MAPNaS: A Lightweight, Locality-Aware Peer-to-Peer Based Name Service for MANETs

Thomas Zahn, Jochen Schiller
Institute of Computer Science
Freie Universität Berlin, Germany
{zahn, schiller}@inf.fu-berlin.de

Abstract

Name services for mobile ad hoc networks are essential to discover and bind resources given by their name or URI to a specific network address. As there is no fixed infrastructure available in MANETs, nodes cannot rely on any DNS-like system as is taken for granted in the Internet.

To keep up with the dynamicity of MANETs, a name service needs to be flexible and scalable. For this purpose, we propose MAPNaS (Mobile Ad-hoc Peer-to-Peer Name Service)*, a lightweight and locality-aware peer-to-peer based name service. MAPNaS runs on top of MADPastry [4], a general-purpose DHT substrate especially designed for MANETs.*

Through simulation we compare the performance of MAPNaS – both in terms of the success rate and overall network traffic – against an unstructured, broadcast-based reference approach. We will thereby demonstrate how to efficiently build a peer-to-peer based name service in mobile ad hoc networks.

1. Motivation

To overcome scalability issues and the static nature of dedicated DNS servers, a number of peer-to-peer based name services have been proposed recently for the Internet (e.g. [1]). Common to all these approaches is the idea to replace the static and hierarchical DNS infrastructure with structured P2P networks, also known as DHTs (e.g. [3]). These approaches use the intrinsic load-balancing of DHTs to distribute resources (or references to them, rather) among nodes throughout the network and to discover them later on.

Since MANETs and P2P system share many key characteristics (e.g. lack of fixed infrastructure, self-organization), we present MAPNaS – a DHT based name service for MANETs. Instead of having a number of dedicated directory servers, every MAPNaS node serves both as a resource directory for certain remote resources and as a resource host (for its own resources). MAPNaS efficiently provides resource name resolution, thereby demonstrating how to build such DHT-based services for mobile networks.

2. The MAPNaS Name Service

In MAPNaS, every resource (e.g. a file, a service, etc.) is identified by a unique resource key that is mapped into the logical MADPastry id space. As determined by MADPastry, every node keeps track of the network addresses of those resources whose resource keys it is responsible for. Nodes store the resource descriptors (the resource key along with the physical network address of the provider) they are responsible for in their local MAPNaS repository. Furthermore, every node advertises its own resources that it is willing to share through MAPNaS.

When a node A wants to make a local resource available to other nodes in the network, it needs to assign a hash key to that resource, e.g. by hashing the resource's URI. Using that key, node A will then construct a resource descriptor consisting of the resource key and the physical network address (e.g. IP address) of the resource provider (in this case node A's address). Using MADPastry, the descriptor is routed to the node currently responsible for the resource key. That recipient node will then store the resource descriptor in its local repository.

Resource discovery works analogously. To resolve a resource's URI, a node will simply hash the URI of the resource and route a lookup request to the node currently responsible for that hash key. The eventual destination node will check its local repository and send back the matching resource descriptor.

For the scalability and feasibility of a MANET, it is essential to restrict network traffic to local regions as much as possible. Therefore, MAPNaS makes use of MADPastry's clusters to store local replications of resource descriptors. When advertising a resource, a node will now insert the resource descriptor under two different keys. The first key is the regular hash key (of

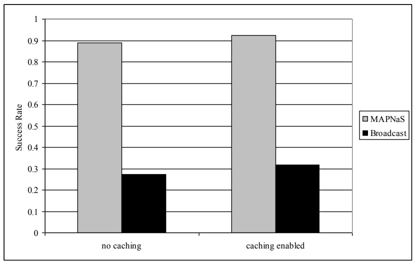

Figure 1. Success rates.

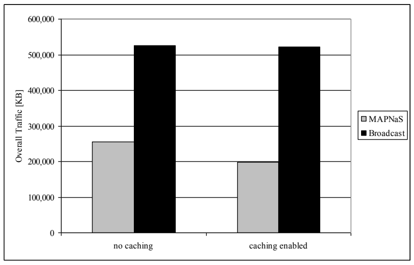

Figure 2. Overall traffic.

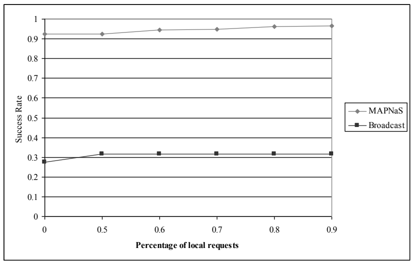

Figure 3. Success rate / locality.

the resource's URI, etc.) and the resource descriptor is inserted into the network as described above. To obtain the second (local) key, the regular resource key's prefix is replaced by the provider's own to make sure the descriptor will be stored in the provider's own cluster.

When a MADPastry node joins a new cluster, it will hand over the resource descriptors it was responsible for under its old overlay id to its old leaf set neighbors and acquire the new descriptors it is now responsible for from its new leaf set neighbors.

In MANETs, a node typically overhears a good number of packets that are not destined for it. Exploiting this virtually cost-free extra information, a MAPNaS node caches the information of all advertisement, handover, and lookup response packets that it overhears.

3. Simulation Results

To evaluate the performance of MAPNaS, we implemented a MAPNaS reference application running on top of a MADPastry routing agent in ns2. All simulations model wireless networks of 250 nodes over the course of one (simulated) hour. Nodes are always moving around according to the random way point model with no pause time (constant movement) and at a steady speed of 1.4 m/s – a quick walking speed. For data transmission, nodes are using the 802.11 communication standard with a transmission range of 250m. The node density in the networks that we investigate is always 100 nodes/km². All nodes send out random requests at a rate of 1 req/10s.

We also implemented a second reference application where nodes do not make their own resources public and resource discovery requests are simply broadcast (already forwarded requests will not be forwarded a second time). Every receiving node checks its own resources and if there is a match, it sends back a direct response using AODV [2].

Figure 1 shows the success rates – the percentage of random lookups that eventually deliver a response containing the correct resource descriptor back to the originator node – for the MAPNaS and broadcast reference application. Both with caching disabled and enabled, MAPNaS achieves significant better success rates (around 90%. vs. around 30%) than the broadcast reference application does (with caching enabled, we are using an expanding ring search so that the broadcast application could also take advantage of locally cached entries). The reason for this becomes obvious in Figure 2 which displays the overall traffic (e.g. AODV route requests, MADPastry packets, resource advertisements, handover packets, etc.) during a simulation run. The broadcast approach produces more than twice as much traffic as MAPNaS does, resulting in a highly increased number of packet collisions/drops which brings down the success rate.

Figure 3 shows how the success rate of MAPNaS further improves when the traffic pattern is such that a certain percentage of requests can be satisfied inside the requestor's own cluster (local requests). As there are no local clusters in the broadcast application, we use the expanding ring search results to provide a reference line.

Conclusion & Open Questions

MAPNaS demonstrates how to efficiently build a DHT-based name service for MANETs. It significantly outperforms a simple broadcast-based approach. In the future, it will be interesting to see how MAPNaS fares compared to more elaborate MANET name services, especially other cluster-based ones. It would also be interesting to investigate the behavior of MAPNaS under varying network conditions such as churn, varying node densities or different traffic patterns.

References

[1] R. Cox, A. Muthitacharoen, and R.T. Morris. "Serving DNS using a Peer-to-Peer Lookup Service". In *Proc. of IPTPS*, February 2002.

[2] C. E. Perkins and E. M. Royer. "Ad hoc on-demand distance vector routing". In *Proc. of IEEE WMCSA*, February 1999.

[3] A. Rowstron and P. Druschel. "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems". In *Proc. of Middleware*, November 2001.

[4] T. Zahn and J. Schiller. "MADPastry: A DHT Substrate for Practicably Sized MANETs". In *Proc. of ASWN*, June 2005.