Self-Configuration for Composable Networks

Simon Schuetz[1], Marcus Brunner[1], Zoltán Lajos Kis[2], Csaba Simon[2], Robert Szabo[2], Gergely Molnar[3]
[1]Network Laboratories, NEC Europe Ltd, Germany, email: {brunner, schuetz}@netlab.nec.de
[2]Budapest University of Technology and Economics, Hungary, email: {kiszl, simon, szabo}@tmit.bme.hu
[3]Ericsson Hungary Ltd., Hungary, email: gergely.molnar@ericsson.com

Abstract— **Self-configuration of networks is key to reduce operational cost and enhance the usability for inexperienced users. Specifically, this holds for new network architectures where whole networks compose, decompose, move in a dynamic way. Various network technologies already perform self-configuration or plug and play configuration. For example, Ethernet is self-configuring. More types of networks are based on the Internet Protocol (IP) also in mobile networking. Therefore IP routers or IP-like routers, and new packet-based network architectures must be plug and play as well. In this paper we propose an architecture and protocols for self-configuration of Ambient Network Elements such as routers, end-systems, and the control plane running on them and we describe the protocol and first prototypical implementations.**

Index Terms— **plug and play, self configuration, mobile networking, communication networks, network management**

I. Introduction

The main purpose of self-management technologies is the reduction of the cost of network deployment and operation. Providing self-management technologies increases the number of network elements managed per person or decreases the time needed to operate a network. Additionally, self-management increases the usability and enables inexperienced users to run networks with little networking skills and knowledge.

Additionally, there are environments where manual configuration is impractical, namely when the change of configuration is more frequent compared to long living network installations. Such types of networks are the focus of studies in the Ambient Network (AN) project [1]. The base assumption for ambient networks is that whole networks compose and decompose dynamically. For example, personal area networks might compose with a train network when the user is traveling. In such a scenario there are three somewhat different steps involved. First, single network components are put together to form an Ambient Network. This means single AN elements (including hosts and routers since no separation between hosts and routers is foreseen) are connected wired or wireless and need to get the base connectivity up and running. Second, the AN control plane components need to get configured and bootstrapped. Given these two steps, an AN domain is formed and operational. The third step is then concerned with composing AN domains, where two or more networks compose into a larger Ambient Network. The third type is not a focus of this paper.

Note that the self-configuration of AN elements is on a time scale different than some work in mobile ad-hoc networks, which is also part of the Ambient Network project. There topology changes, node additions and removals happen fast, and ongoing transport sessions need to seamlessly handle the change. We target a bit longer-lived installations, which work for some time in the same or similar topology. Still some of the mobility architectures such as mobile IP require a fast configuration of a care-of-address.

The Ambient Network project is mainly focusing on control plane aspects of the problem. However, it assumes a packet-based data plane. For ease of understanding we can assume throughout the paper, that an Internet Protocol (IP) based data plane is used.

II. Related Work

Related to this work a number of technologies are already available. We classify them into three areas, namely host-focused, router-focused and service-focused. Since the boundary between them is quite difficult to draw, we list here some technologies in all three areas. Concerning address auto-configuration in mobile ad-hoc networks, which is not the focus of this paper, but still is related, [11] gives an overview of several approaches.

Host-focused technologies concentrate on configuring end-systems only. No network equipment such as routers is configured. Version 6 of the IP protocol [3] and Dynamic Host Configuration Protocol (DHCP) [5] include several mechanisms to achieve host configuration. Also the IETF Zero Configuration Networking WG defines a zero configuration protocol [6] for LANs.

Router-focused mechanisms include Automatic Prefix Delegation [8] and some new work extending OSPF [9] for router auto-configuration. UIAP [10] is a protocol that allows an application to validate and defend the uniqueness of an identifier presented by an application within a scope, called domain, which is one of the base technologies for duplicate identifier, address or locator detection.

III. Overview of Ambient Networking

Since its beginning, the Internet's development has been founded on a basic architectural premise: a simple network service is used as a universal means to interconnect intelligent end-systems. The end-to-end argument has served to maintain this simplicity by pushing complexity into the endpoints, allowing the Internet to reach an impressive scale in terms of inter-connected devices. However, while the scale has not yet reached its limits, the growth in functionality - the ability of the Internet to adapt to new functional requirements - has slowed with time. In addition, the ever-increasing demands of mobile applications and network services currently face the relative inflexibility of IP infrastructures. In this sense, the pervasive use of the Internet in mobile networks to support such applications has revealed important deficiencies. One such structural deficiency is the lack of ambient control [1], [2].

The notion of the Ambient Control Space (ACS) is introduced to encompass all control functions in a certain network domain (see Figure 1). The ACS together with an IP-based connectivity network is called an Ambient Network (AN). The ACS hosts a set of control functions. In addition to the basic functions required for management, security, and connectivity, the ACS hosts additional functions, such as control functions for supporting mobility, connectivity, multi-radio access, smart media routing, security, QoS and management, as well as more abstract functions like the provisioning of context information. Each of these control functions is accommodated in different aspects of the AN project.

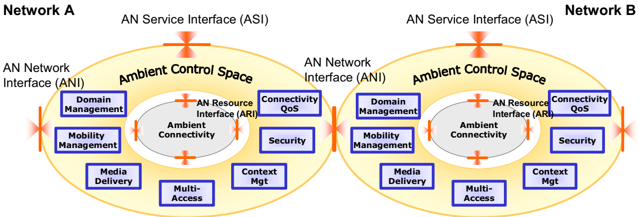

Figure 1: Ambient Control Space (ACS)

In this AN environment, we focus in the following on configuration management in a plug and play way. This allows for affordable network management, integration of network management functionality into the control plane, and fully distributed management functionality. Additionally, the dynamic composition calls for self-configuration of components and for supporting the composition of networks.

IV. Self-Configuring Ambient Networks

In Ambient Networks (AN), AN elements are likely to enter or leave specific networks at any time. Each time their configuration has to be adapted. Consequently, manual configuration is merely impossible and plug and play (PnP) mechanisms should automatically configure the AN elements. In contrast to nowadays networks, this self-configuration is not limited to (mobile) end-hosts, but also includes router elements, as an AN element may also act as such, or any other system in a mobile network including base stations. Therefore, novel PnP mechanisms are required to not only configure end-hosts but also to completely integrate router and base station elements into an Ambient Network.

Figure 1 shows a very generic view on the Ambient Network Domain. So the AN domain consists of a set of AN elements. The AN runs an ACS, where each network node has a set of functions running for different aspects of the control plane.

Given the AN domain is already configured, a newly attached node must get configured automatically when attaching to that network (see Figure 2). This means getting layer 2 up and running, assign IP addresses to all interfaces of the connecting node. Naturally, these addresses must be unique at least within the AN domain. When the domain runs off-the-shelve IP, the addresses must be unique also globally.

In a second step, ACS functions on the new node have to be bootstrapped. For example, it must start up the same routing daemon as the rest of the network runs. Additionally, all the other functions of the ACS might need some configuration. The start up configurations for the control functions need to be sent to the node, and the right instances of the control functions need to be started.

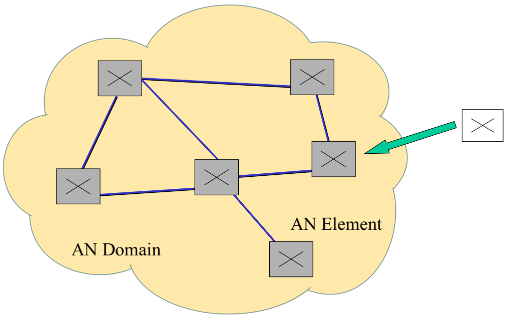

Figure 2: Self-Configuration of a Newly Attached AN Element

The more complex scenario is when two full AN domains compose. In this scenario, different outcomes are possible. The networks can either fully compose into a single one or they can remain completely autonomous and only run a gateway between them. In the former case, a new so-called Super Peer has to be elected to take over the management of the newly composed AN domain. See later for a detailed description of the concept of super peers. These two possibilities are the extreme cases, several possibilities in between are possible, for example, they coordinate addressing and routing, but keep QoS management separated etc.

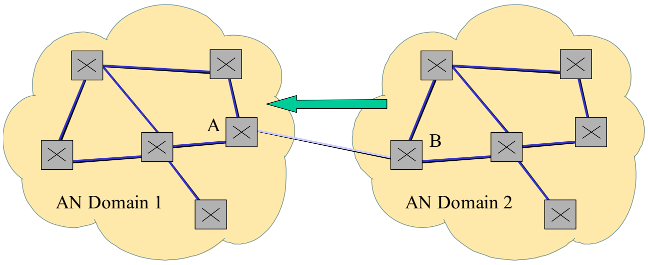

Figure 3: Composition of two ANs

Also in this case, a base connectivity needs to be self-configured – including possible address conflict resolution – when physical connectivity is established between AN Element A and B in Figure 3. Then a negotiation between the domains needs to take place to decide on what type and degree of composition. In order to practically perform an automatic negotiation, A and B need to notify their own domain's top-level Super Peer, who can jointly decide on the type of composition. In the following, we describe the various plug and play components in more detail.

V. Plug-and-Play Components

The PnP management requires a basic set of components that each manages a part of an AN elements configuration. This section describes these components.

A. *Base Station/L2 configuration*

The Base Station/L2 configuration component is responsible for providing operable link layer configurations that serve as communication channels between AN elements. Without any link layer connection, no communication is possible at all. For example, in case of a WLAN access points, this task involves configuration of a proper ESSID, selection of wireless channel, adjustment of signal power, etc., in case of standard Ethernet no further action is required.

In order to avoid bottlenecks and single point of failures, the base station/L2 configuration component uses a fully distributed approach. Base stations use their wireless interfaces to scan the environment. As a result, they obtain a list of neighbored base stations, which they can contact to retrieve or update information about the network. Collected information is entered into a local database and updated when network state changes. By propagating this information from neighbor to neighbor, state information flows through the entire network of a single AN domain. Thus, each base station can infer an optimal configuration from its local database without the need for contacting some centralized services.

B. *IP client/Locator configuration*

PnP configures interfaces associated with available link layer connections to provide basic network connectivity. It enables AN elements to act as end-hosts. They are then able to contact or being contacted by other elements of their current AN domain. Note that AN locator is denoting the location of an interface within a network, and it is distinct from the identifier of the node or interface. In IP networks normally both properties are used together in the IP address. (Note, that it is foreseen that the next generation packet networks are based on an identifier/locator split).

In IP-based networks, an appropriate IP address has to be retrieved for the interfaces. PnP management handles detection of existing AN domains, which may predetermine IP/Locator settings. In this case, mechanism like or similar to DHCP [7] or IPv6 auto-configuration [4] should be used to configure the interfaces.

The IP client/Locator configuration is also responsible for duplicate address detection (DAD) in case of network compositions. Comparing complex sets of addresses during every composition would result in a non-scalable solution. Therefore in the ideal case, these address sets should be continuous (e.g. each AN domain has a subnet). This way the DAD is reduced to comparing these subnets only. Of course in complex scenarios with many hierarchy levels, the comparison of these low-level subnets would result in the same complexity. For a scalable solution, we introduced subnet aggregation in some higher level Super Peers. For example in Figure 4, Super Peers A and B propagate their address sets to Super Peer E, while C and D do the same to Super Peer F. Both E and F aggregate the received sets, and propagate the aggregated sets only to Super H. The same way super Peer I will receive only three aggregated sets of addresses. This way the top-level Super Peers can compare aggregated sets, and resolve the conflicts by spreading the changes on their hierarchy tree.

During the first phase of the actual composition process, we resolve local conflicts among the peers, who are gatewaying between the AN domains. In the second phase all composing domains' top-level Super Peers resolve any present address conflicts among themselves. After these two phases, it is ensured that the top-level Super Peers are able to communicate over the network layer. In the final third phase, the top-level Super Peers compare their aggregated subnets, and negotiate new subnets in case of address conflicts. After the negotiation is finished, changes are spread downwards on the hierarchy tree of each AN.

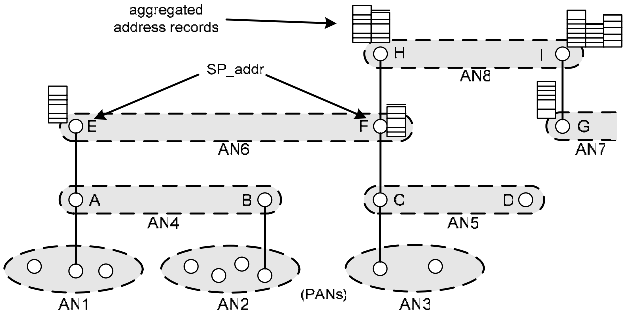

Figure 4 - DAD using Super Peer Hierarchy

C. *Router configuration*

PnP management configures all available interfaces of the AN element. It has to prepare and manage address spaces for these interfaces. To perform this task AN elements use basic connectivity to existing AN domains, which is provided by the IP client/Locator configuration.

Router configuration uses a management hierarchy, which builds a spanning tree overlay on network topology. The root of this tree is a master router, which is responsible for the complete address space of the corresponding AN. To prevent bottlenecks and single points of failure, higher level routers delegate management tasks to their subordinates on a lower level. Subordinated routers try to manage their children with minimal interactions to their parents. Only in case that a router is unable to manage the subordinated part itself, it contacts a parent for assistance. An example could be that the assigned address space gets exhausted due to a large amount of clients. In this case, a subordinated router might request a new or additional address space from its parent router.

If a new AN element enters an already configured AN domain, it scans the network for configured routers and attaches to the most fitting one as a subordinate. In case two ANs compose, the root routers have to synchronize their address space allocation.

D. *Routing configuration*

Once the AN element has configured available interfaces, it starts acting as a router. Therefore, it must configure and activate an appropriate routing protocol, e.g. OSPF. For single AN elements joining an AN domain, basic configuration parameters can be retrieved from already running neighbored routers that the AN element connected to in the previous steps and adopt their configuration. This allows now for intra-domain connectivity of nodes.

In case of network compositions, the configuration of routing protocol parameters is not as trivial as with a single joining AN element. As the two networks potentially have conflicting configurations, dedicated routers in each AN domain should have complete information about the domain structure to be able to resolve conflicts or to optimize domain-wide configuration. For robustness, the information base of dedicated routers should be replicated either on backup dedicated servers or in distributed hash tables.

E. *ACS bootstrapping*

Now the rest of the Ambient Control Space needs to be bootstrapped. Only after that the full Ambient Network is up and running both on new joining nodes or on merged

networks. This basically means, that the various functions of the ACS are brought together and they can then negotiate further configurations on the control level.
Changes in the domain – such as new elements or composing AN domains – should also trigger the Super Peer election process. The process should elect a new Super Peer or reaffirm the current Super Peer's status.

F. Maintenance

Running each of the PnP components for a new AN element in the order stated above will allow the AN element to become an integral part of AN domains. However, this set of components has not only to be executed once per new AN element. Due to the dynamics in ANs, each component must constantly or periodically adapt and optimize the AN elements' current configuration. The Base Station/L2 configuration component might have to reconfigure Base Station parameters, e.g. due to two approaching AN networks using the same wireless channel. In some cases, IP/Locator configuration component has to reassign an interface's IP/Locator, e.g. if detection of duplicate addresses occurs. The router configuration component can change address spaces, e.g. when running out of addresses due to many connecting clients. The routing configuration component has to reconfigure routing protocols parameters when topology changes for any reason. Last but not least, the information base of the Super Peers should be updated as the status of the peers change within the AN. The Super Peer of the AN might then resign if there are more suitable peers for that role. Furthermore a new Super Peer should be drafted if the earlier elected one fails (or more likely leaves the network).

G. Interaction between PnP Components

In general, the configurations of different components are not independent of each other. Consequently, the PnP components have to interact with each other to build a consistent configuration. In particular, the PnP components have to implement the following interactions:

1. A network interface that established a new link layer connection should also verify its IP/Locator configuration. If it is not configured yet, an initial configuration has to be installed. Otherwise, if it is already configured, it must be ensured that the current configuration is consistent with the network the interface connected to. In case of inconsistencies, the configuration has to be adapted. Therefore, the Base Station/L2 configuration component should trigger the IP/Locator configuration component after configuring a link layer connection.

2. After configuring or updating the configuration of an interface, the IP/Locator configuration component should notify the Router configuration component. (Re)-Configuration of an IP/Locator might force a router to adapt its address space allocations, e.g. if the newly configured IP/Locator is not in its currently managed address space. Furthermore, if the new configuration is caused by a newly available connection, this connection might provide additional information, e.g. about other AN domains. Using this information, the Router configuration component is able or possibly even forced to adapt or optimize its configuration.

3. The Routing configuration component should constantly verify and optimize its configuration. This is even more important in case of topology changes. As most router configuration changes result in topology changes, Router configuration components should propagate topology changes to the Routing configuration (or the routing protocol itself adapts). Since as a result the routing tables might be updated, the peer management database should be adjusted, as well.

4. The last PnP process triggered is the Super Peer election. This step is required to enable the network management processes functioning appropriately. The Super Peer is elected among the peers of the new AN domain, or among the current top-level Super Peers of the merging AN domains. The elections are based on the candidate peers' aggregated parameters. To avoid single point of failure and increase robustness a distributed data store is used to replicate common information among peers of the AN.

The interactions above basically define the information flow from bottom up when a new element enters an AN. However, there is also the possibility that interactions have to be performed top down within an already configured AN. Thus, additional interfaces for component interactions are required:

5. Based on the location of a newly elected Super Peer, a routing optimization process might be initialized to assure optimal routing.

6. Depending on the routing protocols used, the routing configuration component might be able to optimize its policies and parameters if some routers modify their current configuration. Providing an interface to the router configuration component, routing configuration components could suggest or demand router reconfigurations, which lead to better network performance.

7. If a router reconfigures its current settings, these changes must be propagated to its client nodes that had received configuration parameters from this router. Especially in case of address space renumbering, all client nodes have to adapt their IP/Locator configuration, accordingly. Address space renumbering could be required e.g. due to address space exhaustion, address collisions after network compositions or because it provides routing optimization options.

VI. Implementation and Results

In the following we present some preliminary implementation of the system.

A. IP/Locator configuration

We have created a platform with the capability of realizing AN's both by joining new AN elements into ANs and by the composition of AN domains. The platform also realizes the overlay hierarchy and Super Peer concepts, thus serving as a base for the implementation of the proposed IP/Locator configuration procedures. We have also studied the possible implementations of the network layer configuration.

B. *Router/Routing configuration*

We have implemented a first prototype on Linux using IPv6 because the division of IP addresses is easier in IPv6. Basically we configure the site local part of the IPv6 addresses. As routing protocol implementation we use the Zebra OSPFv3. The protocol daemon is implemented in user space and uses TCP to talk with the neighbors, except for the very first phase of neighbor discovery where we use UDP multicast on an interface.

The implementation does work in our limited testbed with six Linux routers. We can easily plug together some of the routers (without any configuration) and a running IP network is resulting from running the protocol daemons on each node.

In the current implementation, all routers are working in the same routing area (area 0, also called backbone area in OSPF). They all will have a complete database describing the network topology in the whole area. Thus, paths calculated in this phase will be optimal in terms of minimal distance, but they are not optimal concerning memory and bandwidth waste. This becomes a critical issue when the self-configured network grows and route aggregation should take place. But this task can be done in maintenance and self-optimization phase. The first task is to get the network up and running at all.

In order to perform more tests, we made the software run on a single PC in emulation mode. The daemons run as a single process communicating with each other, and the configuration is written to file compared to really configured. This allows us to perform measurements of the performance of the protocol, even so the speed we regard not as really important. We measured from the start up of the processes until the configuration files for each emulated node is written to disk. For a 6 node network the protocol converges in 3.5 seconds. For a 12 node network it converges in 6 seconds.

C. *Super-Peer election*

A prototype implementation of the Super Peer election has been created. The implementation was integrated into the platform mentioned in sub-section VI.A. The election mechanism has been validated to work fine in a demonstrator of the Ambient Network project, where it even interacts more closely with the AN composition and context-aware networking components of the ACS.

VII. Conclusion and Further Work

In this paper, we have described a proposal for the self-configuration of Ambient Networks, a prerequisite of upcoming new network architectures in the mobile wireless industry. Both from an economical point of view, namly the reduction of cost, as well as from a user point of view, ease of use, self-configuration is a requirement. We showed that self-configuration does work in small, not yet fully integrated implementations.

In the future, we want to focus on various shortcomings of the current version of the protocols. Concerning security the problem is that there is no security possible without a certain security infrastructure or still manual configuration or pre-configuration before shipping the routers. There are possibilities to authenticate a router when sold with a certificate already configured. This allows for checking the vendor of the router, but not what operator it is running in. Additionally, we will run a number of scalability and performance tests, however for that we need to simulate or emulate the systems.

VIII. Acknowledgements

This document is a by-product of the Ambient Networks Project, partially funded by the European Commission under its Sixth Framework Programme. It is provided "as is" and without any express or implied warranties, including, without limitation, the implied warranties of fitness for a particular purpose. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks Project or the European Commission.

References

[1] Niebert, N., Flinck, H. Hancock, R. Karl, H. Prehofer, C. "Ambient Networks – Research for Communication Networks Beyond 3G"- 13th IST Mobile and Wireless Communications- Summit 2004, 27-30 June 2004, Lyon, www.mobilesummit2004.org

[2] WWI-AN Ambient Networks Project WWW Server - www.ambient-networks.org

[3] Deering S, "Internet Protocol version 6 Specification", RFC 2460, December 1998

[4] Thomson S., "IPv6 Stateless Address Auto-configuration", RFC 2462, December 1998

[5] Droms R. et al., "Dynamic Host Configuration Protocol for IPv6", Internet Draft, Work in progress, draft-ietf-dhc-dhcpv6-28.txt, November 2002

[6] Linton J., "Zerouter Protocol Requirements", Internet Draft, Work in progress, draft-zerouter-requirements-00.txt, December 2002

[7] Johnson R, "Subnet allocation using DHCP", Internet Draft, Work in progress, draft-ietf-dhc-subnet-alloc-00.txt, February 2002

[8] Haberman B., "Automatic Prefix Delegation Protocol for Internet Protocol version 6", Internet Draft, Work in progress, draft-haberman-ipngwg-auto-prefix-02.txt, February 2002

[9] Chelius G., "Using OSPFv3 for router auto-configuration, Internet Draft", Work in progress, draft-chelius-router-autoconf-00.txt, June 2002

[10] White A., "Zero-Configuration Subnet Prefix Allocation Using UIAP", Internet Draft, Work in progress, draft-white-zeroconf-subnet-alloc-01.txt, October 2002.

[11] Weniger K., Zitterbart M., "Address Autoconfiguration in mobile Ad Hoc Networks: Current Approaches and Future Directions", IEEE Network, Vol 18 (4), July 2004.