

# Active Networks - A Survey

Sohail Munir, [munir@cis.ohio-state.edu](mailto:munir@cis.ohio-state.edu)

---

Active Networks, unlike the traditional networks are not just passive carrier of bits but instead provides the capability for the user to inject customized programs into the networks. The network nodes would interpret these programs and perform desired operation on the data flowing through the network. This paper introduces the concept of Active Networking and surveys the research in progress related to Active Networks.

---

[Other Reports on Recent Advances in Networking](#)

[Back to Raj Jain's Home Page](#)

---

## Table of Contents:

### Introduction

[Need for Active Networking](#)

[Goals of Active Networking](#)

[Impact of Active Networking](#)

### Active Networks - HOW TO

[Discrete Approach](#)

[Integrated Approach](#)

[What happens to Current Infrastructure](#)

[Active Networks Encapsulation Protocol \(ANEP\)](#)

[Proposed Packet Format](#)

[Active IP](#)

### Active Networks - Research

[ANTS -Active Networks Tool kit](#)

[SwitchWare](#)

[NetScript](#)

[PLAN - Programming Language for Active Networks](#)

## Summary

## References

---

# Introduction:

Active Networks is a relatively new concept, where a network is not just a passive carrier of bits but a more general computation model. Active Network may be simplistically viewed as a set of 'Active Nodes' that perform customized operations on the data flowing through them. Traditional data networks provide a transport mechanism to transfer bits from one end system to another, with a minimal amount of computation (e.g., header processing and signaling). In contrast to that active networks not only allows the network nodes to perform computations on the data but also allow their users to inject customized programs into the nodes of the network, that may modify, store or redirect the user data flowing through the network. These programmable networks open many new doors for possible applications that were unimaginable with traditional data networks. For example, there may be a video multicast session where at every node the video compression scheme is modified, based on the computation done by that node and depending on the network bandwidth available. The research community has realized the potential of Active Networking and a lot of work is underway at different research sites. This paper surveys the research related to Active Networking.

## Need for Active Networking:

Active Networking however, is not a totally new concept. Various approaches on solving current networking problems use a part of the Active Networking concept. Applications such a packet filtering in firewalls (also routers) where the filters in the firewall decide which packet should go through and which should be blocked. Other examples would be web proxies, multicast routers and video gateways etc. that perform user driven computations "within" the network. For example web proxies provide a user transparent service to serve and cache web pages. Nomadic Routers determines the means by which a host is connected to a network (e.g., a modem or a high speed network connection) and adapt to the conditions. For example it might do link compression and perform more file caching when connected through a modem. Also, it would enable lets say encryption when user is remotely connected. The list goes on and on. A lot of recent trends and developments in networking are a subset of the Active Network Architecture. For example VLANs in the PHY layer, Multiprotocol Router and transport technologies like RSVP and RTP and Application Layer Routing are all functions supposed to be provided by an Active Network Architecture. The task that lies ahead however is to replace all these ad-hoc approaches with a generic capability that would allow programming of these networks.

## Goals of Active Networking:

The goal for active networking is to have programmable open nodes, with the ability to deploy programs dynamically into node engines. The functions of the network nodes should not be the one standardized but the execution environment should be standardized. This is so, because with standardized functions, new innovative ideas get the back burner because of the need to protect investment.

## Impact of Active Networking:

The impact of an active networking architecture would be wide spread. It would accelerate infrastructure innovation. Network innovation currently is a slow and expensive process. A new technology might take over a decade to travel from prototype stage to large-scale deployment. The technology might even die because of this delay. Examples of current Internet services in this backlog are RSVP, Multicast Services, Security and Mobility Extensions and last but not least IPv6. Currently IP standardizes packet format, addressing schemes etc. and all routers or IP switches implement these, changing any thing in IP would require replacement of all this hardware. As opposed to this, active network nodes can execute different programs. It has an abstract layer supported on all devices, thus incorporating new innovations would mean downloading those new services into the infrastructure. Active Network can make the basic network services selectable on a per packet basis, eliminating the need for formal standardization. It can make the basic network service programmable, reducing the deployment time. Last but not least, an advanced, robust programming environment may be used, reducing the service development time. New applications like the merging and distribution of information; user aware network protection and active network management etc. would be feasible because of the flexibility of active networking.

[Back to Table of Contents](#)

---

## Active Networks - HOW TO:

There are two possible approaches to build active networks. A discrete or out-of-band approach and an integrated or in-band approach.

- **The Discrete Approach:** This may also be called a Programmable Node (Switch/Router) Approach. Here programs are injected into the programmable active node separately from the actual data packets that traverses through the network. 'User' would send the program to the network node (switch or router), where it would be stored and later executed when the data arrives at the node, processing that data. The data can have some information that would let the node decide how to handle it or what program to execute.

- **The Integrated Approach:** In an Integrated Approach, also termed as the Encapsulation Approach, the program is integrated into every packet of data send to the network. Each message or capsule as the literature calls it contains a program fragment that may or may not have some embedded data. When these capsules arrive at the active node, it interprets the programs and sends the embedded data depending on its interpretation of these programs. This concept is similar to Postscript code, where actual data is embedded in program fragments that the printer understands.

In this approach, each active node would have built-in a mechanism to load the encapsulated code, an execution environment to execute the code and a relatively permanent storage where capsules would retrieve or store information.

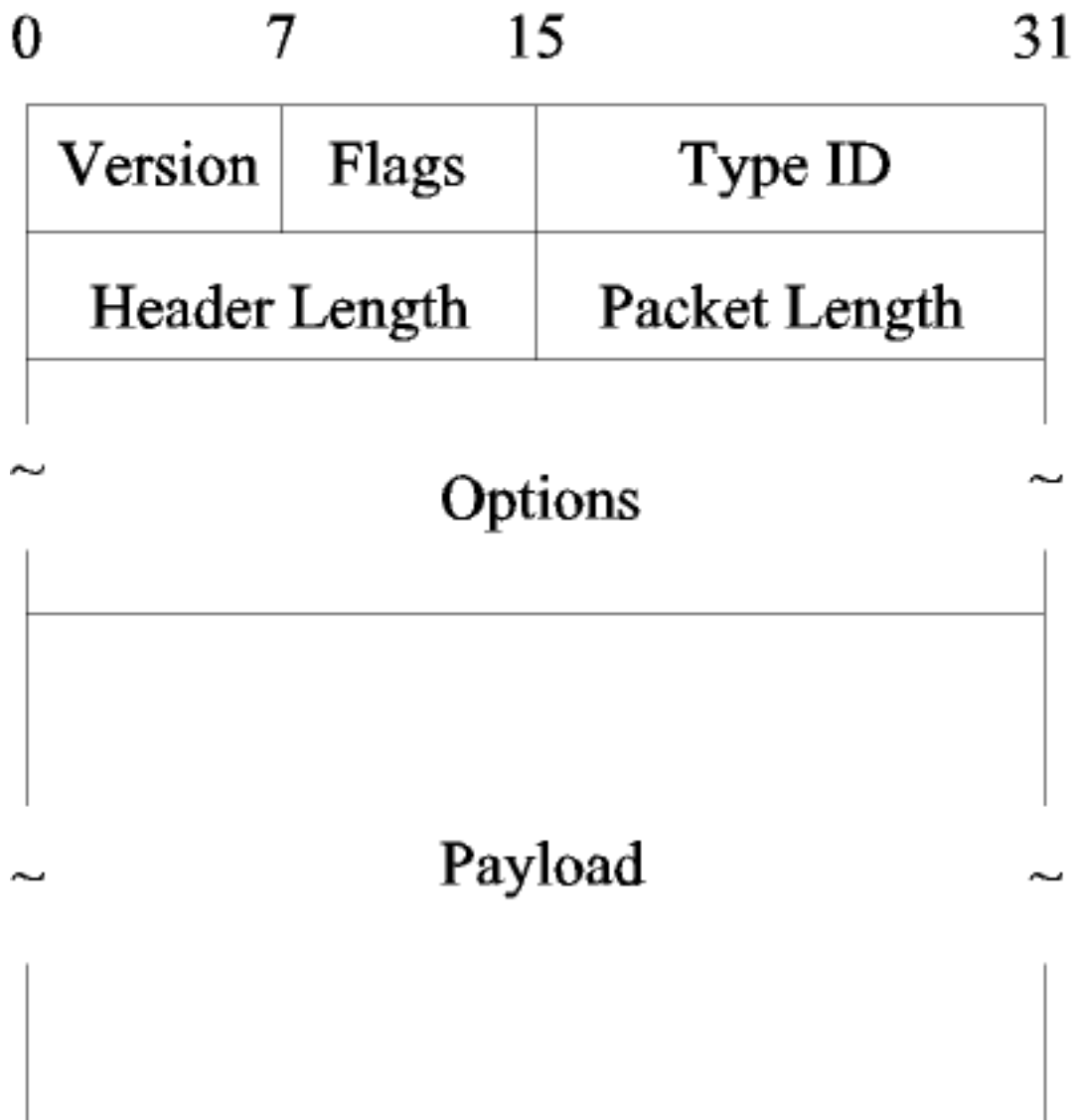
***What happens to existing infrastructure?*** One major issue in the large-scale deployment of active networks would be its interpretability with the legacy network nodes. The good news is that there would be no need to overhaul the existing infrastructure. Active nodes can co-exist with legacy nodes by **tunneling** through these legacy nodes. Much in the same way, MBONE tunnels through 'non multicast aware' routers by having the multicast aware routers at the two ends of a path comprising of both multicast and legacy routers.

## **Active Network Encapsulation Protocol (ANEP):**

Active Networks Group (a group comprising of prominent researchers in the field of active networking) in a memo (RFC) outlines a possible mechanism for encapsulation of active network frames [[Alexander RFC](#)]. The proposed protocol would allow the use of existing network infrastructure in the deployment of active networks. The proposed mechanism promises to be generic, extensible and ensures co-existence of different execution environments. The proposal calls for an active network frame format that would have the programs to be executed integrated into its payload. These programs would be executed at the receiving node in an environment specified by the protocol. Just like IPv6, ANEP would allow specifications of various options in its header that includes but is not limited to authentication, confidentiality and integrity.

### **Proposed Packet Format:**

The document proposes the following format of the ANEP header:

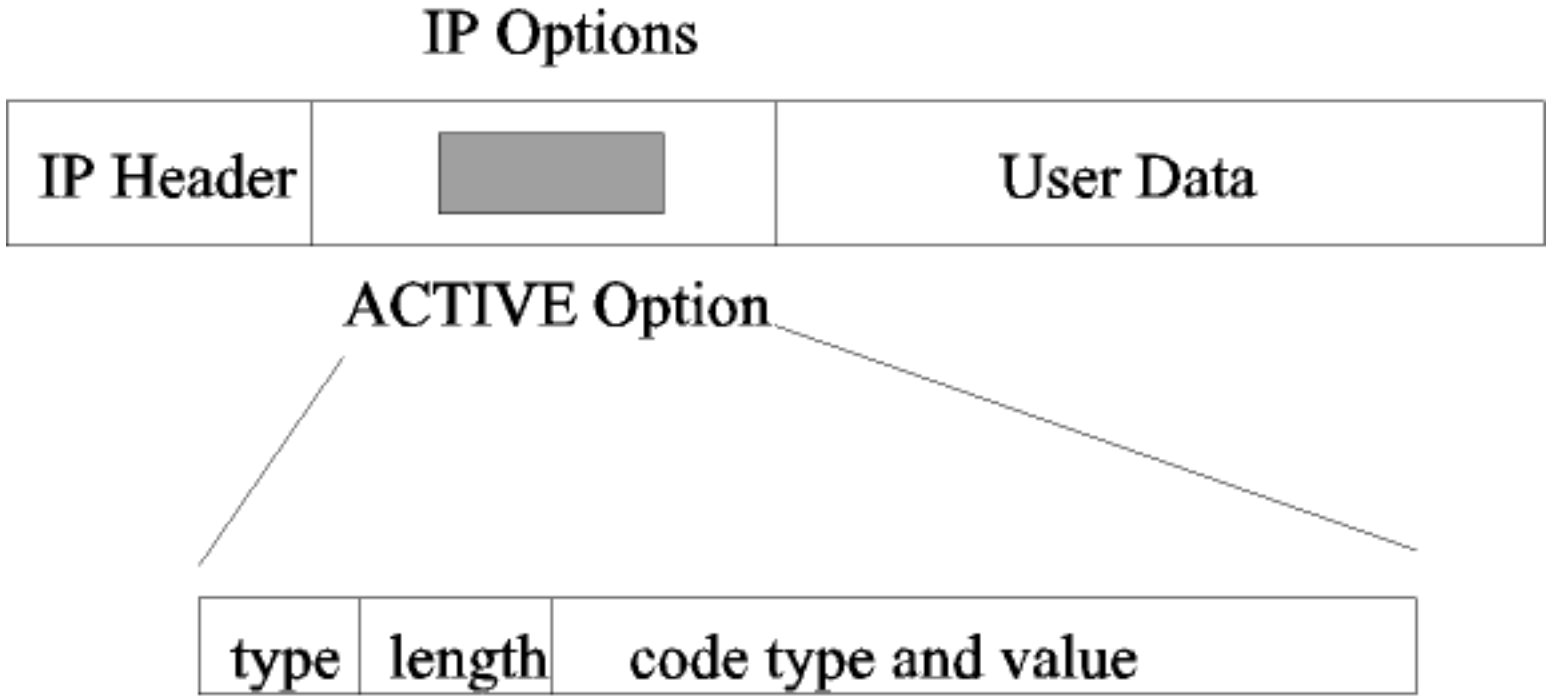


**Figure 1: ANEP Packet Format**

Here, the 8-bit version field indicates the header format in use (current version being 1). Only the most significant bit of the 8-bit flags field is used in version 1 and it indicates whether the node should forward the packet or discard it, if it does not recognize the following Type ID field. The Type ID field would indicate the evaluation environment of the message. A proposed ANANA - Active Networks Assigned Numbers Authority (currently it comprise of one person) would assign the Type IDs, with 0 being reserved for error messages and possible future network layer information. This is followed by header length and packet length fields, both in units of 32-bit words. Which in turn is followed by the options field. Each option in the options field has a 2 bit flag, followed by a 14-bit Option Type field (again controlled by ANANA, however, a 0 in the flag field would mean a non-standard option). A 16-bit option length followed by the Option Payload follows this. The robustness of the proposed protocol ensures that implementing a new protocol or evaluation environment would mean only assigning a new Type ID and a new Option Type.

## Active IP:

As opposed to ANEP, Wetherall and Tennenhouse in [Wetherall 96] propose an extension to the IP protocol that would retrofit active capabilities to the existing Internet. The IP Options are already being used to incorporate new capabilities to the IP network providing for applications like Network Monitoring and Measurement.



**Figure 2: Active IP Packet Format**

As seen in the above figure, the Active IP option field provides a mechanism for embedding a program fragment in an IP datagram. These fragments in the options field are then evaluated and executed at every router along the path. Since, the ACTIVE option is in the payload of the packet; the legacy router would not even see it and forward the packets to active routers. At these active routers, the Active Option code would be invoked for the routers to execute.

[Back to Table of Contents](#)

---

# Active Network Research

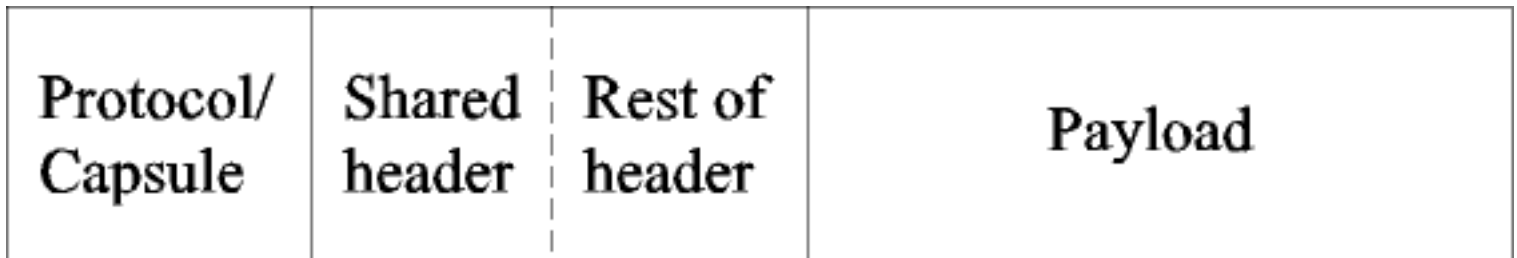
A lot of research effort is put in the development of Active Networking. Key areas of interest are [[Tennenhouse 97](#)]:

- Developing languages and compiler for "active" software.
- Development of platforms suitable for deployment of Active Network nodes.
- Development of programming models emphasizing cross-platform interpretability.
- Active control and algorithms for network configuration.
- Development of middleware services to demonstrate active network capabilities.

This paper would survey the major research project in the field of active networking or related to it. Some of the major research efforts follow.

## ANTS - Active Network Tool kit:

[[Wetherall 98](#)] explains the active network tool kit - ANTS, developed at MIT. ANTS use mobile code techniques that enable new protocols to be deployed at both end systems and intermediate nodes. ANTS use the encapsulation approach with a code distribution mechanism. The protocol incorporates code-forwarding routines into each node, which acquires the application specific routines by a code distribution scheme. The forwarding routines are transferred to related capsule types, i.e. a code group by the code distribution system. The capsule format is shown in the following picture.



**Figure3: ANTS Capsule Format**

Each capsule would carry an identifier for its protocol type, which is described from the code description of the protocol. The identifier is based on the fingerprint of the protocol code, which reduces significantly the risk of protocol spoofing. Also, the protocol enables capsule types to be allocated and enabled in a de-centralized manner quickly. The remainder of the capsule comprises of a shared header common to all capsules in the code group and then the unique header. The protocols are executed within a restricted environment with limited access to the shared resources, thus safeguarding it from runaway protocols. An initial set of primitives is loaded to the nodes. The node provides with an execution model that support issues like network security and resource management. The protocol then has a code distribution mechanism that propagates the program definitions to where they are needed. The scheme works as follows:

- Firstly, the capsule identifies their type and protocol

- Then, when a capsule arrives at the node, it checks its cache to see if the required code is present, if not it request its previous node to send the code to the current node.
- When the previous node receives a load request that it may answer, it sends immediately
- When the current node receives the load response, it loads it into its cache and performs the required action on the waiting capsule.

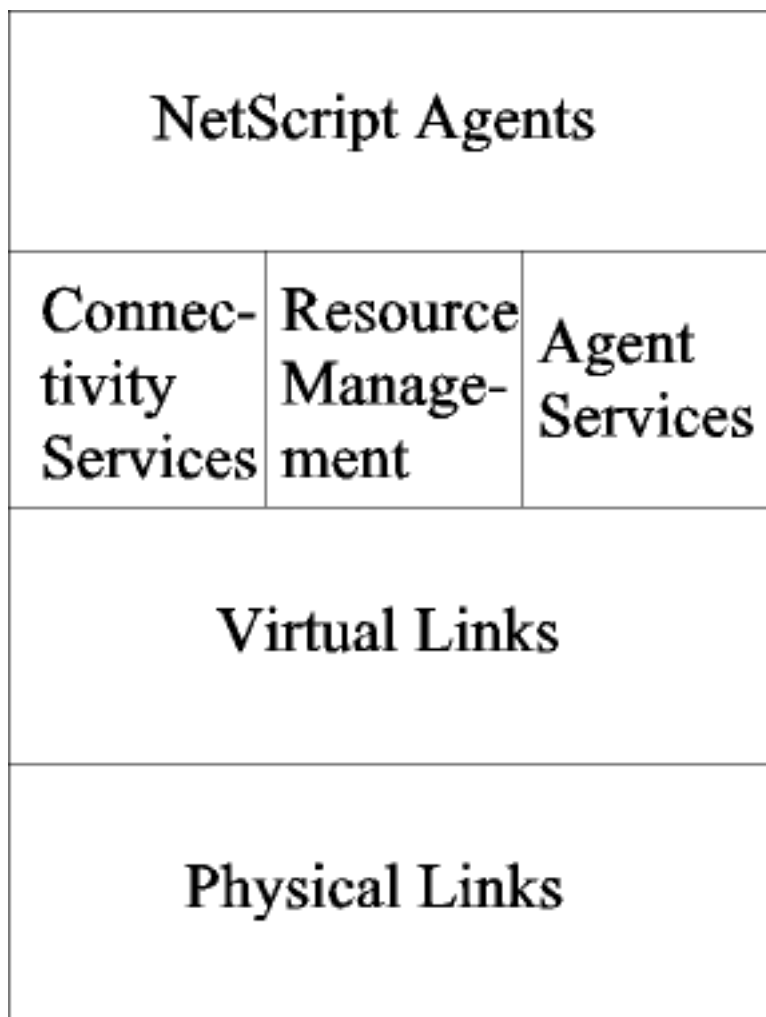
This is an efficient and robust scheme that would allow the nodes to take advantage of active networking.

### **The Switch Ware Project:**

The Switch Ware Project at UPenn proposes a SwitchWare switch, which has a programmable element performing switching functions [[Smith White Paper](#)]. The switch consists of input and output ports controlled by this software programmable element. Programs fragments called 'Switchlets' are sent to the switch, which interprets, and execute them. The goal being to accelerate network evolution, the Switch Ware proposal creates a Virtual Network Infrastructure consisting of SwitchWare switches. SwitchWare is programmed in a higher layer programming language specialized in programming of data switches. Some of the research challenges, SwitchWare hopes to explore includes but are not limited to the implementation of security and resource management policies. These would be embedded into the SwitchWare OS/run-time environment.

### **The NetScript Project:**

As described in [[Yemini 96](#)] NetScript architecture uses the encapsulation approach. A scripting language is developed to process a common set of abstractions of the primitive functions of a traditional data network. To support the processing of these scripts an abstracted 'virtual machine' has been created. NetScript scripts are used to build various packet stream-processing engines supporting flow of packets through them. Such scripts may be used to build routing protocols, protocol analyzers, virtual networks etc. An Active Network thus would provide the environment to execute these NetScript scripts/programs. A collection of these programs may be distributed among multiple nodes forming a Virtual Network Engine (VNE). A collection of these VNE's is connected via Virtual Links (VL). NetScript agents that process packet streams and sends these streams to other VNEs using VLs program VNEs. The collection of these VNEs and VLs is termed NVN (NetScript Virtual Network). NetScript provides the language support to program the NVN.



**Figure 4: VNE Architecture**

The above figure depicts the VNE architecture. The Agent Services Layer is an execution environment for the execution of the NetScript Agent Programs. The Connectivity Services layer controls the allocation of VL's and maintains existing VLs to neighboring VNEs. A NetScript program would be a collection of threads distributed among different VNEs, processing packet streams flowing through these VNEs. The language itself is a small, simplistic data flow language designed for communication based tasks. The language simplifies stream-based computations, since the programs operate on streams of data packets. The language provides the programmer with a "universal abstraction" of a programmable networking device. So a NetScript program would execute on any VNE, which not necessarily have similar configurations. It is a dynamic language, since devices may be added or removed from the VNE on the fly. Since, the language is based on providing data flow; its computations are organized as a set of concurrent, communicating processes where the flow of data drives the computation within the network.

### **PLAN - Programming Language for Active Networks:**

[[Gunter White Paper](#)] describes the effort at UPenn to develop a *Programming Language for Active Networks*. PLAN programs use the encapsulation approach to execute programs at the routers. It basically is a resource limited scripting language that performs resource discovery and diagnostics of the network. The proposed architecture have two levels for router functionality. A *PLAN level* that would

consist of PLAN programs. This level would provide support for resource discovery and diagnostics, that would not require authentication (for example memory and processor cycles available, services available etc.). The higher *Service Level* includes services available only to certain authenticated users. This higher level would house most of the protocols that would comprise an active network.

\* *PLAN References and PLAN 1.0 software distribution is available from <http://www.cis.upenn.edu/>*

[Back to Table of Contents](#)

---

## Summary

This papers highlight some of the research efforts in the filed of Active Networking, which by no means is an exhaustive list. A lot of research effort in put into development of Active Network or similar technologies. Being, a highly dynamic runtime environment, that supports a variety of network services, and allows injection of newly designed services into the infrastructure, active network deployment raises a lot of concerns. Various research groups are researching security being the most important of these concerns. Other crucial research topics include routing, resource allocation, network management services and most important of all mobility. A lot of research effort has been put into the deployment of demo active networks and standardization of all these efforts. Most of the effort is involved in parallel deployment of few different programming models providing an opportunity to explore alternatives [[Tennenhouse 97](#)]. The possibilities offered by Active Networking technologies have already begun to change our perception of a computer network and would play an important role in the shaping of the future technologies.

[Back to Table of Contents](#)

---

## References:

[Tennenhouse 97] Tennenhouse et.al., "*A Survey of Active Network Research*", IEEE Communications Magazine, Vol. 35, No. 1, pp 80-86. January 1997.

Available at: <http://www.tns.lcs.mit.edu/publications/ieeecomms97.html> A very precise introductory source that introduces the reader to the concept of Active Networking, the underlying issues and the ongoing research.

[Wetherall 98] Wetherall, Guttag, Tennenhouse, "*ANTS: A Tool kit for Building and Dynamically Deploying Network Protocols*",

Submitted to IEEE OPENARCH'98, San Francisco, April 1998.

Available at: <http://www.tns.lcs.mit.edu/publications/openarch98.html> Describes the Active Networks Tool kit developed at MIT and gives example of its usage.

[Alexander RFC] Alexander et. al., "*Active Network Encapsulation Protocol (ANEP)*", Draft of an Experimental RFC for the Active Networks Group. This memo describes the proposed protocol for the standardization of Active Network Technologies.

[Wetherall 96] Wetherall, Tennenhouse, "*The Active IP Option*", Proceedings of the 7th ACM SIGOPS European Workshop, Connemara, Ireland, Sept. 1996.

Available at: <http://www.tns.lcs.mit.edu/publications/sigops96ws.html> This paper describes a possible extension to IP, to support the encapsulation of program fragments in datagram.

[Tennenhouse 96] Tennenhouse, Wetherall, "*Towards an Active Network Architecture*", Computer Communication Review, Vol. 26, No. 2, April 1996.

Available at: <http://www.tns.lcs.mit.edu/publications/ccr96.html> A paper explaining the vision and concept behind active networks, includes description of work in progress at MIT.

[Yemini 96] Yemini, da Silva, "*Towards Programmable Networks*", IFIP/IEEE International Workshop on Distributed Systems:

Operations and Management, L'Aquila, Italy, October, 1996. This paper describes the NetScript programming environment, a research project related to Active Networks at Columbia.

[Smith White Paper] Smith, et. al., "*SwitchWare: Towards a 21st Century Network Infrastructure*", White Paper.

Available at: <http://www.cis.upenn.edu/~gunter/wip/ieee.ps.Z> This paper provides an overview of the SwitchWare (a programmable switch) research conducted at UPenn.

[Alexander 97] Alexander, Shaw, Nettles, and Smith, "*Active Bridging*", To appear in the Proceedings of the ACM SIGCOMM'97

Conference, Cannes, France, September 1997.

Available at: <http://www.cis.upenn.edu/~gunter/wip/sigcom97.ps.Z>

[Gunter White Paper] Gunter, et. al, "*PLAN - A Programming Language for Active Networks*", White Paper.

Available from: <http://www.cis.upenn.edu/> Describes the PLAN language for Active Networks

[Bhattacharjee TR] Bhattacharjee, Calvert and Zegura, "*On Active Networking and Congestion*", Technical Report

GIT-CC-96-02, College of Computing, Georgia Tech.

Available at: <http://www.cc.gatech.edu/fac/Ellen.Zegura/papers/anandc.ps.gz> Describes the work done at Georgia Tech on Application of Active Networks in Congestion Control.

[Bhattacharjee 96] Bhattacharjee, Calvert and Zegura. *"Implementation of an Active Networking Architecture"*,  
White Paper presented at Gigabit Switch Technology Workshop, Washington University, St. Louis, July 1996.

Available at: <http://www.cc.gatech.edu/fac/Ellen.Zegura/papers/anwhite.ps.gz>

[Bhattacharjee 97] Bhattacharjee, Calvert and Zegura, *"An Architecture for Active Networking"*, High Performance Networking (HPN'97), White Plains, NY, April 1997.

Available at: <http://www.cc.gatech.edu/fac/Ellen.Zegura/papers/anwhite.ps.gz>

[Legedza 98] Legedza, Wetherall and Gutttag, *"Improving the Performance of Distributed Applications Using Active Networks"*,  
Submitted to IEEE INFOCOM, San Francisco, April 1998.

Available at: <http://www.sds.lcs.mit.edu/publications/infocom98lwg.html>

[Back to Table Of Contents](#)

---

Last Modified: August 14,1997.