IPv6: Paving the Way for Active Networking [1][2]

David Larrabeiti, María Calderón, Arturo Azcorra, Alberto García
Universidad Carlos III de Madrid

Jens E. Kristensen
Ericsson Telebit A/S

Abstract. Active Networking is built upon a new network architectural paradigm where routers or switches are considered as regulated open execution environments where users can dynamically load and run code that can perform any sort of processing to packet flows - usually above layer 3. In many ways, IPv6 technology has addressed several problems that can pave the way for this technology. This paper analyses IPv6 from the active networking viewpoint and introduces an active network architecture developed in the context of the IST project GCAP.

1 Introduction

During the last years most of internet actors' interest has focused on service creation and content production, and this trend has grown at the pace of the business development of the internet. Being the first one to put a new service into the internet market represents an advantage over competitors whose magnitude is deemed to be much higher than the one obtained in conventional markets. In some cases, new services mean new protocols that may have to be deployed at network nodes, rather than at end-points. Specific processing of packets, performed for the users inside the network in a transparent way, is a requirement for advanced services such as address translation, port masquerading, transparent proxying, TCP spoofing, intelligent multimedia flow rate adaptation and multipoint aggregation, scalable reliable multicasting, etc. Although some of these applications can also be provided on an end-to-end basis, its inclusion as part of the network infrastructure can offer an added value in terms of scalability, network resources control, performance or ease of reconfiguration. The basic idea behind active networking technology is exactly to provide a network architecture offering flexibility concerning where to put the intelligence and open it to users.

[1] This work has been funded by the IST project GCAP (Global Communication Architecture and Protocols for new QoS services over IPv6 networks) IST-1999-10 504.

[2] Disclaimer: this paper reflects the view of the authors and not necessarily the view of the GCAP Project.

An important factor contributing to the success of internet protocols is its rapid development cycle, that is open to comments and implementations from all over the world, showing in some regards a clear parallelism with the mechanisms of the free software movement. Even so, what we could call *time-to-network*, i.e. the lapse of time between protocol conception and availability at every node in a multi-vendor network, is still very high when compared to the timing scales of internet development. This seems an important hinder to the deployment of new protocols. Traditionally, protocol deployment takes the following steps: step 1) host-only implementations making use of existing networks as bare carriers (typically using UDP encapsulation); step 2) CPU-based native protocol implementations available at a few routers and usage of tunneling through non-supporting nodes to interconnect protocol-supporting islands, and finally, once the protocol is considered mature; step 3) optimized code is available and distributed over most network nodes. Examples of this process can be found in many routing protocol implementations, and, more recently, in the deployment of multicast routing and of IP version 6.

This deployment process can be dramatically simplified if routers give some sort of standard execution support to external code in a controlled way. That is the target of active networks. In this paper, we are giving a brief overview of this user-oriented service and its relationship with IPv6.

2 Active Networks

During the last years, active network technology has been subject to intensive research, as a response to critical analysis on the fundamentals of Internet protocols started in 1995 in DARPA workgroup discussions. Such critics are based on the important hinder to the fast deployment of new network services and protocols stem from the current internet node definition. Unlike application layer technologies where the time-to-market is extremely short (usually estimated in 6-month), the time-to-network of network protocols depends heavily on agreements among vendors and must follow a standardization process before most network devices implement them (in the 5 to 10 year timeframe). Two examples of this phenomenon are the slow pace followed by IP version 6, in spite of the broad range of transition mechanisms provided in its design, and the case of multicast in IPv4, a service that market pressure in content delivery networks was expected to push to production networks time ago.

The commercial phase of the Internet has leveraged the demand of new services and applications. In this environment, service providers need to quickly adapt to the market winds at the speed of Internet development. The problem comes when new services require processing inside the network (or just the usage of the network resources ets significantly improved if processing is inside). Classical examples are: QoS multimedia flow adaptation by means of trans-coders or intelligent packet discard, scalable reliable multicast, distributed network management, flow control, im-

plementation of traffic engineering procedures and CoS policies, congestion control, etc.

The term active networking refers to an evolution from the traditional network model, that claims the enhancement of network programmability to the extreme of offering this facility to end users. In other words, active networking is an emerging technology that introduces the idea of routers as execution environments open to users' code in order to achieve ad-hoc processing of user packets. In this sense, this model is strongly related to agents technology when considered the target service the network behaviour itself. This means a big change in the network paradigm: from nodes just able to transport bytes in a passive way, to nodes able to process packets at any layer of a protocol stack.

Therefore, active networks are considered active for two main reasons:

- network nodes load dynamically the code needed to process the different packet types. Several approaches have been studied: from packets ("capsules") carrying the code to process them at traversed nodes, to control-driven code shared by a set of flows. Fig. 1 illustrates the idea.
- network nodes can perform specific processing to packets up to application layer. For example, a node running an audio mixing active application could mix audio samples from N packets input from N interfaces and output a single mixed packet.

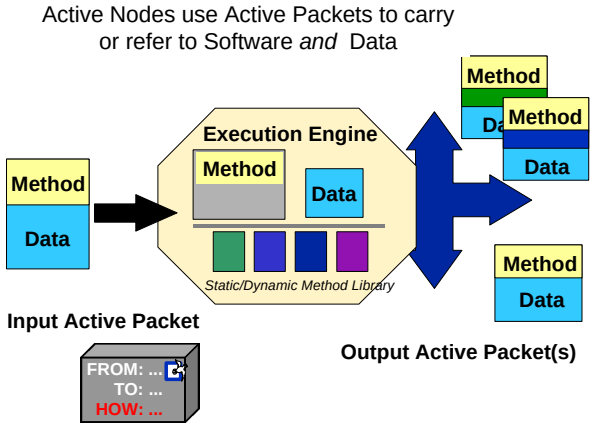

Fig. 1. An active packet traversing an active node

From this definition, it easily follows that the main advantage of active networking technology is the dramatic reduction in the time to deploy new protocols and services. This feature has attracted the interest of industry, timidly adopting a few ideas from active networks in the design of services for which the programmability of the network is essential. The key idea is that the active networking framework can be an effective means for fast deployment of applications, services and new protocols across different vendor architectures under the basic premise that a standard execution sup-

port is widespread and that featuring this support does not cause too much penalty on routing/forwarding performance.

3 Active Networks and IPv6

The classical mechanism used by hosts to access most high layer services other than packet forwarding is by means of control plane entities that communicate with service agents. This procedure is predominant in IPv6, like it is in IPv4 applications. In this method, the user node is responsible for locating the nearest agents (real servers or proxies) that provide the service. One of the simplest and most manageable way of dealing with automatic server location within a given administrative domain in a centralised way is by means of DHCP (Dynamic Host Configuration Protocol). DHCP provides a stateful way of assigning resources with programmable policies. For example, a DHCP agent can feed a user node with all the information needed, not only to self-configure IPv6 addresses on start-up, but any other network service: NTP agents, HTTP proxies, DNS servers, mobility home agent, H.323 gatekeeper, and of course, even the TFTP server where to download its OS from. If the main DHCP server is not link-local to the self-configuring node, and the IPv6 host has not yet a global unicast address assigned, a link-local proxy can talk to the server on its behalf.

A complementary and more automatic way of searching for the nearest servers is using multicast incremental ring searches. This is especially suitable for IPv6 since multicast support is a mandatory feature in the new version of IP. In this approach, the client host relies on a set of well-known multicast groups associated to each service. When the client needs to locate a server, it sends successive multicast probes addressed to the service group over the IPv6 network with an increasing hop limit in its header from 0 up to 255. Obviously, one will be the maximum hop limit value if the client has not been assigned a global address yet.

The approach described above is especially appropriate for services implemented statically, usually of transactional nature, based on servers. But, how does an end-user access and program network services when the packet processing is distributed and performed by the routers along the path to a destination? This is the case of active networking. Obviously, the address of the nearest active node could be obtained by one of the general procedures described above. But it seems that the explicit addressing of routers and relaying between active nodes does not match the dynamic nature of the structure of an IP backbone. Moreover, it would not work with native multicast packets where a tree of processor nodes are implied. Last but not least, it is important that active processing can operate on regular (non-active) packets. Therefore, an important requirement for the effective deployment of network-based services seems to be agent location transparency or, in other words, active node location transparency, also tightly linked to mobile agent technology. This means that senders expecting special processing to their packets by the network simply address the packets to their

destination, and routers recognize them as special packets and process them according to a given code.

A clear example is an n-to-n multi-QoS multimedia flow service over a best effort internet, for example in videoconferencing applications, for terminals with heterogeneous capabilities or different access link capacities. One valid option is using layered coding and multicast, letting IP reduce the rate of the flow at congested links. A second (and complementary) way to implement this service, more complex but also more effective in terms of overall bandwidth usage is performing intelligent packet discard, rate adaptation, transcoding or layer selection, etc at network nodes branching to heterogeneous receivers on different links. The advantage of this approach is that intelligent processing within the network can adapt the flow to the specific needs of a subtree of receivers and prevent forwarding packets that will never reach its destination due to bottlenecks downstream or that will not be profitable in the playback. In this case, if active node location transparency is enabled, each party can multicast its traffic unaware of which network nodes in the distribution tree will adapt the flow. A second important example is reliable multicast, which can become scalable thanks to distributed retransmission and aggregation (avoiding the well-known nak implosion problem). And the same can be applied to multiple packet processors available in the market today: TCP or HTTP spoofing, layer 4 switching, NAT, masquerading, content filters, etc which are in fact incipient forms of statically preconfigured transparent active networking.

However, there is an important drawback to the implementation of transparency of active nodes: efficiency (letting alone other important efficiency problems already analysed by other authors like making the router execute user code, that we will assume here solvable by means of ad-hoc active code processors). To apply active networking in a realistic context, it should be assumed that regular packets are to be processed by the same router as active packets, and active routers are supposed to live together with non-active legacy routers. How can a router - whose behaviour is optimized for packet forwarding just by checking the destination address - keep its performance up if we require a special treatment to packets not explicitly addressed to them?

This problem is not new. A solution already deviced for signaling protocols that need such a feature is the Router Alert Option of IP (RFC-2113) [1]. As described in this document, the Router Alert Option has the semantics: "routers should examine this packet more closely (check the IP Protocol field, for example) to determine whether or not further processing is necessary". A new option type is necessary because some IP options are already implemented in the fast path of some routers. Only options not supported in the fast path will push the packet into the slow path and hence, in principle, no performance penalty is caused to regular data packets.

```
   Type    Length        Value
+--------+--------+--------+--------+
|10010100|00000100|  2 octet value  |
+--------+--------+--------+--------+

Value:
```

```
0 - Router shall examine packet
1-65535 – Reserved
```

Fig. 2. RFC-2113 Router Alert Option format for IPv4

Current protocols using this option include IGMPv2 (RFC-2236) [2] and RSVP (RFC-2205) [3], where PATH packets, addressed to their final destination, are modified at every step to fetch the path characteristics from source to destination.

In IPv6, Router Alert (RFC-2711) [4] is a Hop-by-Hop Option with the same general semantics. However, unlike IPv4 where only value 0 is defined, in IPv6 three values have already been reserved, one of them (value=2) for active network messages. The others are assigned to RSVP and to Multicast Listener Discovery messages included in ICMPv6.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 0 0|0 0 1 0 1|0 0 0 0 0 0 1 0|        Value (2 octets)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

   Value:  A 2 octet code in network byte order with the following
   values:

      0        Datagram contains a Multicast Listener Discovery
               message [RFC-2710].
      1        Datagram contains RSVP message.
      2        Datagram contains an Active Networks message.
      3-65535  Reserved to IANA for future use.
```

Fig. 3. RFC-2113 Router Alert option format for IPv6

Figure 3 shows the router alert option format in IPv6: the first three bits of the first byte are zero (nodes not recognizing this option type should skip over this option and leave it unchanged) and the value 5 in the remaining five bits is the Hop-by-Hop Option Type number.

Another important feature of IPv6 very useful in active networking is the base header flow label field. Router alerting packets are extremely useful to program the behaviour of active nodes along a given path or domain. However, per-flow processing also requires fast paths to specialized processors. This can be achieved basing internal flow management on flow labels. The base IPv6 header is a priviledge position for a label assigned by IP at the source host for a single data flow and preserved throughout the network. Its purpose is preventing the analysis of higher layer protocol headers when packet classification is necessary. Once a pair <source address, flow label> has been identified by a previous signaling router alerting packet as an active flow, the node gets programmed to provide ad-hoc processing to that flow at full speed. In this sense, active networking can be used as a vehicle for the rapid development of protocols for traffic engineering.

If active node location transparency is not desired, or the user just wants to simply select the network provider that will implement active processing, IPv6 also features a

standard powerful tool: the routing header. Thanks to the ubiquitous support of this type of source route specification - as well as the necessary security methods enabling it in practice - as a conformance requirement for all IPv6 implementations, it is possible to specify the active nodes that will implement the service. This is very important in networks where forward and backward paths are different. The requirement that the reply from the endpoint will carry the same routing header is important to make sure that signaling requests and responses of the active protocols designed will follow a coherent sequence of nodes i.e. the active entities are the same, despite the direction of packets. This feature can be used in junction with anycast addressing to pass through the nearest active nodes or let the network decide the cheapest active route at a given time.

4 A Pragmatic Approach: Router-Assistant Architecture

So far, most work carried out on active networking has been very theoretical and therefore just a few ideas have been taken to industrial products. With the purpose of creating a pragmatic approach to active networking for IPv6 more realistic in an industrial context, a new active networking architecture design is being developed in the context of the IST project GCAP [5]. In this framework, user code is dynamically loaded on nodes by active packets featuring location transparency using the IPv6 router alert hop-by-hop option. The environment assumes that user code is validated off-line prior to its deployment and, once proofed harmless, stored in secure servers that can be accessed using existing authentication and encryption technologies. This code is shared by a given flow and has a programmable lifetime in a soft-state fashion. An important architectural component in this design is the concept of active *Router Assistant*, which is connected to an Ericsson Telebit IPv6 AXI462 router. The router delegates active processing functions to the router assistant which is directly attached to the router by means of a high-speed local area network. Another original feature is the usage of SNMP to provide active applications with a view of the router, an important component if active applications must work on behalf of the router. There is a parallelism between the concept of assistant-router and the concept of controller-switch in IP Switching (Ipsilon, 1996). In the first case the router delegates high layer processing on the assistant; in the second, the controller delegates forwarding of a given flow on the ATM switch.

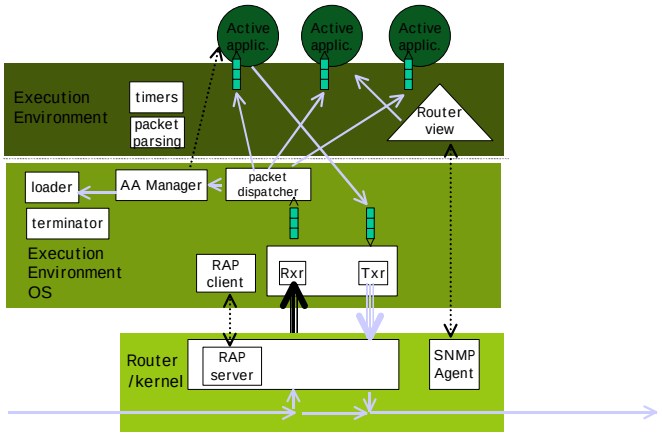

Fig. 4. A Router-Assistant Active Node Architecture

Figure 4 shows how active applications running on network nodes are dynamically downloaded by users onto a Java execution environment (at the top of the picture). A network node is here split into two functional blocks: a layer-3 forwarding engine (a router) and a higher-layer processing unit (the assistant). The system is controlled by a module featuring the role of OS for active applications that dispatches packets as they are filtered by the real router (at the bottom) on their way to their destination. One of the main advantages of this approach is a clear decoupling between regular routing/forwarding and active processing functions.

5 Conclusions

The new network layer standard IPv6 is expected to gain momentum in the short term. Active networking has a big potential in next generation networks where the network programmability is paramount. The design of IPv6 has included new features and rules that can pave the way for active networking. However, practical approaches that preserve router performance on regular packets are very important for this to become real. This issue has been the rationale for the design of the Router-Assistant Active Node Architecture presented in this paper.

Is active networking technology awaiting a killer application? For the authors of this work, the answer is clearly no. In fact, some of the active applications are present in the market today but implemented in a less versatile way than applying the active network paradigm. Effectively, in the broad sense of the definition of active networks, there do exist statically loaded active network applications usually running at edge devices. Level 4-7 switching, NAT, masquerading, filters, CoS markers, traffic condi-

tioners, etc are actually instantiations of transparent active applications. Whenever a real network service demand appears, an ad-hoc device implementing it goes out to the market. Therefore, the main practical applications of active networking still are prototyping and customized network processing for specific user needs.

References

[15] RFC-2113. IP Router Alert Option. D. Katz. February 1997.(Status: PROPOSED STANDARD)

[16] RFC-2236. Internet Group Management Protocol, Version 2. W. Fenner. November 1997. (Updates RFC1112) (Status: PROPOSED STANDARD)

[17] RFC-2205. Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification. R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin. September 1997. (Updated by RFC2750) (Status: PROPOSED STANDARD)

[18] RFC-2711. IPv6 Router Alert Option. C. Partridge, A. Jackson. October 1999. (Status: PROPOSED STANDARD)

[19] RFC-2710. Multicast Listener Discovery (MLD) for IPv6. S. Deering, W. Fenner, B. Haberman. October 1999. (Status: PROPOSED STANDARD)

[23] http://www.laas.fr/GCAP