Implementing the Future Internet:
A Case Study of Service Discovery using Pub/Sub in the ANA Framework

Theus Hossmann, Ariane Keller, Stephan Dudler, Martin May
Communication Systems Group, ETH Zurich
Gloriastrasse 35
8092 Zurich, Switzerland
firstname.lastname@tik.ee.etz.ch

Abstract

The goal of the Autonomic Network Architecture (ANA) project is to define and implement a novel, flexible and extensible network architecture to interconnect and federate multiple heterogeneous networks in an autonomic way. In order to achieve this, a framework which allows to dynamically compose a protocol stack from different modules, called Functional Blocks, was built and is currently in the prototyping phase. In this paper we discuss how the flexibility and modularity of the ANA framework helps to integrate new protocols beyond the legacy IP world. As a case study we describe the implementation of a Publish/Subscribe system. Specifically, we use Pub/Sub mechanisms for service discovery in a wireless self-organizing network by routing service subscriptions to published services, with a field based routing protocol. We exemplarily decompose the functionality of such a system into small, exchangeable Functional Blocks. By means of this example we demonstrate the benefits of such a flexible architecture and show how it enables truly autonomic networking.

1 Introduction

The goal of the ANA project [1] is to explore novel ways of organizing and using networks beyond legacy Internet technology and to demonstrate the feasibility of autonomic networking. To reach this goal, we have designed and implemented a framework[1] which allows to dynamically compose protocol stacks at runtime, according to the current needs of the network. For example, security mechanisms are added on the fly if a device is in a hostile environment.

In the following, we will give a brief overview of the relevant concepts and paradigms of the ANA framework [5][6]. Basically, we introduce a new network abstraction used to describe the interactions in the ANA framework. In a nutshell, we distinguish three basic elements: Functional Blocks (FB) - elements that implement protocol or other functionality; Information Dispatch Points (IDP) - used to address the Functional Blocks; and Information Channels (IC) - used to illustrate the communication with a distant node. Fig. 1 depicts the internal organization of a node in the ANA world.

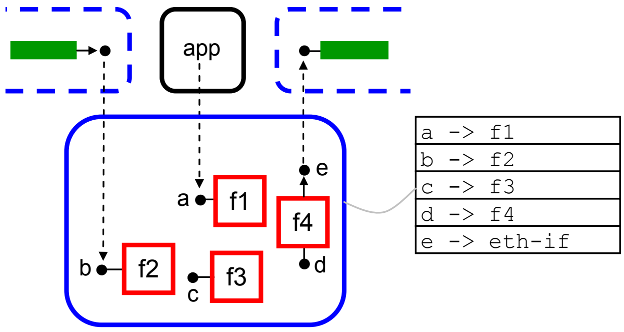

Figure 1. Internals of an ANA node

The boxes, labeled `f1 - f3`, represent *FBs*, dynamically composable units of functionality. Data encapsulation, checksum computation or encryption are examples of such Functional Blocks. Each FB has one or multiple *IDPs* attached to it, labeled `a - e`.

Conceptually, an IDP is similar to a file descriptor in a file system. IDPs and their bindings to FBs is eminently important to the ANA world, since it allows to dynamically compose and recompose whole protocol stacks as we will describe in the following. When data is sent from one FB to the next, it is not addressed explicitly to the destination Functional Block, but to an IDP instead. What function is behind this IDP is transparent to the sending FB and can change, since the ANA framework allows to

[1] The ANA framework is currently in prototype phase. The first public release is scheduled for July 2008.

bind and rebind IDPs to FBs at runtime. Thus, the concept of IDPs is a layer of *indirection* between the individual functions. This indirection enables the ANA framework for *autonomic communication*. Imagine, that the framework notices that the node is moving from a secure to a hostile environment[2]. By simply rebinding some IDPs the framework can, transparently to the rest of the protocol stack, add an encryption FB before sending data to the wire or the air.

An Information Dispatch Point can not only be bound to a FB, but also to an IC, the ANA abstraction for a (unidirectional) communication channel. Since we do not use the concept of ICs here, we will not further elaborate on this.

In order to dispatch the data within a node, the node maintains an *Information Dispatch Table* which keeps track of the binding of IDPs to FBs (resp. ICs). Modifying the protocol stack according to the current needs of the network is simply a matter of changing the binding entries in this IDT.

The flexible and IP independent architecture of the ANA framework allows the integration of new communication paradigms and protocols for which the legacy Internet architecture is not well suited. As an example of such a communication paradigm, we discuss a *Publish/Subscribe* system in the following. In the Pub/Sub paradigm, messages are not sent to a specific receiver address, but instead to all nodes that indicate their interest by a subscription [3][4]. In our specific example, we envision a Pub/Sub system for *service discovery* in wireless self-organizing networks. For instance a node which is connected to the Internet can publish its service using the keyword ”gateway”. Nodes that are interested in accessing the Internet can then subscribe to the service gateway.

In order to route the service subscriptions to the published services, we use a field based approach as described in [7]. In field based routing, for each service that gets published, a potential field is established in the network. This field is analogous to a physical potential field, for instance an electrostatic field. Similar to routing using potentials introduced in [2], a service (e.g., a printer or a gateway) is modeled as a positive point charge and a service request message (e.g., a print request) is modeled as a negative charge which is attracted by the service instances. If there are several nodes publishing the same type of service, the potential field is the superposition of the individual fields. Fig. 2 shows a hypothetical superposition of the fields of ten service instances.

A node n calculates its potential in the field according to

$$\varphi(n) = \sum_{j=1}^{N} \frac{Q_j}{|n - n_j|}, \quad (1)$$

where N is the number of service instances (e.g., the number of gateways), Q_j is the service capacity (e.g., the bandwidth of the gateway) of a specific instance and $|n - n_j|$ is the distance (in number of hops) of node n to this service instance. Each node then forwards the requests it gets for a certain service according to the steepest potential ascent, that is, to the neighbor with the highest potential for the given service.

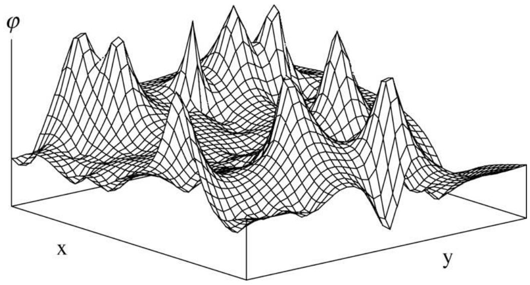

Figure 2. Illustration of a potential field with 10 service instances

2 Implementation

The main goal of this case study is to show the flexibility of the ANA architecture. As mentioned before, we selected a Publish/Subscribe system as example system, implemented independent of IP. That means that we use an IP agnostic addressing scheme and we will also send the messages directly over to the Ethernet interface. Since our target scenario of a wireless self-organizing network is typically limited in size, we do not have to rely on the hierarchical IP addressing, thus do not require IP. However, note that because of the modularity of the protocol stack, one could easily run the same system on top of IP in ANA[3]. In a first step, we have identified the individual building blocks of the Publish/Subscribe system. Each of these Functional Blocks can be developed independently of the other blocks, if the communication interfaces between the different FBs are given. In the first part of this implementation section, we describe the decomposition of the publish subscribe system, whereas in the second part, we shortly discuss the interfaces between the different FBs as well as the messages exchanged between the nodes. Finally, we will show a validation scenario in a wireless testbed.

2.1 Decomposition

The Publish/Subscribe system has mainly three tasks: framing, forwarding, and routing. Fig. 3 illustrates the overall implementation, including the decomposition of the networking functions.

[2]This would be noticed by the *monitoring* part of the framework, which we do not describe further here.

[3]In fact, we have an ”lightweight” IP implementation in the ANA framework which offers basic IP functionality including RIP routing.

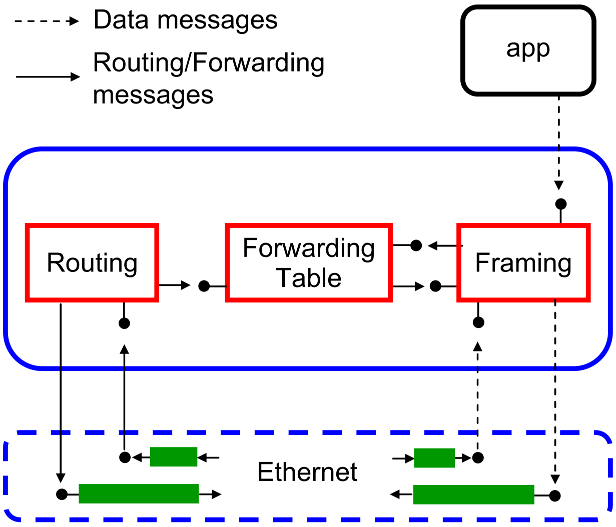

Figure 3. Routing and forwarding in ANA

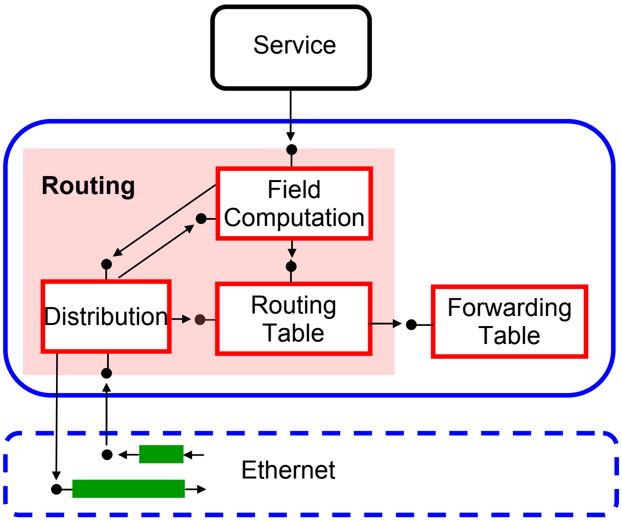

Figure 4. Field based routing decomposition

The framing FB is responsible for handling service discovery request messages from an application. Thereto, the framing FB encapsulates the data with a packet header. In order to forward the message to the next node, it needs to know the IDP to which to send the message. This IDP is received from the forwarding FB. The forwarding FB holds a list of all known services along with the IDPs to be used to forward the data to the next hop. The forwarding FB is updated by the routing FB which is responsible for the computation of the routes. We have separated routing from forwarding in order to be able to exchange the routing process or to add a second routing process without the need of modifying the framing process. Regardless of the number and characteristics of the routing processes the framing FB can use one single interface to get the next hop address.

The routing FB is the most complex block, since it needs to communicate with other nodes as well as to compute the routes. In our case study we use field based routing. For this protocol the routing block can be further subdivided into: a field computation, a routing table and the distribution of routing information. This decomposition is depicted in Fig. 4.

Field based routing is designed to work in a dynamic environment in which connections between nodes change frequently. To provide an adequate route we have to continuously recompute the actual state of the routing field. Therefore a service which wants to be available in the network has to send periodic publish messages. These messages are handled by the field computation FB which broadcasts the field information (e.g. service type, capacity of service, etc.) to the network. Based on this field information the field computation FBs in the other nodes are able to calculate their potential. In a second step this potential is sent to all neighbors of a node. This information is needed from the routing table FB of the neighboring nodes to determine the next hop (the node with he steepest gradient) for a given service.

Since the sending of the routing information is a generic task, which has nothing to do with the field computation, we have developed the distribution FB which is solely responsible for the distribution of the routing information.

Due to the decomposition of the FBSD system into individual FB each functionality can be further developed independently. For example it is possible to develop a new field computation FB which computes the potential field with an other potential function. The only thing that needs to be known is the communication interface with the other FB.

2.2 Communication Interfaces

A major difficulty in providing a generic Publish/Subscribe system lies in the interfaces between the different elements. For example it has to be known which field function should be used for a given service type or how the routing messages are formatted. For this purpose we make use of a message format based on on XRP (eXtensible Routing Protocol) [8]. This message format can be seen as a container which may hold arbitrary data. It is based on a set of COMMANDS and ATTRIBUTES. Each message consists of one COMMAND which identifies the purpose of this message and a set of ATTRIBUTES which specify some parameter for the COMMAND (e.g. service type, time to live, sequence number etc.). We use this message format for the communication within one single node as well as for the communication between different nodes. This allows us to exchange a given Functional Block as long as the new Functional Block adheres to the same XRP commands.

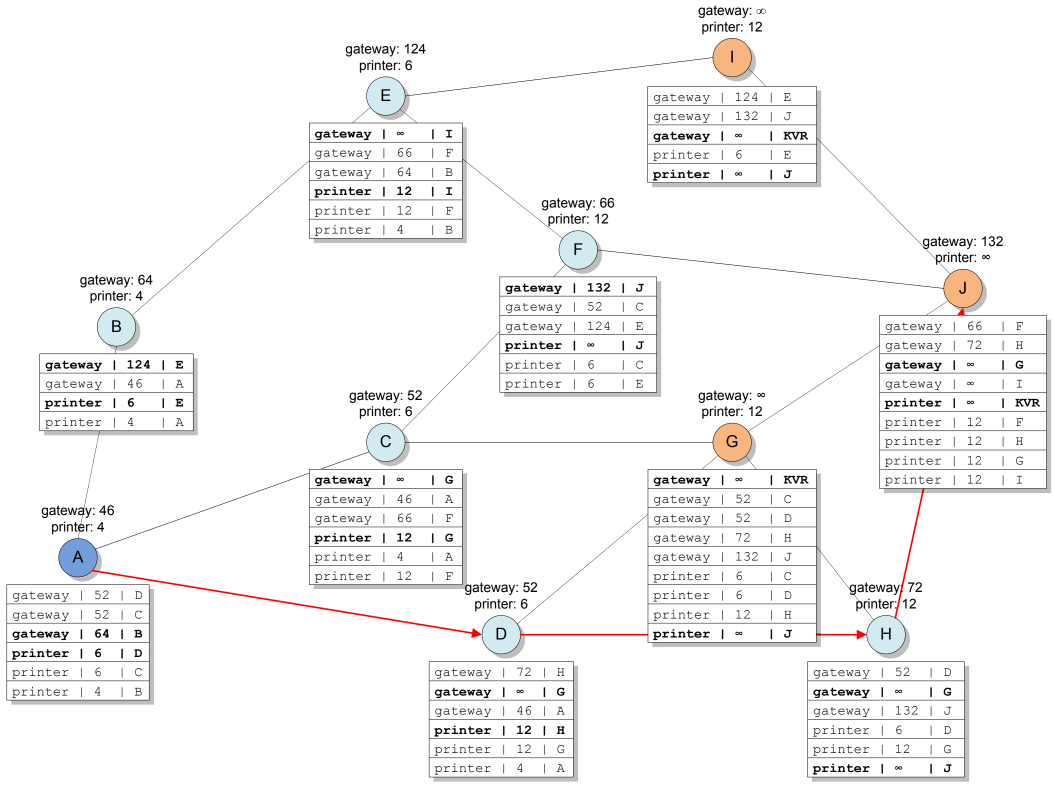

Figure 5. Validation in wireless network with 10 nodes and two services

2.3 Validation Scenario

In order to validate our implementation, we set up different test scenarios in our wireless mesh testbed[4] on which we run the ANA prototype. For the validation we use a topology of 10 nodes (labeled `A - J`), connected with 802.11b WLAN in ad hoc mode. We validate the implementation under different scenarios, such as multiple published services, multiple service instances of the same service and node failures.

In a first scenario, node `J` publishes a print-service (with service capacity 12) in the network. Thus, `J` broadcasts a XRP publish message containing the identifier string `<printer>` to the network. Remind that we do not use IP and its addressing scheme in our Publish/Subscribe protocol. Instead, node `J` is simply addressed with the identifier `<printer>` after the potential field is established. Each node, when receiving a publish message, calculates its potential according to Eq. 1 and propagates the message containing its own potential further to its direct neighbors. All nodes store their own potential value for the given service. Further, all nodes maintain a routing table containing the potential values of their neighbors. These values are used to calculate the local potential, as well as to determine the next hop of a message by following the steepest gradient. Once the potential field has propagated and reached a steady state, subscribe messages with identifier `<printer>` are routed toward node `J`.

After validating this basic scenario, we add another service as a next step. Two nodes, `G` (service capacity 12) and `I` (service capacity 120) now both offer Internet connectivity and publish the identifier `<gateway>` and their respective service capacities to the network. Thus, we have now two different services, `<printer>` with only one service instance and `<gateway>` with two service instances. The potential field for the new service is established the same way as described above, only this time is the superposition of the individual fields of `G` and `I`.

Fig. 5 depicts the resulting potential fields having published both services (three service instances in total) in the same network. The node potential is indicated above the nodes, service nodes have the potential ∞ by definition. The routing tables are shown below the nodes with entries being triples of service identifier, potential and next hop identifier. The special next hop identifier `KVR` indicates that the given service is available locally. The best next hop for both services is highlighted in the routing tables. The routing FB writes this entry into the forwarding table.

Looking at the local potential values of the `<printer>` field we observe, as expected, that the direct neighbors of node `J` have potential 12. Nodes

[4] See http://tiknet.ee.ethz.ch/ for more information about the testbed.

at two and three hops distance have potentials 6 and 4 respectively. Exemplarily, the path from node A to the print service is highlighted in the figure. For the <gateway> service, the computation of the potential is slightly more difficult, since the field is the superposition of two service instances. For example, node E is one hop away from node I (with service capacity 120) and three hops away from node G (capacity 12). Thus, its potential is $\frac{120}{1} + \frac{12}{3} = 124$. Having two instances of the same service leads to the situation that <gateway> subscribe messages are routed to either of them. In the example nodes A, B, E are subscribed to I and C, D, F, H, J are subscribed to G.

Note that the goal of this section is not to evaluate the performance of the Field Based Service Discovery. Instead, we use Field Based Service Discovery as a showcase to illustrate the capability of the ANA framework to implement new communication paradigms. Of course, we also validated our implementation under realistic circumstances and were able to demonstrate a non-IP networking example.

3 Discussion

As a case study, we have shown the decomposition of a field based Pub/Sub system into its Functional Blocks and how they collude. The ANA framework allows to compose and adapt the protocol stack at runtime and, because of its independence of legacy Internet technology, allows to implement newly emerging communication paradigms and protocols which go beyond the limitations of the well known IP world.

This example is supposed to be educative and the principles learned should be applied to the implementation of further protocols as well. For routing protocols, the decomposition of routing, forwarding and framing Functional Block can be seen as a generic approach which should be followed in future implementation efforts. Also the dissection of the routing FB into a route computation block, a distribution block and a routing table Functional Block is valid in general.

Because in ANA messages are dispatched to IDPs, and the framework allows to dynamically change the binding between IDPs and the Functional Blocks, parts of the protocol are interchangeable, even at runtime. Whereas this indirection through IDPs offers a great new flexibility, it comes at the cost of some additional overhead during the processing of packets. In the future we will further investigate the tradeoff between flexibility and overhead. In order to change parts of the protocol, it is essential to define clear interfaces between the Functional Blocks. We have seen how in ANA the XRP message format is used as a generic message format, by means of which such interfaces can be easily defined.

Having implemented the prototype framework of this network architecture, in the future the project will focus on implementing autonomic features. Thus, we will investigate how the protocol stack can be tuned at runtime in order to achieve the desired self-* properties, such as self-organization, self-configuration, self-protection and self-healing. To achieve these properties, we need two more building blocks: *Monitoring* and *Functional Composition*. The monitoring subsystem will monitor the local system state as well as local and distributed network properties. In autonomic systems, monitoring is of eminent importance since the first step to achieve the desired self-* properties is to be aware of the current state of the system. The functional composition subsystem reacts to the information it gets from the monitoring part and tunes the protocol stack according to the current state of the node and network. It does so by making use of the rebinding capabilities of the IDPs to FBs, as described in Section 1. Amongst others, monitoring and functional composition are two of the major parts we are currently working on in the ANA project.

References

[1] Autonomic network architecture. *http://www.ana-project.org*.

[2] A. Basu, A. Lin, and S. Ramanathan. Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm. In *Proceedings of ACM SIGCOMM*, 2003.

[3] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Achieving scalability and expressiveness in an internet-scale event notification service. In *Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing (PODC)*. ACM, 2000.

[4] Y. Huang and H. Garcia-Molina. Publish/subscribe in a mobile environment. *Wirel. Netw.*, 10(6):643–652, 2004.

[5] C. Jelger, C. Tschudin, S. Schmid, and G. Leduc. Basic abstractions for an autonomic network architecture. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2007.

[6] A. Keller, T. Hossmann, M. May, G. Bouabene, C. Jelger, and C. Tschudin. A system architecture for evolving protocol stacks. In *Proceedings of the17th International Conference on Computer Communications and Networks (ICCCN)*, 2008.

[7] V. Lenders, M. May, and B. Plattner. Service discovery in mobile ad hoc networks: A field theoretic approach. *Pervasive and Mobile Computing*, 1(3):343–370, September 2005.

[8] C. Tschudin and R. Gold. Network pointers. *SIGCOMM Comput. Commun. Rev.*, 33(1):23–28, 2003.