Towards Self-Organizing, Integrated Service Placement in Ad Hoc Networks

Georg Wittenburg and Jochen Schiller

Department of Mathematics and Computer Science
Freie Universität Berlin, Takustr. 9, 14195 Berlin, Germany
Email: `{wittenbu,schiller}@inf.fu-berlin.de`

1 Background

It has been recently advocated [1, 2] that nodes in ad hoc networks should cooperate by dynamically choosing which nodes are to provide application-level services to other nodes. Service placement [3] addresses the questions of *how many* instances of the same service should be available; *where* these service instances should be placed, i.e., which nodes are best suited for hosting them; and *when* to adapt the current service configuration. The benefit of this approach is that the service configuration, i.e., the set of nodes to host a service, is adapted automatically at run-time, thereby reducing overall network traffic and latency.

A key aspect in selecting a service configuration is the traffic between service instances that is required to keep global state and data synchronized. For high synchronization demands, e.g., for transactional databases, the optimal service configuration is to create only a single instance of the service (cf. Fig. 1). On the other extreme, if no synchronization is required, e.g., for a spell checking service, each client hosts its own service instance (cf. Fig. 3). For the more interesting case of moderate synchronization demand, e.g., for services such as DNS or WWW, the most cost effective way to provide a service is to dynamically create service instances in the vicinity of clusters of high demand (cf. Fig. 2).

2 The SP*i* Service Placement Architecture

We propose a novel approach to service placement that optimizes the number and the location of service instances based on usage statistics and a partial network topology derived from routing information. The SP*i* service placement architecture takes advantage of the interdependencies between service placement, service discovery and the routing of service requests and replies. The system only requires minimal knowledge about the service it is tasked with placing in the network. It is furthermore unique in that it explicitly considers the communication between service instances that is required to synchronize shared data.

In our architecture, a middleware collects usage statistics of each service instance by inspecting service requests and replies. Additionally, local network topology information is extracted from enhanced routing packets which piggyback data concerning routing paths and neighborhood connectivity. This data is transmitted periodically to a service-specific coordinator node, which calculates the optimal service configuration using a cost metric based on service demand and the network topology. The node then establishes a set of actions required to migrate from the current to the optimal configuration, and issues the commands for adapting the configuration to the nodes that currently host service instances.

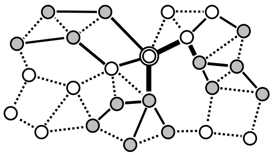

Fig. 1. High sync traffic; one single service instance

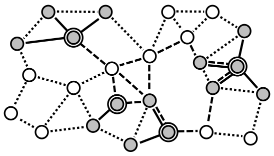

Fig. 2. Moderate sync traffic; several instances

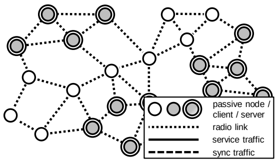

Fig. 3. No sync traffic; one service instance per client

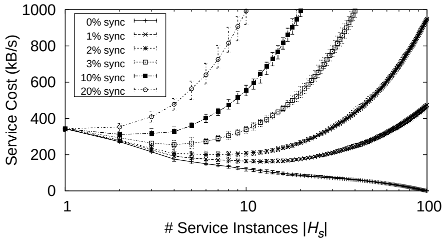

Fig. 4. Service cost as a function of different service configurations

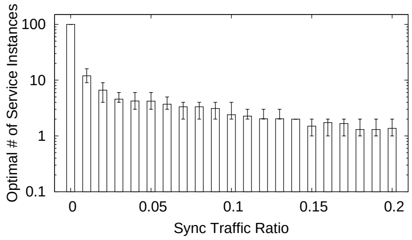

Fig. 5. Optimal number of service instances depending on sync traffic ratio

3 Preliminary Results

In our preliminary evaluation, we focus on the general applicability of our system. We used the `ns` network simulator to study an IEEE 802.11 network of 100 nodes aligned in a fixed 10×10 grid at a distance of 30 m and with an average radio transmission radius of 50 m. We assume that the traffic required for synchronization between service instances is a simple linear function of the traffic caused by service requests, and call this dependency the *sync traffic ratio*.

Figure 4 shows the service provisioning cost for different numbers of service instances and values of the sync traffic ratio. The cost metric is the bandwidth used for service provisioning. For sync traffic ratios between 1% and 20%, the cost functions have a minimum between 13 and 1 service instances respectively. Figure 5 shows the number of service instances of the optimal service configuration for different sync traffic ratios. We observe that these configurations always consist of multiple service instances for sync traffic ratios below 15%.

We conclude that our approach is beneficial for services with synchronization traffic below 15%. This supports our claim that the SP*i* service placement architecture is applicable to wide-spread services such as DNS or WWW.

References

1. J. Gossa et al. Proactive Replica Placement Using Mobility Prediction. In *Proc. of DMCAC '08*, Beijing, China, Apr. 2008.
2. M. Lipphardt et al. DySSCo - A Protocol for Dynamic Self-organizing Service Coverage. In *Proc. of IWSOS '08*, Vienna, Austria, Dec. 2008.
3. G. Wittenburg and J. Schiller. A Survey of Current Directions in Service Placement in Mobile Ad-hoc Networks. In *Proc. of PerCom '08*, Hong Kong, Mar. 2008.