Chapter 2

Background on Active Networks

Although our planning approach is applicable for ONA systems, we did the actual implementation of the planner for Panda, an active network middleware system. In this section we will briefly describe the concepts of active networks and Panda, their architectures and problems.

2.1 Active Network Concept

Active networks [Tennenhouse96] represent a new approach to network architecture that incorporates interposed computation. These networks are "active" in two ways: routers and switches within the network can perform computations on user data flowing through them; additionally, users can "program" the network by supplying their own programs to perform these computations. Active networks allow users to deploy new services by tailoring components of the shared infrastructure to suit their requirements. Active technologies have been emerging in the fields of programming languages and operation systems for some years: NetScript [Silva98], Safe-Tcl [Levy98], x-kernel [Hutchinson91], Scout OS [Petersen01], etc.

2.2 Architecture of Active Network Node

The architecture of an active network node consists of three basic components [Calvert98]:

1. Node OS that supports communication channels
2. Execution environment (EE) that exports an API or virtual machine that users can program and control, and also provides an interface through which end-to-end network services can be accessed
2. Active application (AA) that performs user application message handling.

The message created by the sending application goes to the Node OS which determines which of several EEs should handle the message, or directly to EE as in, for example, ANTS [Wetherall98]. The EE in turn may choose to select an AA or other piece of code to handle the message. Once the AA completes the handling, the EE calls the Node OS to request actual physical transmission along some network link. On intermediate nodes, the message can be handled by local AAs that serve the message's data stream. At the destination node, after the EE and AA have done their work, the EE requests the delivery of the message to the destination object, instead of requesting further transmission.

ANTS is an EE for an active node transfer system distributed as a Java-based toolkit for constructing an active network and its applications. We used ANTS as a base for our research. It provides a programming model that allows some kinds of protocol processing to be expressed, a code distribution system for loading new protocols into the network, and node runtime for executing them. ANTS's intended function is to support novel network services for routing, caching, transcoding, combining, filtering, regulating, and otherwise processing packets within the network itself. This includes the notion of “application-specific” protocols where portions of the application processing are “pushed” into strategic nodes of the network. The ANTS design is guided by three goals:

1. The nodes of the network should support a variety of different network protocols being used simultaneously.
2. New protocols should be supported by mutual agreement among interested parties, rather than requiring centralized registration of agreement between parties.
3. New protocols should be deployed dynamically and without the need to take portions of the network “off-line.”

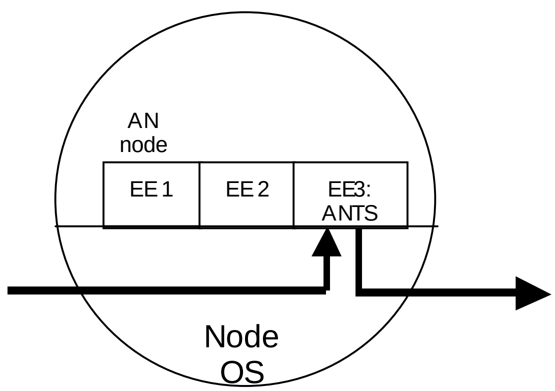

Figure 2.1: Active network node

The combination of a packet and its forwarding routine is called a “capsule”; the forwarding routine is executed at every active node the capsule visits while in the network. The forwarding routine can be cached on active nodes and be executed for a particular data stream. The design of ANTS presumes that applications that use

ANTS must be aware of it. An active network node that hosts ANTS as one of its EEs is presented in Figure 2.1.

2.3 Panda as an Active Network Middleware

Panda [Reiher00] and [Ferreria02] brings the benefits of active networks to legacy applications. Panda automatically traps non-active data streams and converts them into streams of active packets. Panda also creates plans for which active services should be performed at each active network node or switch along the path. A Panda prototype has been used in our lab for over a year. The planning capabilities of the original prototype were extremely primitive. We have implemented an improved prototype that offers better planning support. Panda currently works with the ANTS EE to support UDP applications. The underlying ANTS system is capsule-based and makes no guarantees regarding the delivery of capsules or the order, in which capsules will be received at the destination, much like UDP. Multimedia applications, which tend to use UDP, are good candidates for benefiting from a distributed adaptation system.

Active services in Panda are implemented as adapters that should be deployed according to a plan on nodes designated by the plan. After the plan is activated, the adapters modify all data packets arriving at the node. Adapters increase the cost of the connection, using resources such as CPU cycles, storage, network controls, etc. The deployment of adapters also requires extra time and network bandwidth. Two adapters may have different characteristics even if they handle the same problem for a data stream. For example, one adapter might compress a data stream by converting color images to black-and-white images; another adapter can achieve the same level of compression by reducing the resolution. The choice of a particular adapter, as well as its location can depend on the user's requirements for the data stream.

2.4 Panda Node Architecture

The Panda prototype consists of three basic components (see Figure 2.2).

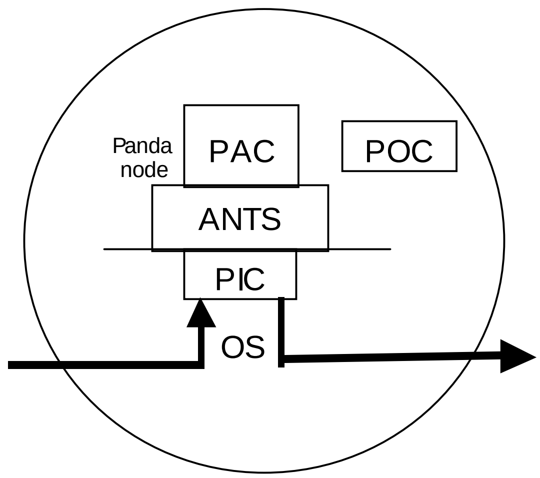

Figure 2.2: A Panda-enabled node

The Panda interception component (PIC) traps messages sent by applications that do not use active networking capabilities. It examines such messages and gives those, which can be assisted by Panda, to the Panda adaptation component (PAC).

The PAC is responsible for planning which adapters to use on behalf of a given data flow and deploying them at proper locations in the network. The PAC is the core of the Panda system. The PAC installs the necessary adapters for a data stream, delivers capsules to the proper adapters, and generally controls the flow of a data stream through Panda nodes. Because these responsibilities heavily overlap the typical behavior of an execution environment, this portion of Panda is tightly coupled to the underlying EE (ANTS) in the current implementation. The planning function of the PAC requires information about conditions in the network.

The third Panda component, the Panda observation component (POC), provides this network conditions information. The POC observes network and node conditions and provides information to the PAC as required for planning. If conditions change drastically, the POC can signal the PAC, which may choose to abandon the existing plan and re-plan. Panda must be deployed at any node where adapters are to be run. Panda planning requires information and cooperation from all Panda nodes traversed by a data flow.

2.5 Problems of Active Networks

AN is an interesting technology, but it adds many challenging problems to the Internet.

The need for planning is one example, but planning is not the only problem of active networks. Other problems from [Zegura98] and other sources are:

- *Need for extra hardware to run adapters*. Hardware trends may make this requirement reasonable, allowing abundant processing resources. Network links are likely to remain a bottleneck for networks of the future, however.
- *Extra cost for networks to take care of every packet.* If the trend of lower computing costs continues, AN technology is likely to be feasible.
- *Extra instability of active networks.* AN increases the level of uncertainty and dynamism in the Internet. Th extra complexities may make the future

lives of network designers harder. For example, the diversity of computing resources and connectivity of network nodes will be much greater in an AN environment. Slow and fast nodes that participate in the same connection and provide adaptation can increase instability in the packet rate. The adapters must be distributed with respect to the execution capability of the nodes. Proper planning can distribute the adapters fairly and may compensate for diversity.

- *Extra dimension of network insecurity.* Today, Internet routers and links are rarely compromised. In an AN, migrating malicious adapters can create problems for the nodes that give them temporary shelter. AN requires more access control for resources and some authentication.
- *Parallel and complex execution of adapters.* The effects of adapters that run concurrently, or that call one another during data processing, add significant complexity to the planning problem. These aspects of planning are not within the scope of this work.
- *Error handling.* AN should provide some mechanism to handle absence or loss of a component, such as the planner or adapter. If the network does not have sufficient resources, some mechanisms should be implemented to redistribute the existing resources according to the priority of connections. One of the sites that participates in the connection should coordinate the connection establishment. It will be notified of the failures of other sites that participated in the connection. If an active node or

adapters fail, proper measures should be provided for the recovery of the connection [Yarvis99A].

- *Resource management and accounting.* Fair use of the resources of networks requires an accounting so that proper users have access to resources and are properly charged for their use.

This dissertation's focus is on the planning problem, but we will also touch on the problems of resource management, security, and fault tolerance.