

## GLOSSARY

### A

#### **Active network**

An active network is a network that allows intermediate routers to perform computations up to the application layer at run time. In addition, users can program the network by injecting their programs into it.

#### **Active node**

An active network node is a network node (i.e., a router) that allows performing computations up to the application layer.

#### **Active packet**

An active packet is the type of packet that carries code or a pointer to code along with regular data. Upon arrival at the node the code will be extracted and run, processing either the packet's contents or performing other tasks as part of a service.

#### **Active server**

Term used in the ANDROID project [ANDROID] to refer to an end-system with a general purposed OS and a full protocol stack that offers an execution environment capable of running user-provided processes that are unrestricted above the transport layer.

#### **Activity diagram**

A special case of a state diagram in which all or most of the states are action states and in which all or most of the transitions are triggered by completion of actions in the source states.

#### **Allocation Schedule**

In the context of this thesis an allocation schedule is a set of data and configuration processes that pre-assign requested resources during a concrete period of time. See also reservation schedule.

#### **Authentication**

Process through which the system can securely verify the identity of a principal.

#### **Authorisation**

Process that decides whether a requested action by a principal shall be allowed or denied based on the principal's access rights.

## C

### **Call Admission Control**

Connection or *call admission control* (CAC) is defined as the set of actions taken by the network during the call set-up phase (or during call renegotiation phase) in order to determine whether a connection request can be accepted or should be rejected (or whether a request for re-allocation can be accommodated).

### **Class diagram**

A diagram that shows a collection of declarative (static) model elements, such as classes, types, and their contents and relationships.

### **Customer**

A customer is an actor that uses a service made available by a service provider.

## D

### **De-enforce**

Remove the managed device configuration caused by a policy enforcement because the policy conditions of that policy are not longer met. However, the policy is still valid; hence, if policy conditions are met again, the managed devices will be configured again accordingly (i.e. the policy will be reenforced).

## E

### **Enforcement**

See policy enforcement

### **Execution Environment**

The execution environment is essentially the active network's programming environment and is often centred on a particular language or model.

The execution environment is responsible for all aspects of the user-to-network interface, including the syntax and semantics of the packets the user submits, the nature of the programming model and the abstractions supported, and addressing and naming facilities.

It runs on top of a node operating system.

## F

### **Functional Domain**

Term that, in the scope of this thesis, refers to the set of functionality supported by a MANBoP's dynamically installable component and represented by the XML Schema document linked with the policies processed by such component. Examples of functional domains could be the active service, QoS or delegation functional domains.

## I

### **Individual Statement**

The Individual Statement concept defined in this thesis refers to the minimum unit in which a policy condition can be split. It is used to create the logic expression representing the policy condition. Each Individual Statement is monitored individually by a Monitoring Meter component.

## L

### **Logic expression**

Boolean expression formed by ANDed or ORed, negated or not, boolean elements. The overall expression value is after calculating the expression with the element values. In MANBoP each element of the logic expression is an individually monitored unit, that is, an Individual Statement.

## M

### **Managed device**

A physical or logical resource that is to be managed but exists independently of its need to be managed. Managed elements include resources within the communications network that provide communications services and systems resources that make use of the communication network.

### **Managed element**

See managed device.

### **Managed entity**

See managed device.

## N

### **Network operator**

A person or company that manages and maintains a large computer network.

### **nodeSet**

The nodeSet concept defined in this thesis refers to a set of underlying topology devices managed by the same Policy Consumer or Monitoring Meter component. This concept is used to find out if a component needed for the processing of a policy that applies to a particular underlying device is already installed. That is, if the component managing the nodeSet where the underlying device is included has been previously installed.

## O

### **Outsourced policy**

An execution model where a policy enforcement device issues a query to delegate a decision for a specific policy event to another component, external to it. "Outsourced policy" is contrasted with "provisioned policy", but they are not mutually exclusive and operational systems may combine the two.

## P

### **Path**

Within this doctorate thesis we use the term 'path' to define a potential physical connection, composed by nodes and links, between one end-point to another in the managed network (see also Route).

### **Policy**

A policy is a set of rules related to a particular purpose. A rule can be expressed as an obligation, permission or a prohibition.

### **Policy action**

Definition of what is to be done to enforce a policy rule, when the conditions of the rule are met. Policy actions may result in the execution of one or more operations to affect and/or configure network traffic and network resources.

### **Policy agent**

See Policy enforcement point

**Policy client**

See Policy enforcement point

**Policy condition**

A representation of the necessary state and/or prerequisites that define whether a policy rule's actions should be performed. This representation need not be completely specified, but may be implicitly provided in an implementation or protocol. When the policy condition(s) associated with a policy rule evaluate to TRUE, then (subject to other considerations such as rule priorities and decision strategies) the rule should be enforced.

**Policy conflict**

Occurs when the actions of two rules (that are both satisfied simultaneously) contradict each other. The entity implementing the policy would not be able to determine which action to perform. The implementers of policy systems must provide conflict detection and avoidance or resolution mechanisms to prevent this situation.

**Policy decision**

Two perspectives of policy decision exist: (a) A process perspective that deals with the evaluation of a policy rule's conditions, and (b) a result perspective that deals with the actions for enforcement, when the conditions of a policy rule are true.

**Policy decision point**

A policy decision point (PDP) is a logical entity that makes policy decisions for itself or for other network elements that request such decisions.

**Policy enforcement**

The execution of a policy decision.

**Policy enforcement point**

A policy enforcement point (PEP) is a logical entity that enforces policy decisions.

**Policy group**

It is a class representing a container, aggregating either policy rules or other policy groups.

**Policy manager**

See policy server.

### **Policy rule**

A basic building block of a policy-based system. It is the binding of a set of actions to a set of conditions - where the conditions are evaluated to determine whether the actions are performed.

### **Policy server**

Term used in marketing and other literature to refer specifically to a PDP, or for any entity that uses/services policy.

### **Programmable network**

A programmable network is a network that at any level of functionality exposes open interfaces through which it facilitates the customisation, extensibility and control of the packet flow.

### **Programmable node**

A programmable node is a network device that provides a computing platform through which downloaded/injected components interact with networks and with each other. To have an open and secure control of network resources, an active node is built upon one programmable network element, e.g., an IP router with an open interface. In general, it consists of a local operating system (e.g., Linux, or freeBSD); one or more distributed processing environment (e.g., TINA-DPE, or mobile agent platform); and system facilities.

### **Provisioned policy**

A policy introduced following the provisioning policy execution model. The provisioning policy is an execution model where network elements are pre-configured, based on policy, prior to processing events. Configuration is pushed to the network device, e.g., based on time of day or at initial booting of the device. The focus of this model is on the distribution of configuration information, and is exemplified by differentiated Services.

## **R**

### **Reservation Schedule**

In this thesis we refer to reservation schedule as a set of data and configuration processes that reserve requested resources during a concrete period of time. Reserved resources must be allocated before they can be actually used. It can be seen as a virtual tunnel (when forwarding resources), or device (when computing resources), with certain quality of service. See also allocation schedule.

**Resource Schedules**

Abstraction that encloses any type of data and configuration processes that pre-assign or reserve resources for a concrete period of time. More specifically, it encloses both allocation schedules and reservation schedules.

**Resource Scheduling**

Processes that permit and deny resource requests as well as create, modify and remove resource schedules accordingly taking into account available resources and other resource schedules.

**Route**

In this doctorate thesis we call ‘route’ a set of resources from path that have been already allocated to one or more flows (see also Path).

**S****Schedules**

See Resource Schedules

**Scheduling**

See Resource Scheduling

**Sequence diagram**

A diagram that shows object interactions arranged in time sequence. In particular, it shows the objects participating in the interaction and the sequence of messages exchanged. Unlike a collaboration diagram, a sequence diagram includes time sequences but does not include object relationships. A sequence diagram can exist in a generic form (describes all possible scenarios) and in an instance form (describes one actual scenario). Sequence diagrams and collaboration diagrams express similar information, but they show it in different ways.

**Service provider**

A service provider composes services from components delivered by a service component provider or obtained from a service component manufacturer, deploys these services in the network and offers them to the consumers. The services provided by a service provider may be end user services or communication services. A service provider may federate with other service providers in order to build more complex services.

## U

### **Underlying device**

In this thesis we use these generic term to refer to any kind of device controlled by a MANBoP instance. More specifically, such a device can be only another MANBoP instance or a managed network element.

### **Unified Modelling Language**

The Unified Modelling Language (UML) is a language for specifying, visualising, constructing, and documenting the artefacts of software systems. It simplifies the complex process of software design, making a "blueprint" for construction.

### **Use case**

A sequence of actions a system performs that yields an observable result of value to a particular actor.

## V

### **Virtual Active Network**

A Virtual Active Network is a Virtual Private Network containing also computational resources for running active services.

### **Virtual Environment**

A Virtual Environment (VE) is a collection of resources (including one or more Execution Environments) owned by a particular SP. VEs are interconnected to form a truly virtual network.

### **Virtual Private Network**

A virtual private network (VPN) is a combination of tunnelling, encryption, authentication, and access control technologies and services used to carry traffic over the Internet, a managed IP network or a provider's backbone while this network remains private and separated from the rest network infrastructure.

## W

### **Working mode**

Set of processes developed by the MANBoP management framework that are classified based on the way these processes are triggered: policy reception, signalling request reception or event reception.