Diss. ETH No. 15372

SELF-CONFIGURING SERVICES FOR EXTENSIBLE NETWORKS – A ROUTING-INTEGRATED APPROACH

A dissertation submitted to the

SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZÜRICH

for the degree of
Doctor of Technical Sciences

presented by

RALPH M. KELLER

Dipl. Informatik-Ing. ETH
born March 14, 1971
citizen of Rorbas (ZH)

accepted on the recommendation of
Prof. Dr. Bernhard Plattner, examiner
Dr. James P. G. Sterbenz, co-examiner
Prof. Dr. Jonathan S. Turner, co-examiner

2004

Abstract

During the last decades the original Internet architecture evolved dramatically with new functionality being added to the network layer to support a wide range of emerging applications. Network services such as firewalls, congestion control, media gateways, and traffic engineering all require a network that not only forwards packets based on the destination address, but also performs packet processing on nodes interior to the network. In an effort to support such application-specific packet handling requirements, router manufacturers have started to embed programmable elements into routers for providing network service functionality in a more flexible way. However, deploying new services in an existing network is usually a manual and time consuming process requiring the installation of code on multiple routers distributed all over the network. Given the complexity of how services can be composed, the only feasible approach is to automate this process. For this reason, it is crucial to have a suitable service infrastructure built on top of the raw processing capabilities to enable programmability of each node.

This thesis presents a service framework that allows router resources to be programmed and coordinated in such a way that the underlying network provides the anticipated services on behalf of applications. We have developed the ANCS (Active Network Control Software), which can be seen as an additional control layer in an active network environment that offers a generic service abstraction and automates the configuration of processing resources to form network services. Our system accepts processing demands from applications, maps their processing requirements onto the available network resources, and configures appropriate resources on network nodes.

In this thesis we focus on all the control mechanisms needed by such a service framework. Firstly, we propose active pipes as a high-level programming interface to the active network. An active pipe models the processing requirements as a sequence of processing steps performed on a data flow, without the application having to know about the underlying topology and location of processing resources. A processing step can be either mandatory or optional, meaning that the execution can depend on the state of the network. Each processing step can have multiple attribute constraints refining the location of processing. Secondly, we describe a resource discovery protocol for the dissemination of information about processing resources. Our approach is based on extending a link-state routing protocol such as OSPF and distributing the processing capabilities as opaque link-state advertisements. Thirdly, we describe an algorithm that maps the processing requirements expressed as an active pipe onto the physically available network resources. This mapping algorithm solves the problem of finding the optimal location of all specified mandatory and optional processing steps including a path transiting the sites, while minimizing network costs. Since our solution optimizes for both link and processing costs, paths can become non-simple, meaning that a given node can be visited repeatedly. The runtime complexity of the algorithm is polynomial, and thus scales to large networks. Fourthly, we have designed a signaling mechanism for the installation of processing code on selected nodes along with the establishment of explicit forwarding state such that traffic gets routed through these nodes as determined by the mapping algorithm.

We have implemented our service framework along with all the necessary control operations and protocols on top of our modular and extensible PromethOS router architecture. We have demonstrated the viability of our approach in a realistic environment using two applications that benefit from network-interior packet processing. On behalf of a video distribution application, we deploy application-specific congestion control modules before congested links. Using our novel video scaling scheme, we show that the perceived video quality improves significantly compared to traditional best-effort packet queuing. In a second application, we implement a security gateway that performs data encryption on routers in a way completely transparent to end systems. Furthermore, our performance evaluation demonstrates that services can be established efficiently with minimal overhead.

Kurzfassung

In den letzten Jahren hat sich die ursprüngliche Internet-Architektur stark entwickelt, indem zur Unterstützung neuartiger Applikationen neue Funktionalität zur Netzwerkschicht hinzugefügt wurde. Netzwerkdienste wie Zugangsschutzsysteme, Verkehrsstaukontrollen, Konversion von multimedialen Daten, sowie Datenverkehrssteuerungen erfordern ein Netzwerk, welches Datenpakete nicht nur anhand der Zieladresse weiterleitet, sondern auch Paketinhalte untersucht oder gar modifiziert. Aufgrund dieser anwendungsspezifischen Paketverarbeitungsanforderungen haben Hersteller begonnen, programmierbare Komponenten direkt in Router einzubauen. Die Konfiguration dieser Routerkomponenten mittels applikationsspezifischem Programmcode ist jedoch ein manueller und somit zeitintensiver Prozess, da Code verteilt auf verschiedenen Knoten installiert werden muss. Aufgrund der Komplexität wie sich Netzwerkdienste zusammensetzen lassen sollte dieser Prozess möglichst automatisiert werden. Dazu wird eine Infrastruktur benötigt, welche das Erstellen und Betreiben von Netzwerkdiensten grundlegend vereinfacht.

In der vorliegenden Dissertation wird ein Framework zur Programmierung und Koordination von Routerressourcen vorgestellt, damit das darunter liegende Netzwerk entsprechende Dienste für Applikationen erbringen kann. Im Rahmen dieser These wurde das ANCS (Active Network Control Software) entwickelt, welches eine zusätzliche Kontrollschicht zur Erstellung von Diensten innerhalb eines aktiven Netzwerkes darstellt. Das vorgeschlagene System nimmt Anfragen zur Erstellung von Netzwerkdiensten entgegen, bildet die gestellten Verarbeitungsanforderungen auf die verfügbaren

Netzwerkressourcen ab und konfiguriert die entsprechenden Router, damit schliesslich das Netzwerk den gewünschten Dienst für die Applikation erbringt.

In dieser These wird auf sämtliche Kontrollmechanismen eingegangen, welche für ein solches System erforderlich sind. Zuerst wird das “Active Pipe”-Modell vorgeschlagen, mittels welchem sich die Verarbeitungsanforderungen in abstrakter Form beschreiben lassen. Eine Active Pipe definiert eine Sequenz von Verarbeitungsschritten innerhalb des Netzwerkes, wobei die Applikation von der Netzwerktopologie und Lokalität von programmierbaren Knoten abstrahiert. Verarbeitungsschritte können entweder erforderlich oder optional sein, d.h. falls eine bestimmte Bedingung zutrifft. Auf welchen aktiven Knoten die Verarbeitungsschritte ausgeführt werden dürfen kann durch Nebenbedingungen spezifiziert werden. Zweitens wird ein Protokoll zur Lokalisierung und Eigenschaftserkennung von aktiven Knoten innerhalb des Netzwerkes vorgeschlagen. Unser Ansatz basiert auf der Erweiterung von Routingprotokollen wie OSPF, damit diese Protokolle auch Informationen über verfügbare Routerressourcen verteilen. Drittens wird ein Algorithmus beschrieben, welcher die Paketverarbeitungsanforderungen auf die verfügbaren physischen Ressourcen des Netzwerkes abbildet. Dieser Algorithmus bestimmt sowohl die optimalen Knoten für die Verarbeitungsschritte also auch einen verbindenden Pfad, wobei die Kosten der dazu notwendigen Netzwerkressourcen minimiert werden. Als Resultat kann ein Pfad bestimmte Knoten mehrmals besuchen. Die Laufzeitkomplexität des Algorithmus ist polynomial und skaliert daher auch für grosse Netze. Viertens wird ein Signalisierungsprotokoll vorgestellt, welches ausführbaren Paketverarbeitungscode in Routern installiert und Pfade aufsetzt, damit Applikationsdaten entsprechend der Pfadbestimmung durch das Netzwerk befördert werden.

Sämtliche Kontrolloperationen und Protokolle wurden auf unserer erweiterbaren PromethOS Routerarchitektur implementiert. Zur Demonstration wurde eine Multimedia-Applikation entwickelt, welche Verkehrsstaukontrollen innerhalb des Netzes installiert. Eine Evaluation veranschaulicht, wie sich dabei die Qualität des empfangenen Videos signifikant verbessern lässt. Ebenfalls wurde ein Security-Gateway zur transparenten Verschlüsselung von Applikationsdaten implementiert. Zudem zeigt sich, dass sich mittels unseres Systems Netzwerkdienste effizient in kurzer Zeit aufsetzen lassen.