

Diss. ETH No. 14828

# Design of topology-aware networked applications

A dissertation submitted to the  
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH  
(ETH ZÜRICH)

for the degree of  
Doctor of Technical Sciences

presented by  
Roger P. Karrer  
Dipl. Informatik-Ing. ETH  
born May 17, 1969  
citizen of Winterthur, Switzerland

accepted on the recommendation of  
Prof. Dr. Thomas R. Gross, examiner  
Prof. Dr. Roger P. Wattenhofer, co-examiner

2002

# Abstract

The Internet is designed along several simple principles: best-effort delivery, separation between end systems and networks, and peer-to-peer (P2P) connectivity. This simplicity has greatly contributed to the growth of the Internet to its current dimensions, but is no longer suited to efficiently support the communication requirements of today's and tomorrow's applications, such as multicast, mobility, bandwidth-based routing or adaptation to address heterogeneity.

Our thesis is that these communication requirements can be addressed by a single application-layer communication architecture that integrates knowledge about the topology and the available resources in a network into the application context, and that this integration pays off. We thereby assume that nodes inside the network (proxies) exist on which applications can instantiate their code.

This dissertation first describes the architecture, a three-layered *topology-aware framework* we call Octopus. This application-layer framework provides abstractions for services which can be deployed inside the network, algorithms and tools to locate, evaluate and select proxies, as well as mechanisms to steer an ongoing transmission. The framework fixes the process of topology-aware communication while allowing applications express their preferences by customizing the communication abstractions.

The concept of topology-awareness implies that information about the available proxies must be available. At the lowest layer, this dissertation describes a scalable and practicable application-centric solution to discover available proxies. These proxies are organized in a graph to form the network topology for a topology-aware application. The topology discovery is combined with the measurement of the available resources along this topology.

The middle layer contains abstractions to evaluate the topology graph and select a proxy to instantiate application-specific code. The evaluation can be made with application-specific preferences, e.g., to find the path through the graph with the smallest latency or the largest bandwidth. Similarly, the selection can be customized, e.g., to place code for adaptation or for multicasting.

Finally, at the topmost layer, we evaluate our solution with two topology-aware applications. A collaborative application distributes data of different content and size to multiple participants. We can show that a topology-aware path selection can considerably

improve the delivery time or the delivery quality, based on the application preferences. Second, we integrate an existing MPEG-1 application into Octopus to deploy different communication forms. Octopus supports the dynamic creation of a multicast tree or the dynamic change from an existing connection to an alternative connection.

We conclude that the breaking of the end-to-end concept and the layering concept of the Internet allows an application to opportunistically use the available resources in a way that better reflect the application requirements. The topology-aware framework is thereby well suited to deal with the increased complexity of taking topology- and performance information about the network into account.

# Kurzfassung

Der Aufbau des Internets basiert auf einigen einfachen Prinzipien: die Daten werden ohne Garantien übertragen, die Endsysteme und das Netzwerk sind klar voneinander getrennt, und Verbindungen verbinden genau 2 Endsysteme. Die Einfachheit dieser Prinzipien hat viel dazu beigetragen, dass das Internet die heutigen Ausmasse erreicht hat. Nur: ebendiese Einfachheit verhindert eine effiziente Unterstützung von Kommunikationsanforderungen von heutigen und zukünftigen Anwendungen, wie beispielsweise Multicast, Mobility, Datenübertragung entlang von Pfaden mit hoher Bandbreite, oder eine durch die Heterogenität des Netzwerkes bedingte Anpassungen der Daten.

Die These, die dieser Dissertation zu Grunde liegt, ist, dass diese Anforderungen durch eine einzige Kommunikationsarchitektur auf Applikationsstufe unterstützt werden können, bei welcher Information über die Topologie und die verfügbaren Ressourcen eines Netzwerkes in den Anwendungskontext integriert werden; ausserdem bietet diese Architektur der Anwendungen gewisse Vorteile. Wir gehen in unserem Modell davon aus, dass das Netzwerk spezielle Knoten anbietet (Proxies), auf denen die Anwendungscode installieren können.

Diese Dissertation beschreibt zuerst die Kommunikationsarchitektur. Sie besteht aus einem *topologie-sensitiven* Framework, das wir Octopus nennen. Das Framework, das ausschliesslich auf der Applikationsschicht arbeitet, bietet folgende Komponenten an: Abstraktionen für Dienste, die auf den Proxies instanziiert werden können; Algorithmen und Werkzeuge, um Proxies in einem Netzwerk zu lokalisieren, zu evaluieren und den besten Proxy auszuwählen; schliesslich Mechanismen, um eine laufende Datenübertragung zu steuern. Das Framework definiert und fixiert dabei den Ablauf einer *topologie-sensitiven* Anwendung während die Anwendungen ihre Präferenzen durch eine Erweiterung der Abstraktionen des Frameworks ausdrücken können.

Das Framework besteht aus drei Schichten. Auf der untersten Schicht beschreibt diese Dissertation eine sklarierbare und praktische Lösung, um verfügbare Proxies in einem Netzwerk zu lokalisieren. Die Topologie, die sich aus der Anordnung der gefundenen Proxies ergeben, wird zusätzlich mit Messungen der verfügbaren Ressourcen angereichert.

Die mittlere Schicht bietet Abstraktionen, mit denen die Topologie evaluiert und der beste Proxy für eine bestimmte Anwendung ausgewählt werden kann. Eine Anwendung kann dabei beispielsweise bestimmen, ob sie der beste Pfad durch die Topologie auf-

grund der kleinsten Latenz oder der grössten Bandbreite definiert ist. Ebenso bestimmt die Anwendung, welcher Ort der beste ist, um einen bestimmten Dienst einzurichten, beispielsweise für Multicast oder Datenanpassung.

Der oberste Layer schliesslich präsentiert zwei verschiedene Anwendungen, mit denen die These dieser Dissertation evaluiert wird. In einer verteilten Anwendung schicken sich die Teilnehmer Daten verschiedener Art und Grösse zu. Das topologie-sensitive Framework unterstützt dabei die Übertragung so, dass automatisch der beste Pfad für einen bestimmten Datentyp ausgewählt wird. Dabei kann die Qualität der Übertragung oder die Übertragungszeit deutlich reduziert werden. Die zweite Anwendung, die MPEG-1 verwendet, zeigt, wie verschiedene Dienste die Qualität des Videostroms verbessern können, und wie Octopus die Anwendung beim Platzieren von Multicastdiensten oder beim Wechsel von einer bestehenden zu einer alternativen Verbindung unterstützt.

Die Schlussfolgerung dieser Dissertation ist, dass das Aufgeben der Übertragung von einem Endsystem zum andern oder das Aufweichen der strikten Schichtung in Netzwerken eine bessere Nutzung der zur Verfügung stehenden Ressourcen führt, welche den Anforderungen der verschiedenen Anwendungen besser entspricht. Das topologie-sensitive Framework eignet sich, um die erhöhte Komplexität, die durch die Aufgabe der Prinzipien entsteht, aufzufangen und damit die Anwendungen effizient zu unterstützen.