A Generic Active Service Deployment Protocol

M. Sifalakis, S. Schmid, T. Chart and D. Hutchison

Computing Department,
Lancaster University, UK

{mjs, sschmid, chart, dh}@comp.lancs.ac.uk

1 INTRODUCTION

The diversity of active/programmable network application domains has led to different active programming paradigms and a variety of active network approaches that often lack interoperability. Moreover, as network services are becoming more sophisticated/complex and composite, it is less likely that a single active network approach will outclass all others.

From a network client point of view, however, it is important to adopt a uniform/generic interface to take advantage of the programming capability of any available active network platform.

The lack of generic solutions that allow different programming models (i.e. integrated and discrete) and on various levels (e.g. transparently in the forwarding path or application layer) calls for a system-independent active service[1] deployment framework, hiding the heterogeneity of individual active networking implementations.

One way to address the issue is to standardise a common "programming interface" that suits all active network systems. However, as previous standardisation of programming interfaces has shown (e.g. POSIX), this process can be tedious and lengthy.

Alternatively, one can unify the deployment and control of network-side services by providing a "common service interface" across existing active network systems.

In this paper we therefore propose an architectural framework backed by a generic protocol implementation for the dynamic deployment and control of active services. It decouples the mechanisms for querying capabilities, loading code, parameters etc, and activation of the active networking code, in a way that is independent of the active node platform or programming paradigm.

As fundamental research on active networks is slowing down and early deployments are anticipated, we believe that addressing the interoperability problem is timely.

2 RELATED WORK

Work in the area of active service deployment has been traditionally tailored to the needs, and thus tied to the interface, of each specific platform. As a result little work has been done towards an approach that would evaluate the design space for a generic interface

The Active Networks Daemon [2] forms the core service deployment and management module for ABone[2] [3]. It uses the ANEP protocol [4], to provide a generic means of assigning "active packets" to evaluation environments. Although Anet/ANEP provides some limited extensibility to accommodate new evaluation environments, it does not provide any design primitives for querying capabilities, loading parameters, activating and configuring the active code.

3 SERVICE DEPLOYMENT INTERFACE

The service interface offered to a client (i.e. an end-user or software agent in need of an active service) in order to deploy and use an active service should be simple with the minimum possible primitive functionalities. Our experiences have shown that it can be reduced to the following primitives:

- (Un-)load active service
- (Re-)configure active service
- (De-)activate active service

As a result the fundamental functionalities of a generic service interface can be summarised as follows:

[1] We consider both low-level (basic functionalities) and high-level (potentially composed of one or more basic ones).

[2] The ABone is a virtual testbed for active networks, providing an overlay active network on the Internet.

- *Active Node Discovery* – (by new clients in a network)
- *Query Existing Services* – offered by an active node (if needed).
- *Query Active Node Capabilities* – a client should be able to discover what service loading methods the active node supports. This feature must account for extensibility of the loading/deployment methods.
- Service Deployment/Removal – a client must be able to request the deployment of an active service using one of the supported methods.
- Service (Re-)Configuration – a client should be able to (re-)configure a (parameterised) service without reloading or duplicating it.
- Service Activation/Deactivation – to start (or stop) a loaded service.

Depending on the service point in a network, the implementation of the service interface must satisfy all or some of the following requirements:

- ***Extensibility*** – This is a critical requirement in order to account for the integration of new or evolving service loading methods.
- ***Performance (Speed/Lightweight)*** – This requirement is twofold as it encompasses response latency and traffic overhead. Depending on the application domain of an active service, latency and bandwidth may be crucial in concluding about the deployment of the service (ad-hoc networks, roaming/mobile users, etc). As a result the design/ implementation should impose the minimum possible overhead.
- ***Security*** – The notion of active networking *per-se* raises many security considerations. It is imperative that an active service deployment interface implementation must not circumvent (or impose any restrictions on) the security model provided by the active node platform. An open interface to the node-local security and policy subsystem must be provided to permit access control to the functions exported by the service interface.

4 ACTIVE SERVICE DEPLOYMENT PROTOCOL

Part of the work for the ProgNet project [5] in Lancaster University, focuses on the development of an Active Service Deployment Framework. The ASDP protocol presented in this paper is part of this effort.

The current implementation does not aim to address the active node discovery issue, but concentrates on the dynamic deployment of active services in a heterogeneous active network environment.

The protocol is extensible and hence accounts for the incorporation of new active service loading mechanisms. It is based on a simple exchange of a minimal number of messages in order to negotiate and request support for active services.

ASDPv1.0 provides 8 different message types that can be roughly classified in 3 categories: request messages, reply messages and error messages. This message set can be used to query, load, configure and activate active services. Additionally, a set of message options are used to enable active code software updates, control advertisement of the available service loading methods, eliminate duplicate loading requests, etc. Due to the space restriction, we defer further details to the full paper.

Figure 1 shows the main modules of our implementation. Briefly:

- The *core module* provides the message handling and processing routines, and the protocol finite state machine.
- The *NodeOS "mapper" library* exposes a uniform API to the active NodeOS policy and security subsystem (and other services).
- The *active service loading methods libraries* implement the available active service loading methods. They can be accessed through ASDP, and are implemented as dynamically linked libraries for flexibility (runtime deployment).
- The *registry module* maintains a list of supported loading methods.

5 CONCLUSION

This paper motivates the need for an architectural framework for the dynamic deployment and control of active services in a heterogeneous network environment. The primary aim is the decoupling of the loading, configuration and activation mechanisms in a way that is active node platform or programming paradigm independent.

We discuss the central primitives and fundamental functionalities that a generic service interface should support and present the ASDPv1 implementation providing such an interface.

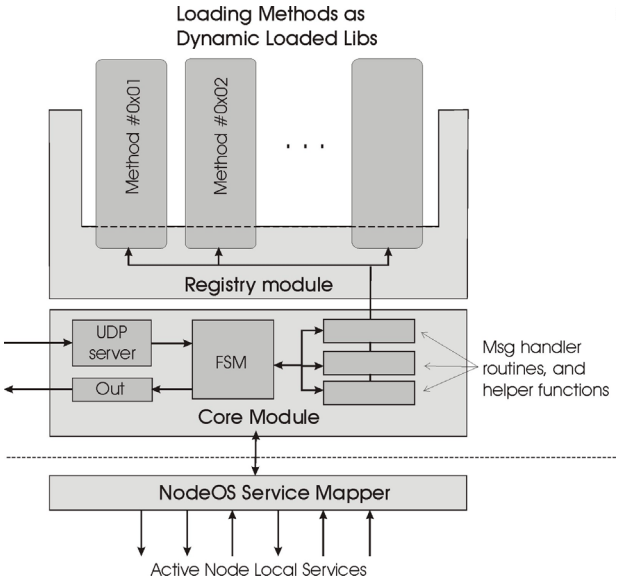

Figure 1. ASDP implementation in LARA++ platform [7]

References

[1] S. Simpson, M. Banfield, P. Smith, and D. Hutchison, "Component Selection for Heterogenious Active Networking", in Proceedings of the 3rd International working Conference on Active Networks (IWAN), volume LNCS 2207, pages 84-100, Philadelphia, USA, September 2001.

[2] "Anetd: Active Networks Daemon", ACTIVATE project, ISI & SRI, http://www.sdl.sri.com/projects/activate/anetd/

[3] B. Braden and L. Ricciulli. "A Plan for a Scalable ABone – A modest proposal", Technical Report, USC – Information Science Institute, January 1999.

[4] D.S. Alexander et al, "Active Network Encapsulation Protocol (ANEP)", Internet draft, IETF, July 1997.

[5] "Programmable Network Support for Mobile Services", Research Project funded by EPSRC Programmable Networks initiative, Lancaster University, 2001.

[6] S. Simpson, P. Smith, M. Banfield, D. Hutchison, "Component Compatibility for Heterogeneous Active Networking", Presented at IEE Informatics - Application Level Active Networks: Techniques and Deployment, 1st November 2000, London, UK.

[7] S.Schmid, J.Finney, A.Scott and D.Shepherd, "Component-based Active Network Architecture", in Proceedings of 6th IEEE Symposium on Computers and Communications (ISCC), pages 114-121, Hammamet (Tunisia), July 2001.

[8] D.J. Wetherall, J. Guttag, and T.L.Tennenhouse, "ANTS: A toolkit for building and dynamically deploying network protocols", in Proceedings of IEEE Openarch, April 1998.

[9] M.W. Hicks, P. Kaddar, J.T. Moore, C.A Gunter and S. Nettles, "PLAN: A Packet Language for Active Networks", In Proceedings of 3rd ACM SIGPLAN International Conference on Functional Programming, pages 86-93, 1998

[10] S.Schmid, J.Finney, A.Scott and D.Shepherd, "Active Component Driven Network Handoff for Mobile Multimedia Systems", in Proceedings of Interactive Distributed Multimedia Systems and Telecommunications (IDMS), Lecture Notes in Computer Science (vol.1905), Springer Verlag, pages 266-278, Enschede (Netherlands), October 2000.

[11] Paul Smith, Steven Simpson and David Hutchison, "Peer-to-Peer Networking for Programmable Service Deployment", in Proceedings of Practical Programmable Networks: Making Inroads into the Internet IEEE, OpenArch 2002 - Short Paper Session, 28th - 29th June 2002, New York, USA; pp 41 – 46.

[12] "MSRL – Mobile-IPv6 Systems research Lab". Research Project funded by Cisco Systems, Microsoft Research (Cambridge), and Orange Ltd, Lancaster University, http://www.mobileipv6.net/, 2001

[13] Stefan Schmid, Tim Chart, Manolis Sifalakis, Andrew Scott, "Flexible, Dynamic, and Scalable Service Composition for Active Routers", IWAN 2002: 253-266