Technical Report 2002-01-01
Internetworking and Media Communications Research Laboratories
Department of Math & Computer Science, Kent State University
http://medianet.kent.edu/technicalreports.html

CS 6/79995 ST: Advances in Internet based Systems and Applications
Spring 2001

A Survey of Ideas in Programmable Networks

Xuebin Xu
email: xxu1@kent.edu, homepage: www.personal.kent.edu/~xxu1

Prepared for Prof. Javed I. Khan
Department of Computer Science, Kent State University
Date: January 2002

Abstract: In this paper, we present a hot research area in network, active and programmable network. Programmability added to traditional network makes network more complex, flexible and cost-efficient. An array of projects is reviewed, in order to interpret this novel networking model and its state-of-the-art. We first introduce the basics of programmable network. Also, different architectures and specific methods for programmable network are examined. At last, we conclude this paper by providing recent development and identifying the challenges in programmable networking.

Keywords: active network, programmable network, distributed operating system, router, capsule

Other Survey's on Internetwork-based Applications
Back to Javed I. Khan's Home Page

Table of Contents:

Introduction

What is programmable network
The general methods to make network programmable
Why active network?

How to build programmable network? - Programmable Networks Architectures

- active nodes approach
- active packets approach
- hybrid approach

Active Network vs. distributed operating system

- congestion control
- multicasting
- cashing

Open Problems and Research Directions

References

- Research Papers for More Information on This Topic
- Research Groups
- Other Relavant Links

Scope of Survey

Introduction:

What is programmable network?

The techniques and protocols used by networks today were developed more than 20 years ago. Intermediate nodes (routers) between two end computers are “passive”. They just pass the information of packets to the next computer on the route and don’t have any ability to process the data transported in packets, i.e. routers only decide the route of packets and don’t change the content. Routing algorithms only work to IP level in OSI model, not application level. With the increase of network popularity, people hope more applications from network. But it is difficult and cost-inefficient to deploy new applications in the traditional network. Internet service providers (ISP) must wait service vendors to implement the whole application and produce the special hardware before the new application can be provided to normal users. Application users cannot change the service once the application is deployed and make new applications of their own. It is critical for ISPs to react to the market needs as quickly as possible. ISPs hope there is a kind of more flexible and simplified means to deploy new applications, and network users hope to make use of the distributed computing environment of network. Programmable networking model is formed to meet this requirement. In programmable network model, intermediate nodes (routers) are not “passive” any more, but “active”. They can change the data in transported packets or the processing environment of routers. As illustrated in Fig.1, routers work to application level in OSI model. ISPs deploy some components on network nodes and network users can assemble these components to achieve some application they want, i.e. program the network [5, 11].

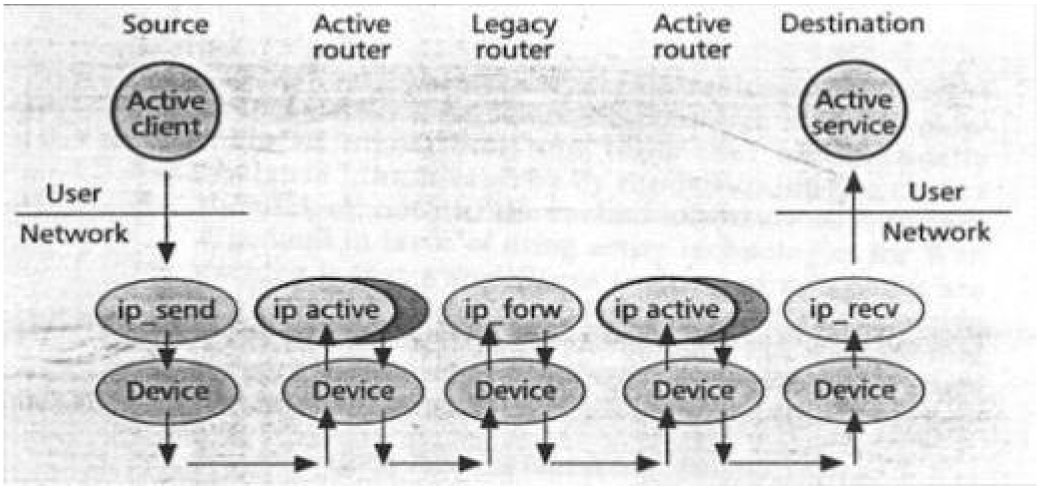

Fig 1. Application-specific processing within the nodes of an active network. *Courtesy to David L. Tennenhouse, etc.*

The general methods to make network programmable

The goal of programmable network is to simplify deployment of new services in the network and let network users create new applications by using existing service among nodes of network. A diversity of projects is working on this goal. In essence, there are two approaches to make network programmable: *open signaling* (OPENSIG) and *active networks* (AN), depending on whether programs and data are carried discretely or in an integrated fashion [1].

Programmable switches (a discrete approach): OPENSIG community thinks that new service is provided by virtual network. All kinds of primitives are installed on routers in the networks. Virtual network is decided during routing the packets. Nodes with needed primitives are visited by packets of a stream in a certain order. Active routers process the data in packets and then re-wrap processing result and pass it to next active node for further computation. Final results reach the destination of the route in the end. The structure of packets used in this approach changes very little comparing to traditional IP packets. The identifiers or references to primitives should be present in the header of packets.

Capsules (an integrated approach): AN community thinks that packets transports not only data, but also code itself. They call this kind of packets “capsule”. In active nodes, the content of capsules is evaluated to decide whether the code should be executed at run time. Capsules embody the “granuity of control”. It ranges from packet level to stream level. The first capsule of a stream might change the processing environment of this stream. This environment will be kept until the end capsule of the stream is processed.

The two approaches are two opposite extremities. Many other approaches fall in between these two. But they share the same goal to make network programmable. OPENSIG is a static method; AN is more dynamic and decide what code executed at run time. Of course, this is not absolute. OPENSIG can also decide the to-be-executed code-load dynamically.

Why active network?

Active network technique can accelerate network infrastructure innovation. Just like we talked above, it will take 10 years ordinarily from the appearance of prototype to large-scale deployment of that model at present. In order to speed up this process, active network model comes up naturally.

Active network enables some new applications [2]. Efficient merging and distribution of information for multi-user at multi-site are enabled in active network. A user can compose data from multi-site on network and multi-user can use data from multi-site according to the requirement of their own. Active network also let network protection can be done on a per-user or per-use basis. User can use a formal approach to specify safety rules. And active network can play a role in network management. It makes congestion control, multicasting and caching more dynamic and improve QoS of network.

How to build programmable network? - Programmable Networks Architectures

Different people have different viewpoints on programmable networks. Some people think traditional network, we are using nowadays, is simple and work stably and very well already. The used technology was developed more than 30 years ago. It is very mature. But other people think the idea about programmable network is appealing and novel. This idea is borrowed from the model on stand-alone computer. People can write a complicated program by only using basic set of instructions CPU understands. So if there are a bunch of primitives in network, people can build the applications of their own by using these “bricks”.

But how to build programmable networks? There are many classification methods from different point of view. I feel the classification criteria of Konstantinos Psounis more reasonable. He classified programmable networks architecture into 3 types: ***active nodes***, ***active packets***, and ***hybrid***.

Active nodes: In the active nodes method, packets hold transported data and identifiers or references to which primitive will be called to process their data. Functions or primitives reside in active routers. The packets are “active” just because they decide what functions will be executed when they reach routers and they provide needed parameters for the called functions. Actual executed code is not carried with packets. They are deployed in some nodes by ISPs as resources in network. ISPs provide the interfaces of these primitives to normal network users. This is the reason why this kind of architecture is called “active nodes”. Examples of active nodes architecture are talked about as following.

Cluster-based active network (CBAN) – this approach [3] builds active router architecture by cluster. The purpose of this architecture is to perform competitively as a conventional router, but also to scale network well with respect to required computational resources. The designation of active router architecture, as illustrated in Fig.2, is composed of a cluster of generic PCs. One PC in the cluster is configured as a normal IP router; while the rest PCs in cluster work as computing elements to provide computational resources for customization services. In this architecture, the network-specific functionality resides on the routing element, while the service-specific processing primitives reside on the computing elements. As seen in the illustration, the cluster routing element behaves as a normal IP router in traditional network. Besides this function, this PC also works as a controller in the active network. It captures unprocessed packets to decide whether they are needed to be processed by computing elements or not.

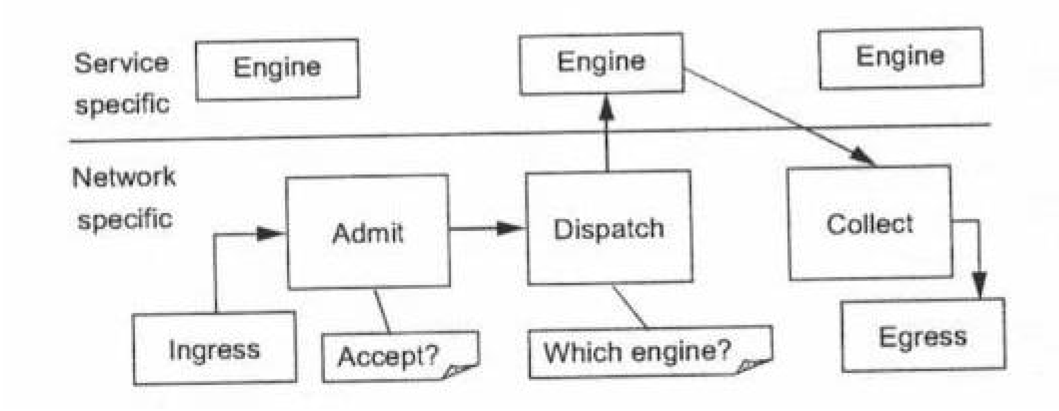

Fig.2. Functional overview. *Courtesy to Girish Welling, etc.*

If no any further computation is needed, the packets will be passed out of cluster and to next node in the network. If some computation is needed, packets are passed to the cluster dispatcher. The dispatcher will choose a computing element for the packets based on the information of packets and pass the packets to that computing element. After processing, packets are gathered in the routing element and then sent out on the appropriate interface according to their routing requirement. Computing elements can join or disconnect with cluster at any time. Its signature will be kept in the routing element, cluster controller, if the computing element is in. Also, this architecture simplifies to add new computing elements into network.

Routing element also functions as cluster manager. It checks the authentication of packets before dispatching them to computing elements for the safety reason. Also, it keeps the accounting of packets and resource use such as memory and cache usage for efficiency.

ANTS – ANTS[12] is an active network toolkit, in which new application can be deployed in network arbitrary by using mobile code technique. Code for one application is divided into several groups. These related groups are located among nodes in the active network. Some functions are "well-known" because every node in network has this code. However, some functions only reside in active nodes. When a capsule arrives, the node first decide which protocol the capsule belongs to and what functions are needed according to the information about protocol identifier and capsule type contained in capsule header. Then this node checks its cache of protocol to see if the needed code exist. If some portion of code is missed, the node will send code load request to its upstream neighbors and let the current capsule "sleep". Neighbors then check if they cache the requested code. If so, they will forward it there. And this code is cached at new site for the following capsule of current stream. Once the node gets the needed code, it wakes up the sleep capsule and process the data of it. If not, the sleep capsule will be discarded.

The code is executed in a restricted environment because of security reason. Capsules in this architecture only have limited access to shared resources. Active nodes play this role. They check the integrity of capsule and recover the system from error. Capsules have a field, called "Time To Live" (TTL). This field is decreased with the consumption of resource. The capsule will be discarded when the TTL field is zero.

Besides the above examples, Jit Biswas etc. [4] also extend ATM switches and circuit switches and deploy open signaling, control and management application on higher level multimedia network services.

Active Packets: In the active packets method, packets hold not only the data, but also code itself. Primitives don't reside in an active node any more. When packets arrive, active node will decompose the structure of packet according to network protocol and process data or change the node environment by using attached code in packets. Therefore, this architecture is named as “active packets” just because code is carried with packets, and active node run the code to fulfill a task. Programming language for code is an important issue for this architecture because network is composed of all kinds of platform and packets size is limited. A good programming language can decrease the weight of programs and improve the performance, safety and security of active network. Examples of active packet architecture are presented next step.

Smart packets – smart packets project [13] makes two decisions: first decision is about programming language. The programs contained in the smart packets must be self-contained. Thus, they don’t depend on the persistent state of a router. Also, programs must be packed into one packet. So program size cannot be too large. Normally, its size is within 1Kbyte long. Second decision is related to system security. Because incoming packets contain executable code, much care should be taken. The operating environment should take measures to prevent packets with bad code from being executed.

Either sprocket or spanner is used as programming language for smart packets. Sprocket is a higher-level language like C, while spanner is an assemble language. Programs in sprocket are compiled into spanner program, which in turn assemble to a platform-independent binary code. This binary code is packed into smart packet at last. The reason why other high-level languages, such as C, java, which seems to be a good choice, is not chosen is that the programs in these are normally too large to be put into one packet.

About security problem, active nodes in smart packet project keep checking the source of smart packets to see if a program comes from an authorized user, data integrity of smart packets and provide a limited running environment for incoming programs.

M0 architecture – M0 system [14] is composed of M0 nodes, between which messengers are exchanged. Messenger is similar to the capsule or the smart packet. Messenger code is programmed in M0 language. M0 language is a high-level language that inherits some data structures, operators and other characteristics from PostScript. When code is small, it is put in only one messenger. But if code cannot be hold in one messenger, the shared memory of M0 node under a chosen name will be used. This mechanism makes M0 architecture very flexible to deploy new protocol, however complex it is. For safety reason, one independent thread of control executes on behalf of one messenger. The memory used by these thread is separation and protected from one another.

Hybrid – Just like what we can see, in active node architecture code resides in active nodes, which provide protocols efficiently. However, in active packet architecture code is carried by smart packets efficiently. Hybrid architecture uses these two characteristics according to the size of code. Heavy-weight program stays on an active node and light-weight program is carried by active packets. Hybrid approach tries to find a tradeoff between active packet architecture and active node architecture. A typical example is SwitchWare architecture developed in University of Pennsylvania.

SwitchWare Architecture – SwitchWare [15] is constructed by using three layers: active packets, switchlet and active router infrastructure. Active packets layer uses active packet approach, while switchlet layer uses active node approach.

In SwitchWare, two level programming languages are used. Heavy-weight primitives are written by higher-level languages, such as C, Java etc. Light-weight code in active packets is written in PLAN (Programming Language for Active Network). It calls primitives named as switchlets, which are heavy-weight programs resides in active nodes. PLAN is a light-weight language, which helps to improve safety, security and efficiency for active network [8].

Active router infrastructure is the base of active packets layer and switchlet layer. It grants the security of SwitchWare as a whole. The authentication of accessing routines is done through this layer.

Active Network vs. distributed operating system

Active network and distributed operating system are two research areas associated with each other. Distributed operating system is a collection of software components based on a physical network computing environment. It is located between physical network components and normal network users and helps users to use network easily and efficiently. Just like operating system for stand-alone computers, distributed operating system also manages software and hardware resources. It makes users feel they were working on a stand-alone computer, although this is not the case. In future, distributed operating system can be built based on an active networking environment. **Congestion control**, **multicasting** and **caching** are 3 characteristics of whole network, which are related to QoS and suppose to be managed by distributed operating system. There are already a lot of active network applications on these issues [6]. Samrat etc. [9] think active networking techniques are consistent with and outperform end-to-end argument.

Congestion control: In the traditional network, when congestion happens at a certain node, that node will send a congestion message to the user from the congestion point. The user will decrease its used bandwidth. But it will take considerably long time for this message to travel across in network sometimes. Circumstances of network will change after this period. Either the congestion has already disappeared, or the user doesn't take any self-regulation before thing get worse, because the message did not get through. Active nodes can be used to control congestion in time. Buffers are used in this case at active nodes. Active nodes monitor the available bandwidth. If congestion happens, they will adjust the data flow speed. Congestion points don't need to send a message to notify users, but control congestion directly at active nodes on the fly.

Multicasting: A lot of network applications are related to audio, video data stream. These applications require multicasting technique. In traditional network, there are many problems for multicasting, such as NACK implosion, useless retransmission and run-time multicasting scale-up etc. These problems limit the QoS of the teleconference. Active network can be the solution. In MIT, ARM project (active reliable multicast) [16] claims that the problems are solved more efficiently. Active nodes are used to cache data for a specified time in ARM. So, NACK does not send back to sender only, but the first upstream active node on the route. This will cause NACK number decrease dramatically. Also, retransmission request can be fulfilled by active node, not by sender. And one active node re-sends the lost packet to the down-stream nodes, not all receivers of this application. The traffic decreases a lot because of these reasons. QoS improved by using active network technique.

Caching: In traditional network, cache close to clients is needed to decrease the network traffic and the time for retrieving the information. Normally, the caching nodes are those computers located near the edge of the network or at strategic points in network constructed with a hierarchical scheme. If we use active network technique, caching site is decided at demand on the fly. The traffic and retrieving time will decrease further and QoS will be better correspondingly.

Open problems and research directions

Researches on active network are very hot. From other papers, there are two schools of thought on how to make network programmable: OPENSIG and AN. Many annual workshops and conferences are held by these two communities and IEEE every year. We can imagine how many papers will be presented in these events. Therefore, it is sort of difficult to make a thorough list of research directions.

Also, active network research area and distributed operating system research area develop twisted. They share some common topics to a certain degree. So it is tough to get all open problems if we only examine papers from one field.

Anyway, I have a try to make a research list on active network as below:

- Distributed operating system problems and active network techniques develop together and have effects on each other;- A standard on active network is needed;

- Nowadays, the projects on active network are small-scale. It is still problematic to scale them up to the scale of world wide web;

- The changes to the organization of end system software are needed to provide the formal specification of network elements and behavior;

- A network programming language, which is easy to use and make programming as quickly as in end system, is still in the lab;

- To use per-packet programs to allow each packet to make decisions about how to route, congestion control;

- The adapted routing algorithms used for active network are needed [10].

References

Research Papers for More Information on This Topic.

1. Andrew T.Campbell, etc., “A Survey of Programmable Networks” Computer Communication Review, vol.29, no.2, pp.7-23, Apr.1999.

2. David L. Tennenhouse, etc., “A Survey of Active Network Research” IEEE Communications Magazine, pp.80-86, Jan. 1997.

3. Girish Welling, etc., “A Cluster-Based Active Router Architecture” IEEE Micro pp.16-25, Jan.-Feb. 2001.

4. Jit Biswas, etc., “Application Programming Interfaces for Networks (DRAFT WHITE PAPER)”, White Paper on Application Programming Interfaces for Networks, Working Group for IEEE P1520.

5. Jonathan M.Smith, “Programmable Networks: Selected Challenges in Computer”, Computer, 1999 IEEE.

6. Konstantinos Psounis, “Active Networks: Applications, Security, Safety, and Architectures” IEEE Communications Surveys, pp.2-16, First Quarter, 1999.

7. Randy Cieslak, etc., “The Programmable Network Prototyping System”, Computer, pp. 67-76, May 1989.

8. Scott Thilbault, etc., “Safe and Efficient Active Network Programming”, IRISA/INRIA Technical Report 1170, Feb 1998.

9. Samrat Bhattacharjee, etc., “Active Networking and the End-to-End Argument” Proceedings of the 1997 International Conference on Network Protocols (ICNP’97), pp.220-228.

10. Sumi Choi etc., “Configuring Sessions in Programmable Networks”, Proceedings of IEEE Infocom 2001, Active Network Technical Program.

11. Thomas M.Chen, etc., “Active and Programmable Networks”, guest editorial for Special Issue: Active and Programmable Networks, Network Interactive, May 1998.

12. D.J. Weltherall, etc. “ANTS: A Toolkit for Building &.Dynamically Deploying Network Protocols, proc. IEEE OPENARCH’98, April 1998.

13. Beverly Schwartz, etc., “Smart Packets for Active Networks”, BBN Technologies, Jan. 1998.

14. Albert Banchs, etc., “Multicasting Multimedia Streams with Active Networks”, ICSI technical report 97-050.

15. C.A. Guter, etc., “The SwitchWare Active Network Architecture”, IEEE Network, special issue on Active and Programmable Networks, May/June 1998, vol. 12, no3.

16. Li-wei H, etc., “Active Reliable Multicast”, Proc. IEEE INFOCOM’98, San Francisco, CA, 29 March-2 April 1998.

Research Groups

1. The research group of University of Pennsylvania works on SwitchWare project, which develops a programmable switch approach. In this project, PLAN is developed for active programming.

2. The research group of Massachusetts Institute of Technology works on a multicasting architecture, which uses active packets approach.

3. The research group of University of Kentucky and Georgia Tech. works on CANEs project to construct composite network services from components.

4. The research group of Columbia University works on NetScript project. This project provides a programming language and execution environment to script the processing of packet streams.

Other Relavant Links

- http://www.ieee-infocom.org/
- http://www.openarch.org/
- http://www.comsoc.org/

Scope

This survey is based on electronic search in IEEE conference sites, IEEE homepage, and their citations, "active network" used as keyword in search. Also, magazine of IEEE also searched, such as IEEE Communication Magazine.