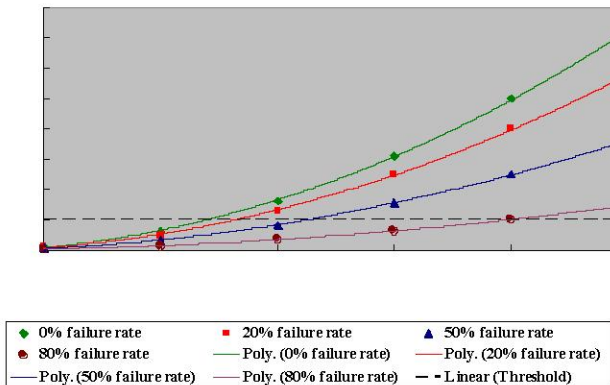


Figure 5 Comparison between robustness

The result of Figure 5 proves that the total number of active SATO Managers (of the same type) that are available at one time in the network can be made constant under different failure rates. For example, when the network is extremely heterogeneous and mobile e.g. with 80% failure rate, we can adjust $N=4$ to achieve a desired level of robustness i.e. to achieve a desired number of capable nodes in the network (i.e. to have 10 capable nodes in the network at one time, as defined by the administrator). Therefore, the provisioning of N

(or M) in SATO Manager approach provides the necessary facility to adjust the level of robustness.

V. CONCLUSION

The self-management of context-aware SATO overlay networks, presented in this paper, enables the setup, maintenance and teardown of SATOs in a completely dynamic and context-aware manner. To manage the context, three concepts have been presented: the sensors that are in charge of collecting information, the Context Coordinators, who are in charge of storing and aggregating the context and a scalable and decentralised approach that dynamically determine which node in the network is an aggregation node for a type of context. This approach relies on the DHT principles, for dealing with large networks and reliability and load balancing has been evaluated. The setup of SATOs, retrieving the context from the context coordinators, the adaptation, via the overlay routing protocol or via the OM FE and the teardown, triggered or automated, have been presented.

VI. ACKNOWLEDGMENT

This paper describes work undertaken in the context of the Ambient Networks (Phase 2) - Information Society Technologies project, which is partially funded by the Commission of the European Union.

VII. REFERENCES

- [1] EU-IST project 507134 Ambient Networks, <http://www.ambient-networks.org>
- [2] N. Niebert, et al, “Ambient networks: An architecture for communication networks beyond 3G,” IEEE Wireless Communications, vol. 11, pp. 14-22, IEEE, April 2004.
- [3] R. Ocampo, L. Cheng, Z. Lai, A. Galis, “ContextWare Support for Network and Service Composition and Self-adaptation”, in Proceedings of the 2nd International Workshop on Mobility Aware Technologies and Applications (MATA) 2005, Montreal, Canada, October 2005.
- [4] L. Cheng, K. Jean, R. Ocampo, A. Galis, “Service-aware Overlay Adaptation in Ambient Networks”, International Multi-Conference on Computing in the Global Information Technology (ICCGI) 2006, Bucharest, Romania, August 1-3, 2006.
- [5] T. Petersen, et al., “SMART – Final Architectural Design”, IST-2002-507134-AN/WP5/D03, http://www.ambient-networks.org/publications/D5_3_SMART_Final_Architectural_Design_PU.pdf
- [6] B. Mathieu, M. Kleis, M. Song, “A P2P Approach for the Selection of Media Processing Modules for Service Specific Overlay Networks”, International Conference on Internet and Web Applications and Services, Workshop on P2P Systems and Applications (ICIW P2PSA 2006), February 23-25, 2006.
- [7] David A. Bryan, Bruce B. Lowekamp, and Cullen Jennings, “A P2P Approach to SIP Registration”, draft-bryan-sipping-p2p-02.txt, March 2006
- [8] K. Jean, L. Cheng, R. Ocampo and A. Galis, “Contextualisation of Management Overlays in Ambient Networks”, International Multi-Conference on Computing in the Global Information Technology (ICCGI) 2006, Bucharest, Romania, August 1-3, 2006.
- [9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A Scalable and Content-Addressable Network”, In Proc. ACM SIGCOMM, San Diego, CA, USA, Aug. 2001