 The definitive version of this paper has been published in Mobile Peer-to-Peer Computing MP2P'06, in conjunction with the 4th IEEE International Conference on Pervasive Computing and Communications (PerCom'06), 2006, 10.1109\/percomw.2006.140.

Using Kademlia for the Configuration of B3G Radio Access Nodes

Simon Oechsner*, Tobias Hoßfeld*, Kurt Tutschku*, Frank-Uwe Andersen†, Luca Caviglione‡

*Department of Distributed Systems, Institute of Computer Science, University of Würzburg, Germany
†Siemens AG, Siemensdamm 62, Berlin, Germany
‡Department of Communications, Computer and Systems Science, University of Genova, Italy

***Abstract*— Configuration and management of radio access nodes in Beyond 3G (B3G) networks such as NodeBs, Wireless LAN or WIMAX access points is a complex task that consumes a high amount of resources. Most configuration changes have to be done by an operator, especially when it comes to the adaption of a node to its environment, i.e., the cells surrounding its own. This is an expensive and inflexible method. Moreover, the required manual interference hampers the adaptivity of B3G networks towards topology changes.**

In this work, we present an architecture that allows for the detection of neighboring cells, thereby enabling an easy or even autonomic configuration of the access nodes. In addition to this, the presented Kademlia-based overlay network enables the storage of information needed for network management, such as configuration data or performance measures.

I. Introduction

Beyond 3G (B3G) mobile networks are expected to be highly flexible in technology and structure. Mobile terminals will move seamless through a variety of wireless access systems (e.g. 3G, WLAN, and WIMAX) as well as they would like to roam between operators. In addition, B3G operators have a strong interest in rapidly extending their networks if additional coverage or capacity is needed. This leads to quickly evolving B3G access networks. In order to provide flexible B3G services, however, concise network information, e.g. about handover possibilities or system authentication, is needed. Hence, B3G network control and management mechanisms have to be efficient (ie. exchange information timely and reliably) as well as adaptive towards a changing network topology.

Today's mobile network management and operation is based on the static and hierarchical architecture of these systems. Multiple network elements are being administrated by a superior element manager, which in turn offers its data to central entities (e.g., a Network Operation Center). These central configuration management systems, however, show the typical vulnerabilities of single control points for failures and overload situations.

This paper presents a P2P-based architecture for supporting configuration management in B3G access networks. The *easy Configuration of Attachments Points (eCAP)* architecture enables all participating APs to identify their neighbors as well as providing a lookup service useable by management functions, like configuration management. The P2P-based mechanism integrates easily the different access technologies by using a self-organizing overlay, also allowing for expansion of the network with little effort. The lookup service provides features like resilience to node failures, redundant storage of information and new search types. The eCAP architecture uses a Kademlia-based algorithm [MM02] to maintain and organize the neighborhood relationship among the APs.

The paper is organized as follows. First, Section II discusses related network configuration mechanisms. The eCAP architecture is described in detail in Section III. Section IV is devoted to the eCAP mechanisms. It first outlines briefly the original Kademlia algorithm and describes then in detail the enhancements made to Kademlia to enable its use in the eCAP architecture. Section V provides an outlook to the performance evaluation of the eCAP architecture and Section VI summarizes the paper.

II. Related Work

In today's network management for 2.5/3G mobile networks all nodes have to be registered with a so-called element manager (EM) [vR04], which in turn is a part of the main network management system (NMS). This results in a hierarchical, centralized network, which is optimized only for

2.5/3G radio technology and doesn't address other technologies, like WLAN or WiMAX. The EM is a network management entity that configures the individual radio nodes and periodically polls them for status or error information. Via this communication path, the radio node is managed. Automatic detection of neighbours, or direct communication between network elements for network management or configuration purposes, however, is not well-known today. And also the classical simple network management protocol (SNMP) [Sub00] follows the centralized, hierarchical approach.

Since NodeBs currently support IP stacks, e.g. in Release 6 [vR03], and since they can be assumed to be in the range of tens of thousands within one operator network, it makes sense to analyze the options for P2P based self-management.

Regarding self-organization of NodeB (UMTS) network elements, a proposal from NEC [Sim05] suggests to use algorithms belonging to the category of "epidemic computing", i.e. non-deterministic algorithms which are capable of detecting other network elements and, as the final result of the discovery process, the structure of an entire network with numerous nodes.

Another approach for node management in packet switched networks is the Cisco Discovery Protocol (CDP) [Sys], which is media and network protocol independent. It obtains protocol addresses of neighboring devices and then discovers the platform of those devices. It is a proprietary protocol, intended to locate and identify Cisco hardware only, building on top of SNAP frames and involves sending messages to a multicast address periodically.

In [AMH05], the authors present a P2P architecture that allows mobiles to retrieve information about access systems at given coordinates. While geo-location is used to generate identifiers for the documents stored in the overlay, it is not used to build the structure of the network. Moreover, the approach does not cover network management functions.

III. The eCAP Architecture

This section describes the eCAP architecture which supports the configuration management in B3G access networks using a overlay structure. This structure is self-organizing and enables the APs to identify and contact their physical neighbors quickly. It also allows for storing and searching information associated with these APs.

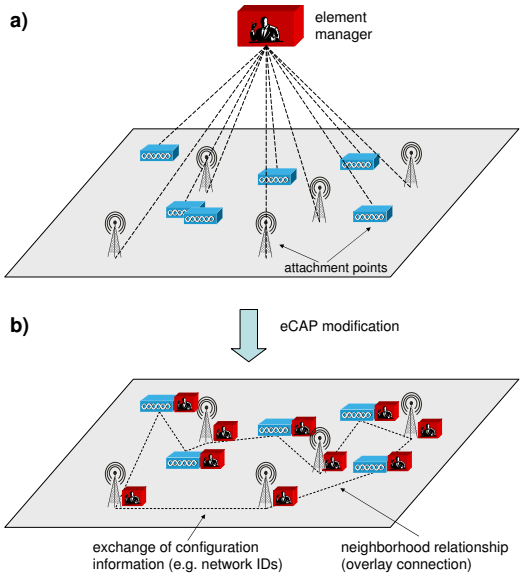

Fig. 1. Administration of attachment points by an element manager (a) and the eCAP modification using an overlay (b)

Typically, in today's 2.5/3G networks a central element manager is administrating multiple attachment points, as illustrated in Figure 1a). The eCAP architecture, now, distributes the functionality of the element manager. Simplified element manager functions are located at the attachment points in the eCAP modification, cf. Figure 1b). The eCAP architecture employs a Kademlia Peer-to-Peer overlay network to quickly store and retrieve documents, i.e. configuration information, between the peers. The APs are the peers in the P2P overlay. The documents contain, for example, network IDs, WLAN SSIDs, or any other configuration data. So, an AP can be represented by one or several documents it stores in the network, each containing information about this AP.

In seamless B3G services, network configuration information is most likely needed for adjacent cells, e.g. for the support of vertical handovers. Therefore, the overlay should reflect the adjacency in its neighborhood relationship. This relationship is accomplished by using the location information of the APs and by structuring the overlay connections according to the physical distances between APs. The structuring is obtained by the eCAP modifications of the Kademlia algorithm which are explained next.

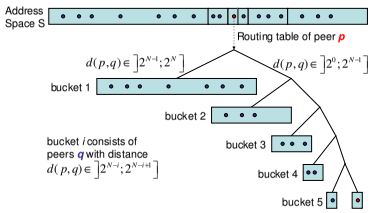

Fig. 2. Constructing the bucket list of a peer with help of the metric

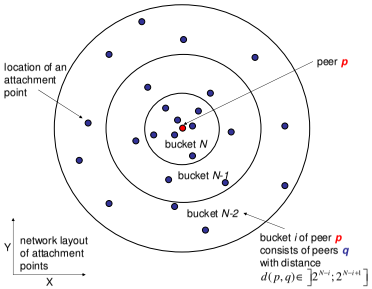

Fig. 3. Buckets of a peer's routing table are mapped to geographical areas

IV. The eCAP Mechanism

A. Standard Kademlia

Kademlia is a DHT-based P2P mechanism to efficiently locate information in an overlay. A hash table is a data structure that associates keys with values. A distributed hash table (DHT) assigns the responsiblity of parts of the value range of the hash function, i.e. of the address space S, to different peers. In order to retrieve the data, DHTs apply sophisticated routing schemes, such as self-balancing binary search trees. Each peer stores contact information about other peers in order to route query messages.

In Kademlia, the branches of the binary search tree are represented as *buckets*, cf. Figure 2, and the collection of buckets form the routing table. The bucket i of peer p's routing table is a list of peers which have a certain distance to peer p. In original Kademlia, 160-bit identifiers for the address space are used and the applied metric is XOR, i.e.,

$$S = \{0;1\}^N \quad \text{with } N = 160 \tag{1}$$

$$\begin{aligned} d: \quad S \times S &\to [0; 2^N], \\ (p,q) &\mapsto p \oplus q. \end{aligned} \tag{2}$$

This means that bucket i in the routing table of peer p covers all peers q with distance $d(p,q) \in]2^{N-i}; 2^{N-i+1}]$, cf. Figure 2. In order to keep the size of the routing table small enough, a bucket has at most k entries and is also refered to as *k-bucket*. This results in a maximal number of routing table entries of $k \cdot N$.

B. Modified Kademlia for eCAP

In order to apply Kademlia for the eCAP mechanism, the metric of the algorithm has to consider the distance between two attachment points in the two-dimensional network layout. Therefore, the standard IDs produced by SHA-1 in original Kademlia are replaced in the eCAP modification by 2D-coordinates and the Euclidean metric is used. The address space S is modified such that the ID of a peer p is a tupel of its geographical position $p = (p_x, p_y)$. Hence, the mapping and the metric is provided by the following equations:

$$S = \mathbb{R} \times \mathbb{R} \tag{3}$$

$$\begin{aligned} d: \quad S \times S &\to [0; 2^N], \\ (p,q) &\mapsto |p - q|, \end{aligned} \tag{4}$$

i.e., $d(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$. To be able to compute the distances between two peers, their coordinates (e.g., GPS coordinates) have to be known.

As a result of the modification, the buckets are mapped onto rings in the geographical network layout, as illustrated in Figure 3. The bucket i of a peer p consists of peers q with distance $d(p,q) \in]2^{N-i}; 2^{N-i+1}]$. Due to the similar range of the metric, the eCAP architecture can apply the original mechanisms for joining of nodes and routing of messages.

Applying the Euclidean metric requires that the parameter N has to be selected appropriately in such a way that all possible distances are covered. For example, in order to cover the whole area of Germany, it is sufficient to use $N = 28$ with length unit of 0.1 m for the discretization of the AP locations.

C. Overlay Structuring and Routing

With these modifications, the overlay sorts itself according to physical closeness without losing the advantages offered by Kademlia, such as scalability and search in $O(\log m)$ for m peers. Since in Kademlia each node sorts its known contacts

according to their distance, and knows more peers that are close than peers that are far away, the entries in the closer k-buckets are more numerous and close to the local node. This means that a large number of neighbors can be found in the closest k-bucket. Thus, the overlay has performed a necessary task via self-organization.

Routing in the described overlay works basically in the same way as in standard Kademlia. However, instead of using the simpler XOR metric, distances have to be calculated to every node that is stored in a k-bucket (this needs to be done only once for these entries). Then, routing means finding the closest node to the target coordinates in one of the buckets. Since the distance to the target node can be computed, each node knows exactly in which bucket the node would fall, and can do a lookup in the right bucket.

A node join procedure works similar to the known method from Kademlia. A new attachment point X joins the system by contacting a known node. The join procedure includes a lookup for the nodes with the numerically closest IDs to the ID of X, thereby allowing X to fill its closest k-bucket. Other buckets are filled over time when X receives messages from other nodes.

While there is no standard leave procedure defined, the original bucket refreshing technique in Kademlia makes sure that node leaves do not go undetected for long. To support our system, a failed node ping must not only have an effect on the bucket of the node that discovered the failure, but must also be disseminated by that node to all entries in its nearest bucket lists.

One problem that remains is that there exist cases where two attachment points are 'neighbors' in the sense that they have overlapping cells, but are some distance apart (e.g., a WLAN cell in a UMTS cell or two nodeBs which have a larger distance than two WLAN access points). This situation mainly occurs if attachment points of different technologies are considered (due to the different cell sizes), where distance is not a primary attribute to decide whether an overlap or handover is possible. To alleviate this, we divide the buckets known from Kademlia into several parts, each part being reserved for entries of just one technology. The relative sizes of these technology sub-buckets is chosen according to the quotient of nodes in the network that support this technology.

D. Reliability, Storage and Retrieval of Data

We can utilize the described network also to store information about the peers participating, and to look up this information in an efficient manner. The documents can include all kinds of data that are of interest for management applications, e.g. configuration data, performance indicators, accounting records etc.

Reliability. Of course, this kind of information can simply be stored on the node that generates this information, to be polled later by any peer that needs the information. However, our approach has some properties that may improve this straightforward approach. Both approaches can be used in parallel, using the traditional method as a default solution. This way, information about a node can be accessed quickly as long as that node is active and connected. If a node crashes or is disconnected for any reason, its information would be lost if it was only stored there. By using the overlay as a backup solution, we additionally replicate the documents on other nodes. Thus, it is still available even if the corresponding node is not available. The information is still stored on a specific location in the network. We will now describe how this location is determined.

Storage and Retrieval of Data. In general, Kademlia stores a document on the node with the closest ID to the document key. Since we have coordinates as node IDs, we have to create similar keys for documents to preserve this function. This means that a document must get a coordinate-like ID. However, we must ensure that an information seeker (e.g., a management application) is able to search for documents, so the ID generation has to be transparent. We can imagine that every node in the network has a unique identifier, e.g., its IP address or any other ID. Since this ID can assumed to be known by network management, we can use it to identify documents concerning the corresponding node. This is achieved by hashing the identifier of a node to the x-coordinate of the Kademlia ID. The second part of the coordinate can be used to identify the content of a document. There might be different sets of information a node can store in the network, e.g., configuration data, performance indicators, accounting data, etc. We simply hash the type of content that is stored in the network to obtain the y-coordinate needed.

Thus, we have a method of generating IDs for documents that a node wants to store in the network to make it accessible for information seekers. If

this information is sought, a search key can be assembled in the same way, hashing the identifier of the node that the information pertains to and hashing the kind of information that is requested. A query containing this search key is then inserted into the overlay network and routed to the node responsible for this key by it.

Range Queries. Another application for the overlay is a search for attachment points in a certain area. Imagine for example a network operator that wants a current list of all APs in the city of Berlin. In general, this means he will specify a center and a radius for the area he wants to cover. With this information, we can conduct an area search by contacting the node nearest to the center of the specified area, which starts an iterative search in the given range.

V. Outlook

Currently, simulations are conducted for investigating the performance of the proposed eCAP architecture. One aspect that is looked at is the resilience of the system, i.e., how it performs when attachment points leave or join the network arbitrarily, the so-called *node churn*. We expect results that are similar to the standard Kademlia performance, since failure detection and the node join procedure was not modified in our approach. Another set of simulations will show how good the coverage of the neighborhood of an arbitrary node is with respect to the bucket sizes, i.e., how many neighbors can actually be found in the closest buckets while not having too large buckets.

First results have shown that with minor modifications concerning the join process and regular updates during the lifetime of a peer, the system is able to stabilize quite well, with each peer knowing a good number of other nodes. This result is shown in Figure 4. When additional traffic is created by search requests for documents, the time until a sufficient number of nodes is known should decrease, since Kademlia utilizes the information gained by handling these requests.

In a next step, we will take a closer look at the performance of the search algorithm, e.g., search times and the tradeoff for the range search. While search speed is not the primary performance indicator for our problem, of course it is not to be neglected. Even more important is the tradeoff between search success (i.e., the persistence of data in the network) and the size of the republish group (i.e., the number of peers one specific document is replicated to).

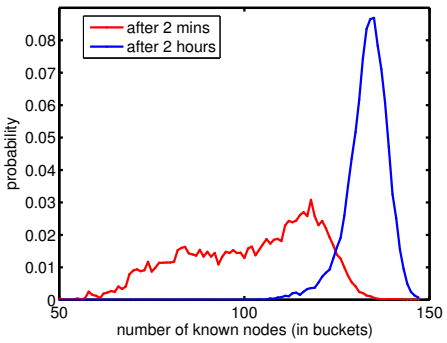

Fig. 4. Stabilization of the network over time.

VI. Conclusion

This paper presented a P2P-based architecture for supporting configuration management in B3G access networks. The *easy Configuration of Attachments Points (eCAP)* architecture enables participating APs to identify their neighbors as well as providing data storage and retrieval service useable by management functions, such configuration management. A Kademlia overlay network is specifically adapted to be used for configuration management for B3G radio access nodes. It offers the advantages of a peer-to-peer overlay, such as resilience, load distribution and scalability. Moreover, it solves specific tasks such as neighbor identification and range queries. We described the changes made to the classical Kademlia architecture and how they are used to achieve the desired functionality.

References

[MM02] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the xor metric. In *Revised Papers from the First International Workshop on Peer-to-PeerSystems*, pages 53–65. Springer-Verlag, 2002.

[Sim05] Schütz Simon. Plug-and-play routers and base stations. Technical report, NEC Europe Ltd., Network Laboratories, 5. Würzburger Workshop "IP-Netzmanagement, Netzplanung und Optimierung", July 2005.

[Sub00] Mani Subramanian. *Network Management: Principles and Practice*. Addison-Wesley, 2000.

[Sys] Cisco Systems. *Cisco Discovery Protocol*.

[vR03] 3GPP TR 21.902 v6.0.0 Release 6(2003.09). Evolution of 3gpp system, 9 2003.

[vR04] 3GPP TS 32.101 v6.1.0 Release 6(2004.12). Telecommunication management; principles and high level requirements, 12 2004.

[AMH05] M. Esterhazy A. M. Houyou, H. De Meer. P2p-based mobility management for heterogeneous wireless networks and mesh networks. Number MIP-0501. 2005.