Active Networks in Mobile Computing Environment

Afrand Agah
Department of Computer Science and Engineering
University of Texas at Arlington
agah@cse.uta.edu

Abstract—

Telecommunication service providers have begun to offer ubiquitous access to data. As a result, the Internet is not limited to computers that are physically connected but is also available to users that are equipped with mobile devices. This access fuels the growth and the usage of dynamic Internet, which needs increasing support for Internet Protocols and transmission control protocols over wireless/mobile networks.

An active networking architecture provides the infrastructure for applications to inject user programs into the nodes of the network. This enables customization of the network nodes so that application-specific services can be downloaded into the network in the form of new protocols. The aim of the active networking is to design, development and implementation of new communication architecture that allows rapid and safe deployment of advanced networking services. These paper discusses active network requirements and issues in mobile computing environment.

I. Introduction

Traditional networks have the drawback that the intermediate nodes are closed systems, whose functions are rigidly built into the embedded software. Therefore, development and deployment of new protocols in such networks requires a long standardization process.

The concept of active networking emerged from discussions within the broad DARPA research community in 1994 and 1995 on the future directions of networking systems. The idea of messages carrying procedures and data is a natural step beyond traditional circuit and packet switching, and can be used to rapidly adapt the network to changing requirements. Along with a well understood execution environment within network nodes, this program-based approach provides a foundation for expressing networking systems as the composition of many smaller components with specific properties. Services can be distributed and configured to meet the needs of applications and statements can be made about the overall network behavior in terms of individual components [3].

Active networking offers a different paradigm that enables programming intermediate nodes in the network. A network is active if it allows applications to inject customized programs into the network to modify the behavior of the network nodes. This allows applications to customize the network processing and adapt it to the application's immediate requirements.

The convergence of telecommunication networks and the Internet, entails increased support for packet switched networks to route packets generated by mobile users and demands smooth inter-operation of Internet protocols. This convergence is in accordance with the aim of integrated services, where all traffic types are transmitted on the same network resulting in cost reduction and smooth implementation of new services [2].

However, one of the fundamental problems of bringing Internet content to mobile users is that widely deployed Internet protocols assume fixed hosts, which means current protocols operating over the Internet have to be upgraded. Furthermore, these protocols must be scalable to millions of mobile subscribers and adaptive to mobile characteristics. Current protocols such as IP and TCP were optimized for fixed networks, and do not directly support the mobility of hosts.

The remainder of the paper is organized as follows. In section 2 we talk about the active networks. Section 3 discusses the role of active networks in a mo-

bile computing environment. In section 4 a comparison between two active networks system is provided. Section 5 is the conclusion.

II. Active Networks

Traditional networking implementations follow a layered model that provides a well-defined protocol stack. Active networking offers a technology where the application can not only determine the protocol functions as necessary at the endpoints, but also one in which applications can inject new protocols into the network for the network to execute on behalf of the application. The nodes of the network are programmable entities and application code is executed at these nodes to implement new services.

In an active network, data packets are information entities. These entities, which are called *Smart Packets*, contain a destination address, user data, and methods that can be executed locally at any node in the active network. The code in the packet can be in any executable format and it is executed at the node if the node has the correct processing environment.

Nodes in an active network are called *Active Nodes*, because they are programmable elements that allow applications to execute user-defined programs at the nodes. Active nodes perform the functions of receiving, scheduling, monitoring and forwarding smart packets. In active networks routers are programmable. The programs executing at the router are either permanently installed or they exist for the duration of the session [3].

Two major approaches to active networks may be distinguished, discrete and integrated, depending on whether programs and data are carried discretely, i.e., within separate messages or in an integrated fashion.

- *Programmable Switches:* This approach allows users to inject programs into required nodes using a special tag located within the packet. Users would first inject their custom processing routines into the required routers. Then they would send their packets through *programmable* nodes. In this approach, the existing packet/cell format is maintained, and a discrete mechanism is provided that supports the downloading of programs. When packets arrive at the node, the user specific program will be invoked to process the packet's content. Further, the program may perform computation on subsequent packets belonging to the current connection. This is particularly useful when the selection of programs is made by the network administrators, rather than individual end users [2].
- *Capsules Approach:* This approach is similar to how a Postscript printer evaluates its content. In active networks, each packet carries a set of instructions which is interpreted by network nodes. This approach replaces the passive packets of present day architectures by active miniature programs that are encapsulated in transmission frames and executed at each node along their path. User data can be embedded within these capsules. When a capsule arrives at an active node, its contents are evaluated. Bits arriving on incoming links are processed by a mechanism that identifies capsule boundaries. The capsules contents are then dispatched to a transient execution environment, where they can safely be evaluated. The execution of a capsule results in the scheduling of zero or more capsules for transmission on the outgoing links. During evaluation, the capsule has access to built-in routines for accessing resources [2].

The passive packets of present day architectures are replaced with active *capsules*. Capsules use the built-in constructs of a programming language to perform packet processing. The *Active IP* Option field, as shown in figure 1, provides a mechanism for embedding a program fragment in an IP datagram. These fragments are then executed by active routers along

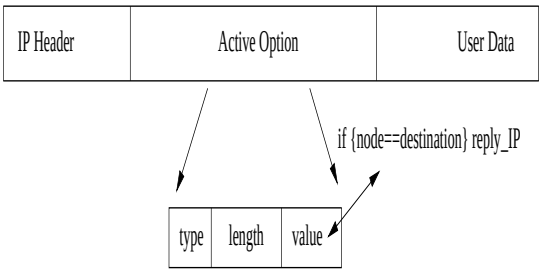

Fig. 1. Format of the *Active IP* field

the path taken by the datagram. Conceptually, the *Active* option transforms the IP options from a fixed set of standardized routines to an extensible collection integrated by a programming language [9].

Today networks can benefit from active networking paradigm with the dynamic deployment of network services that can be tailored to the user's requirements. For the last six years, different vision of the active networking paradigm has led to various implementations of the concept. These active networks architectures differ in many points. The main two implemented models are *Active Node Transport system(ANTS)* from the Massachusetts Institute of Technology, and the *SwitchWare* framework from the University of Pennsylvania. These two are widely available, flexible enough for modifications and experiments, and have been considered as the two candidates for the design of nation-wide IPv6 active network backbone [1].

A. Protocol Classes in Active Networks

The grouping of protocols into classes also helps us to identify common interfaces. We can then define a structure for the packets that implement protocols belonging to the particular class. Protocols of the same class generally require the same set of services from the active node on which they are deployed. For example, protocols belonging to the bridging class require storage at the active nodes. This is because the protocol has to combine packets and sometimes it has to hold a packet from one of the streams while packet(s) from the other streams arrive. Researchers have identified the following classes of protocols [5]:

- *Filtering:* This class encompasses all those protocols that perform packet dropping or employ some other kind of bandwidth reduction technique. Protocols belonging to this class are primarily developed to reduce bandwidth requirements of the application data. Temporary reduction in bandwidth requirements is necessary in the face of transient congestion problems. Applications deploy the filtering code in one of two ways. Applications use smart packets, which "sniff out" the nodes within a set of nodes at which a rate mismatch occurs and installs the filtering protocol at those nodes. Application can also choose to figure out the connection path beforehand and prime the path by installing the protocol at all nodes on the path to tackle any congestion problem [5].
- *Combining:* This class has the property of combining packets that may come from the same stream or from different streams. The Wireless ATM Voice/Data project combines two or more packets from the same stream to form a single packet, which is forwarded to the next hop. Caching research falls into this category because the inherent purpose of caching is to combine common requests from separate streams into one consolidated request and then multicast the reply back to the requesting parties [5].
- *Transcoding:* Protocols that transform the user data into another form within the network belong to this class. Encryption protocols and image conversion protocols belong to this class. Encryption protocols are generally deployed only at the end-points of a connection whereas compression protocols are deployed either at the end-points or at points in the network, where congestion is likely and bandwidth control alternatives are desired [5].

• *Security:* The ability of the user code to access node resources raises the question of security. Active nodes must prevent smart-packets from over consumption of resources, whether done intentionally or not. Active nodes must also guarantee that smart-packets from one application do not interfere with the execution of other unrelated packets. Smart-packets must also be prevented from unauthorized access to node resources such as internal tables [5].

• *Management:* The programmability of nodes in an active network enables the creation of self-configuring, self-diagnosing networks. This involves actions such as alarm and event reporting, configuration management and workload monitoring [5].

• *Routing:* In an active network, routing is an application service as opposed to network management, which is still a network function. Using custom routing strategies, applications create virtual topologies that are overlaid on the physical network [5].

The principal concerns of mobile wireless networks are that the mobile hosts are typically resource-poor, have low bandwidth, and unreliable connectivity to the static elements. The implication of the low bandwidth problem is that data transmitted over the wireless link has to be rate limited to match the bandwidth of the wireless link. This is achieved either by compressing the data before sending it over the link or by dropping low priority data packets. Applications alone can determine what method is the most suitable [2].

III. Mobile Computing Environment

The main contributions resulting from the investigation on the application of active networks in mobile environment are [7]:

• *Management of Location:* The network manages the mobile host's location instead of a home agent. The corresponding hosts can establish connections with the mobile host faster, resulting in efficient routing of packets.

• *Traffic Customization:* Customization is invoked depending on the current state of network. If a particular host is receiving a significant number of binding updates then the network can load an aggregation program, which decreases the number of updates messages, thus conserving bandwidth and reducing the number of packets to be processed at routers.

• *Customized Architecture:* To reduce the number of signaling messages a common practice is to utilize a hierarchical network topology. Hierarchical foreign agents do not necessarily ensure low hand-off latency. Active network-based solutions provide a generic solution that is independent of any topology and also reduce bandwidth requirements of signaling messages.

• *Quality of Service Adaptation:* Variation in QoS is handled internally rather than waiting for end-hosts to adapt. Having the network handle the QoS has the effect of distributing the work load associated with adaptation among the network elements.

• *Reuse of States:* Active networks are particularly useful in connection oriented networks and multicast protocols, where a lot of information is maintained within the network. The advantage over end-hosts based solutions is that these states can be reused. If an end-host solution is used, a lot of probing is required to figure out the location of information within the network before any update is performed.

A. IP and Mobility

Mobile IP research is sparked by the need for a protocol, which supports mobile devices whose point of attachment changes frequently. Current protocols require changes to a mobile host's address, modifications to a number of configuration files and restart of all communication sessions when moving from point A to B. A mobile host must provide the impression

of a stationary host and its point of attachment should be hidden from protocols and applications.

The main goal of mobile IP research is to introduce transport layer transparencies and to eliminate the need for re-initialization, when the corresponding host changes its point of attachment. Another goal is route optimization. Route optimization requires caching the mobile host's care-of-address and updating it when the mobile host changes its point of attachment [4].

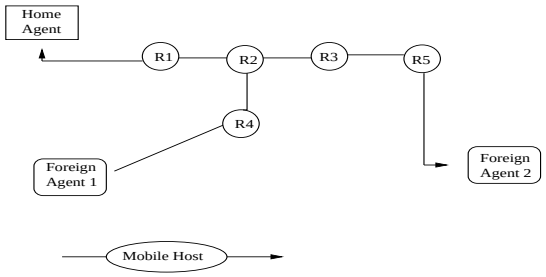

Fig. 2. Setup of the network

In figure 2, R1 to R5 are programmed routers. The mobile host is moving from first foreign agent, to the second foreign agent. Node R2 notes the mobile host's new care-of-address and forwards the registration request. At this point in time any packets destined for the mobile host will be re-tunneled to the mobile host's new location by R2. In addition, a binding update and registration reply message is sent to the corresponding host and mobile host respectively. Thus, enabling the corresponding host to tunnel packets directly to the mobile host's new location and reducing hand-off latency. Besides that new corresponding hosts wishing to communicate with the mobile host have a faster connection setup time since active router can redirect packets to the mobile host's current care-of-address without going through the home agent. This enables efficient delivery of binding updates [4].

B. Hierarchical Hand-off

In the active delivery scheme hand-off process, a registration reply is sent to the mobile host or receipt of a registration request is processed. Hence hand-off latency is reduced since the mobile host does not rely on its home agent. The active delivery scheme works both locally and globally; the active routers are dynamically set up to promote scalability. This is because as the mobile host migrates to a new domain, the local router gets programmed and handles all registration requests within the given domain. The central idea as in cellular IP is to shield mobility from corresponding hosts and home agents [4].

In cellular IP, as long as mobile hosts migrate within a domain maintained by the cellular IP protocol, the home agents or corresponding hosts do not need to be updated.

When an active delivery scheme receives registration request, it sends a registration reply back to the mobile host confirming the hand-off process. As a result, the hand-off latency is not dependent on the delay between the mobile host's new location and its home agent but is dependent on the delay between itself and the closest programmed active router. All active registration requests are relayed to the mobile host's home agent. So the program could relay the request only if the mobile host has moved across subnet. As a result, the number of active registration requests directed towards the home agent is reduced. The benefit of this feature is crucial, where hand-off rate is high. Due to the high mobility rate, the home agent experiences a higher number of registration requests from the mobile host, hence increasing bandwidth used and home agent load [3].

C. Packet Redirection

When a mobile host migrates, packets are redirected at each active delivery scheme during the mi-

gration itself. Simultaneously, the corresponding hosts are updated with the mobile host's new care-of-address. During this interval, active scheme prevents packets from being forwarded to the wrong address. When active scheme receives an ordinary packet it checks whether the packet is going towards the correct destination. If the packet is headed to a wrong address then the packet is redirected. The check also determines whether the packet is encapsulated. If so, the packet is decapsulated and re-encapsulated with the mobile host's care-of-address and sent to the mobile host. When the mobile host has a high hand-off rate or the mobile host is unreachable, the schema can be programmed to delay the delivery of messages until the mobile host becomes available [4].

D. Smooth Hand-off

Buffering at the previous foreign agent might not be useful if the hand-off latency plus the time for the binding update to get to the foreign agent is large. To overcome this problem, foreign agent can be programmed with "Smart-Buffer". This Smart-Buffer sends acknowledgements for packets buffered at the foreign agent while waiting for binding updates from the mobile host. It monitors all data segments and acknowledgement packets going to and from the mobile host. Smart-Buffer only sends acknowledgement for packets that have not been acknowledged due to migration and new packets that arrive after migration [4].

To avoid over-buffering at the new base station, an acknowledgement specifying the available receiver buffer size is sent to the sender. By sending the allowed buffer size to the sender, excessive buffering at the base station can be avoided due to inappropriate window size at the sender.

There are two ways in which the appropriate buffer size is determined. First, if a smart-buffer program is running at the new base station, the average advertised window size is used by other connections is used. The smart-buffer program only needs to record the allowed window size from the acknowledgement packets. Therefore, the average window size can be easily calculated from connections managed by smart-buffer. Second, if the smart-buffer program is not installed, then the buffer size allocated by the active environment is used given that the window size advertised by the mobile host is larger. In other words, if the mobile host is requesting more than it should, then the advertised window size is replaced with the buffer size allocated by the active environment [4].

E. Mobile Host Migration

When the mobile host migrates, the router previously responsible for the corresponding host may no longer be valid. Therefore, an update mechanism is required to determine whether the responsibility roles have changed. In active delivery scheme, the update process is initiated during the hand-off process. In other words, when registration request is received or upon receipt of a binding update sent to the previous base station. When any of these messages are received, a binding update is sent towards the corresponding host in which it has responsibility. In addition, each corresponding host is informed to generate an active discovery capsule towards the mobile host's new location. Also, in the active discovery capsule, the local IP address is included. When a router serving the mobile host intercepts the new active discovery capsule, this capsule is authenticated. If the responsibility variable of active router in the capsule is similar to that of the local IP address, then no roles have changed. On the other hand, if this variable does not match the local IP address, and the recorded hop count is smaller, then an invalidate message is sent to

the router [4].

F. Multicasting

Current multicast protocols such as distance vector multicast routing protocol are designed with static host in mind and hence are prone to problem in mobile networks. Typical problems with multicasting in mobile networks are:

- After migration, multicast protocols that are based on shortest path tree may route packets incorrectly or drop packets.
- At the receiving end, when a mobile host migrates to a cell with no other members, it will experience delay. This is mainly caused by subscription delay, tree rebuild or non-existent multicast routers in the region.

When a receiver migrates, three issues need to be considered. First, the foreign network may not support multicast service. Therefore, the receiver is unable to rejoin the multicast session until it migrates to a network, which supports multicast. Second, the foreign network may support multicast service but does not join the multicast group in which the visiting mobile host is subscribed to. Third, in case where the foreign network has joined the multicast group, the receiver may receive duplicate packets or only subsequent packets.

The active network-based solution, which is called AMTree, addresses these problems. AMTree takes advantage of the processing capabilities at routers, which enable mobile hosts to continue sending packets to receivers after migration. Hence the multicast tree can be maintained with minimal modifications and minimal packet latency resulting in low hand-off latencies [4]. The multicast tree incorporates the bidirectional state of tree and is not dependent on the state maintained by the underlying routing protocol. An active router can be easily programmed with monitoring and traffic management protocols applied only to parts of the tree. Here only one multicast tree is required for a given session. Furthermore, periodic messages are not required to acquire topology changes, and routers do not need to maintain pure information. Senders that are not members of the multicast session can send packets to the tree. When a source is mobile, the source only has to form a new connection to the active router. As far as the router concerned the new connection formed is from a new source given that the mobile source has obtained a new care-of-address. At the receiving end, receivers may experience delays due to the reconnection but the multicast tree formed is still valid [4].

G. Congestion

Transmission bandwidth and computational power will both continue to increase, and so the application requirements for bandwidth. Network node congestion will be due to bandwidth limitations, and even congested switches will have considerable processing power available. Also there are always applications that prefer to adapt their behavior dynamically to match available network bandwidth. So an active processor must support some of the following functions [2]:

- *Buffering and Rate Control:* The most direct translation of sender-based adaptation to active networking is to have active processor monitor the available bandwidth and rate-control the data, buffering it and metering it out at the available rate.
- *Media Transformation:* Active networks can do more than simple intelligent dropping of data. A particularly powerful capability is the transformation of data at a congestion point, into a form which reduces the bandwidth but preserves as much useful information as possible. This may allow the active node to create the form of the data, which the application

would have created, had it known about the bandwidth limitations encountered at the congestion point.

H. Security

Some of the most challenging aspects of securing active networks concern the authentication support for authorization. Authorization decision requires the authentication of the entity making a request. Authentication normally implies the use of cryptographic techniques [11].

Security processing in the nodes would follow the following sequence of actions: (i) receive packet; (ii) verify hop-hop integrity; (iii) assign packets to existing domain; (iv) extract credential list; (v) check credentials authenticity according to authentication policy for the domain; (vi) check credentials against access control policy for domain; and (vii) deliver entire packet to the domain, including the credentials, authentication protection fields, etc.

Security processing in the execution environment would include: (i) receive a packet including credentials; (ii) create a sub-domain, providing the security context parameters for the domain; (iii) modify the access control policy of a domain; and (iv) add or remove cryptographic protections to user data.

But the application of existing cryptographic techniques to the active networks environment presents certain challenges. First, the identification of the principal itself in active networks is challenging. Existing Internet interactions are typically client/server, where the explicit individual identity of the client and server are important. The Internet community has begun to move away from explicit individual identities to attribute based identities. Second, the choice of an authentication mechanism presents challenges in active networks. Existing mechanisms for providing authentication protection of a packet are rooted in the existing Internet paradigm of client and server based communication. These will not be sufficient in an active network environment, where the packet needs to be authenticated at source and destination and potentially every node in between.

The existing solutions can be used hop-hop in the path but that provides little in the way of end source authentication. When hop-hop protections do not provide sufficient end-end authentication of the principal associated with a packet, we can employ end-end protections. However, the use of end-end cryptographic techniques is also a challenge in active networks. Symmetric techniques could be installed at each node of the packet's path through the network.

The packet modifications at each node could be protected anew with the shared key. However, this has a similar trust drawback as using hop-hop protection: every node on the path must be implicitly trusted. Also, the assurance of authenticity of the principal, derived from the shared key, is diluted if the key is not unique to the principal and the path.

Asymmetric techniques can operate in a datagram model but have difficulty protecting packets that change. Signing a packet with a digital signature provides a cryptographic association from the signer to every potential verifier of the future. Therefore, authentication by digital signature is suited for a datagram model of communication, where the packet may decide in route what nodes it will visit [9].

IV. ANTS Vs. SwitchWare

We can identify a main difference between the two architectures, that we introduced at the beginning of the paper. In respect to the distribution process of codes between active nodes, ANTS differs from SwitchWare. Indeed, ANTS is based on active node, i.e., the packets do not carry the mobile code, but only identifiers or references to predefined functions to be downloaded. At the opposite, Switch-

TABLE I

FEATURE COMPARISON BETWEEN ANTS AND SWITCHWARE

Criteria	ANTS	SwitchWare
Environment	Java	Java
Link Layer	UDP	UDP and TCP
Routing	static	Static and dynamic
Protection	Java security	Authentication

Ware uses an hybrid approach based on active packets and nodes, i.e., packets can carry code which is relatively simple and restricted while active nodes can provide any complex code that dynamically download if needed. Their ability to provide flexible and dynamic deployment of IP network services, have been tested. There is a tradeoff between efficiency, simplicity and other aspects such as security and resource management. The SwitchWare model pays more attention than ANTS to the latter aspects but also receives a higher penalty for processing at the nodes. These systems use two different techniques to distribute the user-controllable mobile codes in active nodes. Experimental results show that short live sessions will gain using SwitchWare and long lived multimedia communication sessions will gain using ANTS. [8]

Differences between these two models are summarized in table 1.

V. CONCLUSION

The active networking paradigm aims of producing a new open networking platform, flexible and extensible at runtime to accommodate the rapid evolution and deployment of networking technologies and to finally provide the increasing sophisticated IP-based multimedia services. Given a programmable network, the main concern is to determine the improvements that can be made to existing protocols, such as mobile IP. Due to software extensibility provided by active networks, significant amount of improvements can be made, if computations are performed at the routers. There are two main implications to local processing. First, end-hosts do need to probe the network. This improves the scalability of any scheme since the number of signaling messages is reduced. Second, processing is only done on the latest information. Due to the dynamic changes of the network, computation can be performed when events happen and not after it has happened. These improvements, in particular to routing in mobile networks, contribute greatly to active networks research and provides a new perspective on possible solutions that may be incorporated into existing protocols.

REFERENCES

[1] N.Achir, M.Fonseca, Y.G.Doudane, N.Agoulmin and A.Mehaoua, "Active Networking System Evaluation: A Practical Experience", *Networking and Information Systems Journal*, vol.3, no.5, 2000.

[2] S.Bhattacharjee, K.L.Calvert and E.W.Zegura, " On Active Networking and Congestion", Technical Report, Georgia Institute of Technology, 1996.

[3] K.L.Calvert, S.Bhattacherjee, E.Zegura and J.sterbenz, "Directions in Active Networks", *IEEE Communications Magazine*, pp:72-78, 1998.

[4] K.Chin, "An Investigation into the application of Active Networks to mobile computing environments", Ph.D Thesis, Curtin University of Technology, March, 2000.

[5] A.B.Kulkarni and G.J.Minden, "Active Networking Services for Wired/Wireless Networks", *Proceeding of INFOCOM*, 1999.

[6] P.Menage, "PCANE: A resource controlled framework for active network services", *In 1st International Working Conference on Active Networks*, Berlin, July, 1999.

[7] B.Schwartz, "Smart Packets for Active Networks", *In 2nd Conference on Open Architectures and Network Programming*, New York, March, 1999.

[8] D.L.Tennenhouse, J.Smith, W.D.Sincoskie, D.J.Wetherall, and G.J.Minden, "A Survey of Active Network Research", *IEEE Communications Magazine*, pp:80-86, 1997.

[9] M.Tiwari, " Active Networks", Technical Report, Indian Institute of Technology Kanpur, April 1999.

[10] D.Wetherall, "Experience with a capsule-based active network", *SIGCOMM*, 1999.

[11] D.Wetherall, "Active network vision and reality: lessons from a capsule-based system", *Operating Systems Review*, vol. 34, no.5, pp:64-79, Dec.1999.

[12] D.Wetherall, "ANTS: A Toolkit for building and dynamically deploying network protocols", *in 1st Conference on Open Architectures and Network Programming*, pp:117-129, California, April, 1998.

[13] D.Wetherall and D.L.Tennenhouse. "The ACTIVE IP Option", *In 7th SIGOPS European Workshop*, Ireland, sept., 1996.

[14] Y.Yemini and S.D.Silva, " Towards Programmable Networks", *International Workshop on distributed Systems operations and Management*, Italy, October, 1996.