A flexible router platform for next generation network services

Lukas Ruf, Arno Wagner, Karoly Farkas, Bernhard Plattner
Computer Engineering and Networks Laboratory (TIK)
Swiss Federal Institute of Technology (ETH) Zurich
CH-8092 Zurich/Switzerland
{ruf,wagner,farkas,plattner}@tik.ee.ethz.ch*

10th November 2004

Abstract

Autonomous services need a flexible router platform that provides the mechanisms to install, modify and remove services at run-time of the node without interfering with others. Instantiated services must have the ability to re-configure themselves and to exchange service functionality on demand. Envisioned router platforms must be able to run multiple services in parallel and are required to scale with the number of network-interfaces while they need to provide a straightforward to use service programming interface.

In this paper, we present the PromethOS NP router platform together with a service architecture to counteract distributed denial of service attacks in an autonomous, policy-based way. PromethOS NP manages and controls a processor-hierarchy composed of host processors and network processors embedded in network interface cards. It provides a dynamically code-extensible router platform of which all processor tiers are at run-time programmable following a unified component programming model. The service architecture illustrates the capabilities of the router platform and its applicability to autonomous network services.

1 Introduction and Motivation

Routers in the not-too-distant future need to provide extensible mechanisms to process packets in addition to legacy packet routing and forwarding. Extended packet processing may range from in-depth packet inspection to service-specific packet transcoding for, e.g. content-dependent filtering or advanced media adaptation, respectively. While todays software routers are able to give proof-of-concept, they fail at high-performance and scalability.

Recently developed network processors provide a suitable processing element to be embedded at the link interface. Together with managing host processors, they provide a

*This work is partially sponsored by the Swiss Federal Institute of Technology (ETH) Zurich, the Swiss Federal Office for Education and Science (BBW Grant 99.0533), the Swiss National Science Foundation under Grant 200021-102026/1 and Swiss Academic Research Network (SWITCH). PromethOS v1 has been developed by ETH as a partner in IST Project FAIN (IST-1999-10561).
An extended version of this paper has been published at the 6th International Working Conference on Active Networking (IWAN) 2004.

Dagstuhl Seminar Proceedings 04411
Service Management and Self-Organization in IP-based Networks
http://drops.dagstuhl.de/opus/volltexte/2005/94

perfect attempt to increase processing capacity in a scalable way for high-performance routers. While a hierarchical network node (built of several host and network processor based link interfaces) overcomes limitations in performance and scalability, it increases, however, the complexity in management and control of a router platform.

Network services are urgently required that are deployed in the network at critical locations to protect the vital communication infrastructure of today. Present day communication infrastructure has been seriously threatened by large-scale distributed denial of service (DDoS) attacks in the Internet. These attacks destroy information or hinder customers from accessing specific services. Services provided in the Internet like on-line stock trading, virtual travel agencies or book-stores are very important to economy already today. The Economist reported in May 2004 [8]: "The 200m Americans who now have web access are likely to spend more than US$120 billion online this year." But in eCommerce, brief inaccessibility of services results in loss of business [8]. Since the impact of eCommerce on economy is expected to grow further, the risk of economic damage resulting from a large-scale Internet attack increases [1]. The situation becomes more dramatic because the number of attacks increases at least at the same pace as the impact of eCommerce does. Of further threatening importance is the fact that newly discovered errors in soft- or hardware are exploited more rapidly for fresh attacks [9].

Fighting DDoS attacks requires in-depth packet inspection to identify malicious streams in the flood of traffic. With today's commercial high-performance routers, however, payload analysis is not possible, usually. Or if it is, the functionality is coded either in firmware or hard-wired in the box. Attack schemes vary a lot over time. In addition, the period becomes shorter between the first detection of an exploit and the widespread launch of the attack. So, it is crucial that large-scale DDoS attacks are defeated on routers as close to the core of the Internet as possible. Specific Anti-DDoS components must be installed, configured and removed on request. For obvious reasons, the deployment of the specific detection and countermeasure components must not interfere with other services. Further, they must be able to tackle the problem of known as well as unknown attacks semi-automatically according to predefined policies.

We propose PromethOS NP [4,5,6] as the dynamically code-extensible router platform for the envisioned Anti-DDoS service. It provides the abstractions required for node-internal communication among service components by which services are allowed to span arbitrary processors. Further, it provides the mechanisms to install, configure, instantiate and remove service components on any code-extensible processor of the processor hierarchy. Hence, the goal in this paper is to briefly introduce the architecture of an Internet backbone Anti-DDoS service for our powerful PromethOS NP router architecture.

2 Router Platform

Fig. 1 depicts the architecture of a PromethOS NP node using a three-tier processor hierarchy[1] and a node control layer.

On all tiers, PromethOS NP provides dynamically code-extensible processing environments (PEs). PromethOS NP creates a hierarchical execution environment (EE)

[1] The current implementation creates a three-tier hierarchical router platform for nodes that are built of host processor and NP-blades. NP-blades consist of a control processor with a set of packet processors (Appl. Ref. Board [7] for the IBM PowerNP 4GS3 [2]).

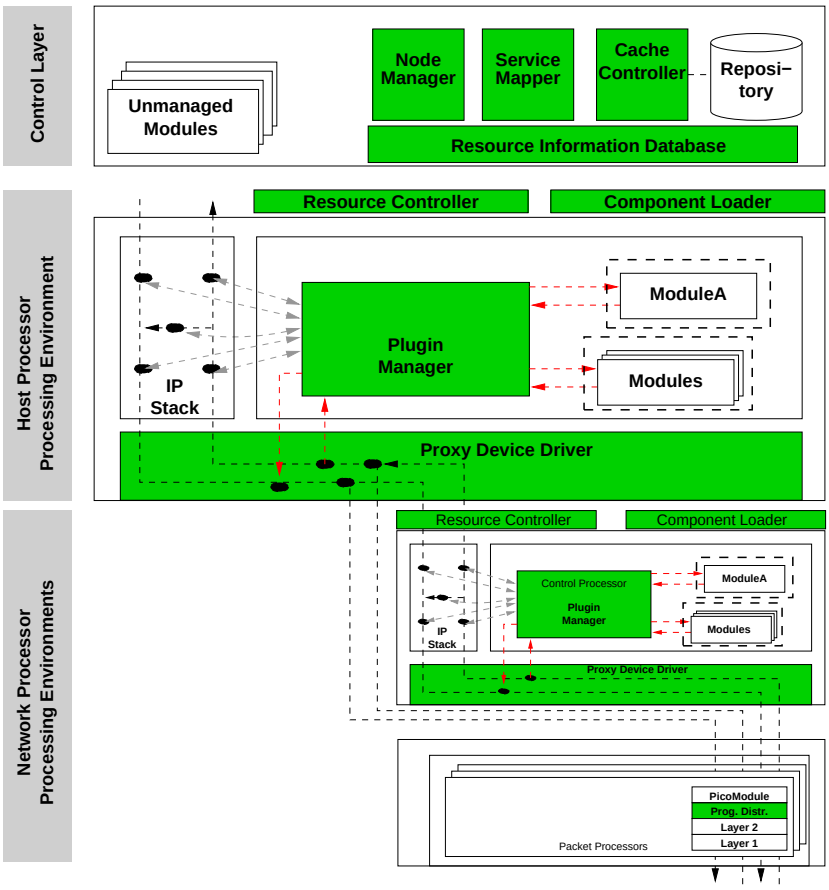

Figure 1: PromethOS NP Node Architecture

by that an interface to the hierarchical EE is provided only via the control layer. Internally, PromethOS NP manages two different types of code-extensible PEs, in which service components can be installed and instantiated. On the general purpose processor cores, the PE is implemented as an extended PromethOS EE [3] (cf. Host Processor Processing Environment in Fig. 1). This PE provides a binary compatible interface to the PromethOS EE. In contrast to the PromethOS EE, that runs on a single processor node only, the other PE (cf. Network Processor Processing Environments in Fig. 1) is embedded in the hierarchical router platform and provides the abstractions to build a service of distributed service components residing in other PEs. On the packet processors (PPs), a PE is instantiated that provides the mechanisms to install and execute service components without stopping the PP.

The control layer contains components which are responsible for the whole node. The *Node Manager* provides the interface to create a service at node run-time and instructs the other components on the node to act according to its decision. The *Service Mapper* creates the required map specification that provides the information to install and instantiate service components on specific processors such that, first, a service can be created and, second, the resources available are not overbooked. It instructs the PE specific *Component Loaders* to load, instantiate, configure and unload service

components.

3 Service Architecture

Fig. 2 visualizes our Anti-DDoS service architecture in a particular configuration that consists of a basis *service infrastructure* and an attack specific *Service Handler*. While the Service Handler must make the required functionalities available to detect and counteract DDoS attacks, the other components are generic in the sense that they provide the fundamental service architecture. Since the path via the Service Handler creates the needed countermeasure functionality, we refer to this path as the *service path*. Irrespective of the functionality provided, for the PromethOS NP router platform service components are black boxes. As such not only the service path but also the service infrastructure are built of service components that provide the appropriate functionalities. The service specification is used by the Node Manager that triggers the installation and instantiation of the service as mentioned above. The service logic, however, is service specific. As such, the service logic may contain mechanisms to request the installation or removal of service components depending on service-internal policies. Due to this autonomous, policy based service-internal management, our service architecture provides the basis of a node-local *autonomous service*.

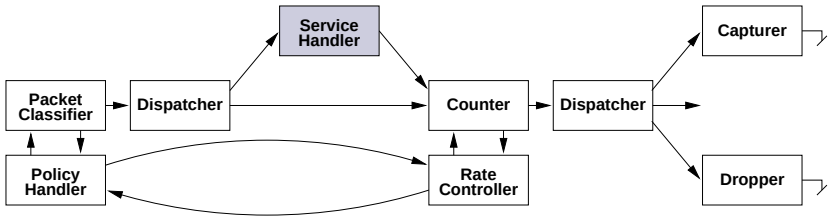

Figure 2: Anti-DDoS Service Architecture

4 Conclusion

The PromethOS NP router platform provides the execution environment to dynamically install and run services on hierarchical network nodes that are built of network and host processors. Services are composed of service components. Components of a service reside in processing environments that provide the required functionality on processing elements with sufficient resources. Channels that inter-connect service components abstract from the underlying communication-complexity. By these mechanisms, a service is created as an arbitrary graph of service components. The service graph is mapped on the processor hierarchy by the control layer to exploit the available resources best.

Our service architecture creates the basis infrastructure for the deployment of Anti-DDoS service components on demand. Based on policy mechanisms, the policy handler installs appropriate countermeasures, and reconfigures the architecture accordingly. Functionality deployed is supposed to detect specific DDoS attacks and allow for appropriate counteraction. We envision slowdown and intelligently blocking or capturing packet filters as suitable countermeasures. They are installed on demand in the specific processing environments as service components.

References

[1] T. Dübendorfer, A. Wagner, and B. Plattner. An Economic Damage Model for Large-Scale Internet Attacks. In *13th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WET ICE 2004); Workshop on Enterprise Security, Modena, Italy*, 2004.

[2] IBM Corp. IBM PowerNP NP4GS3 databook. `http://www.ibm.com`, 2002.

[3] R. Keller, L. Ruf, A. Guindehi, and B. Plattner. PromethOS: A Dynamically Extensible Router Architecture Supporting Explicit Routing. In *Proc. of 4th Annual Int. Working Conf. on Active Networking (IWAN), Zürich, Switzerland*, number 2546 in Lecture Notes in Computer Science. Springer Verlag, Heidelberg, Dec. 2002.

[4] L. Ruf, R. Keller, and B. Plattner. A Scalable High-performance Router Platform Supporting Dynamic Service Extensibility On Network and Host Processors. In *Proc. of 2004 ACS/IEEE Int. Conf. on Pervasive Services (ICPS'2004), Beirut, Lebanon*. IEEE, Jul. 2004.

[5] L. Ruf, R. Pletka, P. Erni, P. Droz, and B. Plattner. Towards High-performance Active Networking. In *Proc. of 5th Annual Int. Working Conf. on Active Networking (IWAN), Kyoto, Japan*, number 2982 in Lecture Notes in Computer Science. Springer Verlag, Heidelberg, Dec. 2003.

[6] L. Ruf, A. Wagner, K. Farkas, and B. Plattner. A Detection And Filter System for Use Against Large-Scale DDoS Attacks In the Internet-Backbone. In *Proc. of 6th Annual Int. Working Conf. on Active Networking (IWAN), Lawrence, Kansas*, Lecture Notes in Computer Science. Springer Verlag, Heidelberg, Oct. 2004.

[7] Silicon Software System. Application reference board for the IBM PowerNP NP4GS3 network processor user manual.

[8] The Economist. E-commerce takes off. *The Economist*, 371(8375):9, May 2004.

[9] A. Wagner, T. Dübendorfer, B. Plattner, and R. Hiestand. Experiences with Worm Propagation Simulations. In *ACM Workshop on Rapid Malcode (WORM)*, 2003.