1. Introduction

This introduction provides motivation and relevance of this thesis work, the problem statement, a description of the approach undertaken to solve such problem, and contributions made.
1.1 Motivation & Relevance

Society demands next generation Internet to increase its capacity, quality and number of services. Due to advances in microelectronics and telecommunications Internet capacity doubles every few months. Advances in networking and distributed systems research are improving Internet quality of service. However the Internet network is a fixed system where introduction of new services is a difficult task except on individual end-hosts. Introduction of new services on the Internet is a laborious, time-consuming task. "The introduction of new services into existing networks is usually a manual, time consuming and costly process", "there is an increasing demand to add new services to networks to match new application needs" [Cam99], "vendors are hesitant to support service before they gain user acceptance, yet the utility of network services is dependant on their widespread availability" [Ten97]; "much human coordination is necessary to achieve functionality enhancements in large-scale deployments" [Gov98]. These citations expose the relevance of facilitating service introduction.

Application networks are overlays, or virtual networks on top of another network. Some overlays provide network services while others provide application services, at the network layer or at application layer. Network layer overlays are composed of routing nodes connected by layer three tunnels: they have equal properties to a real network but several of them coexist on the same infrastructure sharing underlying resources with more or less isolation. They provide network services not provided by the underlying network: multicast, secure communications. Layer seven overlays are composed of servers connected by application layer tunnels. Layer seven overlays providing network services are known as peer-to-peer overlays, since they use Internet client nodes as network nodes. They provide network services such as multicast or active networking. Application networks are layer seven
overlays providing application services. Application networks improve Internet services quality by providing applications with desired properties such as server proximity, high availability, high load capacity, etc. Application networks are a set of cooperating application servers, distributed throughout the Internet, connected and coordinated to provide application services with low latency, load balancing and/or good consistency. Examples of application networks are content distribution networks such as Akamai [AKA01] or file discovery peer-to-peer networks such as Gnutella [Ora01]. Due to coordination among application network servers, client requests are provided with good quality of service. Requests from location “a” are provided by a nearer server “k” instead of farther server “g”; also server “k” load will not increase beyond a threshold, causing next request to be delayed, instead it will be provided by less loaded server “h”.

![Fig. 1.1. Application network on top of Internet](image)

Newly created Internet application services are constantly appearing [Ami98]. However currently introducing new services is a manual process that involves several operators from different organization resulting in a slow, error-prone operation, especially for application network services. Setting up a new application network involves manually
installing a set of application servers and manually configuring connections among them. It is a laborious, time-consuming and costly task. Consider the case of a web caching service [Wes98]. Deploying manually a proxy caching service requires as shown in figure 1.2: first, discovery of hosts and network topology map, second, layout application network in such a map to satisfy demand, third, coordinated installation, configuration and start-up of servers from a management console, fourth, configure redirector service from a management console, and fifth, monitor service performance and adjust from a management console. The cost of putting operational such web caching service can exceed the cost of hardware and software equipment it requires. If we talk about multi-administration services, cost increases are high because coordination among different organizations personnel is a big challenge.

1. Discovery of hosts and network topology.
2. Lay out application network to satisfy demand.
3. Coordinated installation, configuration and start-up of servers.
4. Update name / redirector servers.
5. Monitor service performance and adjust.

Fig. 1.2. Web caching manual service activation steps

Today application network are create on the Internet either by resource providers installing servers on their machines with little or no coordination; or those who can afford it, provision a proprietary servers pool where they activate a service at every node through a centralized management stations. In the first case service providers are not able to specify which service level they desired to provide clients, whereas the management stations model is not scalable and cost-effective. Introduction of application network services by dynamic deployment are needed to allow deployment of application layer services with a predefined service level in a cost-effective way.
Furthermore, Internet application services have demands with spatial and temporal variations [Ren02]. Today modifying application networks to adapt to variable service demands is a manual process resulting in a slow, error-prone operation. Modifying a new application network involves adding and removing servers, migrating servers to locations where new demand arises and cancelling and creating connections among servers, figure 1.3. That is dynamic deployment of application networks permits to adjust application services to spatial and demand variations.

![Application spatial demand variations](image)

*Fig. 1.3. Application spatial demand variations*
1.2. Problem Statement

This thesis work studies the creation of application layer overlay networks in a large-scale dynamic environment, with multiple service creators and resource providers and temporal and spatial service demand variations, such as the Internet. The research question this thesis plans to solve is:

-How to facilitate introduction of application networks in the Internet in a controlled way.
1.3. Approach

The goal of this work is to come up with a solution to the problem stated previously at section 1.2. The solution requires in first place an Internet service programmable infrastructure. An Internet service programmable infrastructure enables application servers to be activated at any location on the Internet. Such infrastructure is composed of programmable nodes distributed throughout the Internet where service providers can remotely activate and stop application servers and connect and disconnect server instances. Programmable infrastructure enables an open, efficient and secure activation of new services, promoting development of new services, and flooding the Internet with services for the benefit of end users. It is analogue to the way UNIX open computer programming platform [Ker84] has promoted development of thousand of end-host applications. A system that facilitates service introduction will make it easy for service creator to provide their service to clients, thereby promoting development of new and innovative services. Besides it will benefit end users by providing them with a wide range of services to choose.

In second place, application networks must be deployed " to spread out or arrange for effective action" [BRI02], instead of being uncoordinatedly activated. Though activating a service in a single node is not difficult, when there are thousands of nodes where to activate a service and/or thousands or service creators interested in deploying a service, controlled service introduction is almost impossible. It is required a framework for application network deployment, so that an application services are activated, connected and coordinated in a number of nodes composing an application networks. As a result application servers are coordinated and placed at appropriate locations for least resource utilizations and/or other application goal.
In third place a solution that permits application network deployment in the Internet environment is pursued. The approach undertaken is to use a resource-initiated allocation and multicast based deployment mechanisms because in the Internet resources are scarce and resource contention among different deployers is a severe problem that can arise with some allocation mechanisms. And because resource-initiated multicast based deployment does not present scalability problems due to resource discovery and monitoring mechanisms.
1.4. Thesis Statement

The thesis presented in this dissertation is:

“A framework for application network deployment in an Internet service programmable infrastructure allows creating application networks which are provisioned dynamically, and coordinated for service control. A resource-initiated allocation multicast-based deployment mechanism is more appropriate for the Internet than service-initiated allocation mechanisms”
1.5. Contributions

This thesis makes two contributions: first, it is proposed a model and framework for application network deployment called Xweb [Ard03a], which has been implemented and evaluated. Application network deployment model takes into account existing resources and service specifications to create a plan on how a service has to be deployed. This plan contains which resources have to be allocated and how services have to be composed. Building blocks of the Xweb application network deployment framework are: programmable Internet service infrastructure, resource discovery, service specification, resource allocation and service composition. Xweb programmable infrastructure consists of a number of nodes, where multiple new application servers can be hosted. These nodes publish their resource availability and characteristics. Deployment managers implement a proactive resource discovery and monitoring functionality, which seeks for new resources by sending resource availability requests to a global multicast channel. Providers of newly created services input service specifications to deployment manager. Deployment managers perform a resource mapping of such specifications to resource available, create a deployment plan which contain commands on where, how and when to deploy the application network, and finally carry it out. Deployment managers contact resource agents at programmable Internet service infrastructure, and command to allocate resources, to distribute service code, to bind application to resources and to compose and activate applications. Resource agents perform such operations onto programmable infrastructure nodes. Service composition involves binding service programs to allocated resources and coordinating different service instances. Coordination requires set up communication among different instances and configuring coordination rules. Experiments were carried out in such prototype to validate this contribution.
Second, it is proposed a new deployment mechanism called multicast injection [Ard01]. It implements a resource-initiated allocation model that performs better in the Internet that service-initiated allocation. Multicast injection mechanisms consist of three main phases: first, application networks specifications are multicast in a global channel listened by all resource agents; second, resource agents decide whether to allocate resource or not, and send an allocation proposal to a multicast group exclusive for that application, resource agents on receiving that allocation proposal confirm their proposal or not by a mechanism called multicast damping; finally resource agents allocate resource and configure a topology among them. Evaluation of multicast injection deployment mechanisms has been completed through simulation.
1.7. Thesis Organization

Chapt 2: Related Work and State of Art
Chapt 3: Model and Framework for Application Network Deployment
Chapt 4: Xweb Application Network Deployment Implementation and Experiments
Chapt 5: Resource-initiated Allocation Multicast Injection Mechanisms for App Net Deployment in the Internet
Chapt 6: Simulated evaluation of Multicast Injection Resource-initiated Allocation Deployment.
Chapt 7: Conclusions and future work.