

## AMPS – Design Specification

Version	Author	Date	Comment
1	AP TR	07/9/05	Draft
2	AP TR	14/9/05	2 <sup>nd</sup> draft
3	AP TR	15/9/05	1 <sup>st</sup> Release
4	TR	26/4/06	2 <sup>nd</sup> Release

Introduction.....	1
Assumptions.....	1
Basic Architecture.....	3
Intra-domain sub-system.....	3
Inter-domain sub-system.....	4
Network Information Service sub-system .....	4
Configuration sub-system .....	5

### Introduction

The Advance Multi-domain Provisioning System (AMPS), which until recently was known by its working title of ‘Provisioning System’, implements the recommendations of the SEQUIN project for providing premium IP service across multiple, independent networks. The purpose of AMPS is to regulate the maximum allowed amount of prioritised IP traffic in a given domain, thus ensuring that approved traffic continues to receive a top-quality service (low delay, low packet loss) in the event of network congestion. It does this by evaluating each new request for Premium IP (PIP) against existing, approved PIP requests and the total available network resources, and then only approving those new requests which will not adversely affect any existing reservations.

Importantly, AMPS has a communication module which enables it work with neighbouring peers in order to establish a PIP service that crosses multiple domains.

### Scope and Assumptions

The first version of AMPS has a limited scope and has made some basic assumptions, which will be true when the system is first deployed but may need to be reviewed in future versions.

#### Scope

AMPS has the following pre-requisites and/or limitations.

It is specific to IP and requires that the domain it manages be either over-provisioned or (better) use Differentiated Services (DiffServ).

Since AMPS will not (initially) directly configure routers, there must be a lead time between making a reservation and the relevant NOC(s) applying the required configuration – currently this is 2 working days.

## ***Assumptions***

The network topology is stable and fixed. Future version will cater for planned changes to the network topology. Unplanned changes (i.e. network faults) are allowed for in as much as the amount of PIP permitted on each link is kept artificially low (10%).

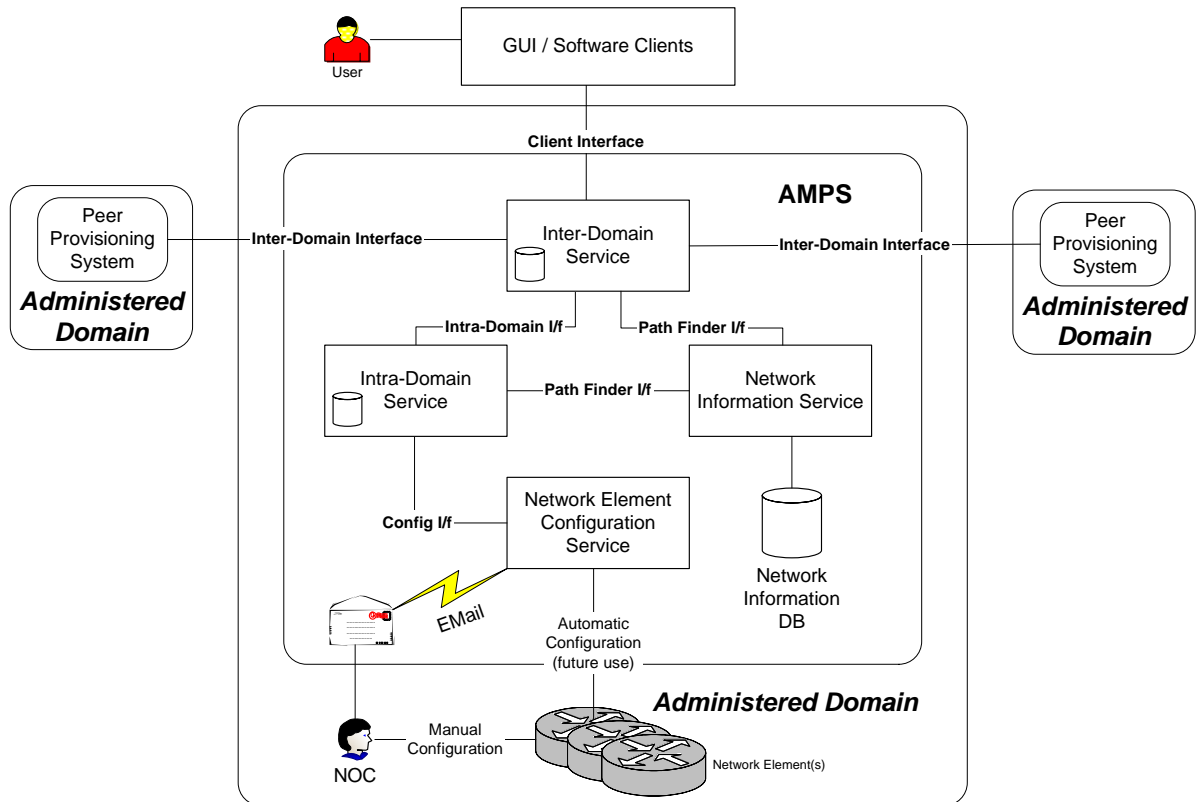
All domains have only one primary link to one another, or in other words, all domains are single-homed (it is possible and acceptable for there to be a primary link and back-up link between domains, but these should have end-points on the same routers). This assumption simplifies the process of determining how a given flow will leave one domain and enter another, and it also means there is no possibility that, as a result of circuit failure and re-routing, a flow will change the point at which it egresses one domain and enters another (except in as much as it may change to a back-up circuit, but this circuit is configured the same as a primary circuit so as far as the Provisioning system is concerned there is no change).

Due to the way traffic flows are inspected, it is difficult to assess overlapping reservations, which is to say reservations which overlap in time and share at least one source and one destination address. In version 1 of the system it is the responsibility of the customer to make sure such reservations are not made. In future versions it is hoped that such overlapping reservations will be handled transparently.

Domains are assumed to use a link-state protocol for their Interior Gateway Protocol (IGP), such as OSPF or ISIS.

## Basic Architecture

AMPS is made up of 4 sub-systems, each of which is a web service (which is to say it communicates with the other sub-systems using XML messages). AMPS is being developed in stages, with the first version being the most basic system, working on assumptions as described above. Future versions will add additional functionality and be more robust (that is, make fewer assumptions).



## Intra-domain sub-system

The Intra-Domain sub-system (also called the Single Domain sub-system) is responsible for its own domain and it keeps a record of all approved PIP reservations that pass through its domain. Requests for new reservations are passed to it by the Inter-Domain sub-system. The reservation will include the following basic parameters: user ID, source end-point, destination end-point, required capacity, required OWD (for future versions, optional), start time, stop time.

The Intra-Domain will first check that the request is valid (that is, the capacity and times requested are within the allowance of the user). This is done by querying its Policy Module. In version 1 there will be a very basic policy, applicable to all users, to the effect that all requests which allow a lead-time of at least 48 hours and are not more than 3 months in duration are acceptable (there is no limit on requested capacity). In version 2 there will be a more sophisticated policy module, where different policies will apply to different groups of users.

If a request is within policy limits then the Intra-Domain sub-system requests the intra-domain path from the NIS. The reservation database is then checked to see if that path has sufficient spare PIP capacity for the requested period. If so, the reservation database is updated with the new info and the Inter-Domain sub-system informed of a successful reservation.

## **Inter-domain sub-system**

The Inter-Domain sub-system (also called the Multi-Domain sub-system) is responsible for receiving messages from and sending messages to external systems as well as handling transactions. These external systems may be either an end-client or the AMPS server in an adjacent domain. Note that an AMPS server cannot communicate directly with an AMPS server in a domain to which it is not directly connected, but such messages can be relayed to a distant AMSP server via intermediate AMPS servers.

When the Inter-Domain subsystem receives a request for a new PIP reservation it immediately responds to the client with a service id and status as 'pending'. Simultaneously, acting as a transaction manager, it starts a new transaction. It then contacts the NIS using the pathfinder interface to check which is the next domain (if any) and what is the current domain's egress interface and next domain's ingress interface. It then sends the reservation request to the Intra-Domain sub-system. Once the Intra-Domain sub-system reports that reservation request is successful then the request is forwarded to the Provisioning System of the next domain along the path. Assuming that the reservation is successful in the next domain this chain process continues until the last domain (having an AMPS server) along the path is reached. Once the request is successful in the last domain; a message indicating that the request is successful is relayed back to the originating domain. This end-to-end chaining process completes the transaction and the status is updated as 'accepted'.

If any domain in the path rejects the request, a message indicating the request is unsuccessful is relayed back to the originating domain. Being the transaction manager it is then responsible to send a rollback message to all domains along the path where a reservation may have been made to undo (cancel) the reservation. Once all domains have rolled back their reservation the transaction manager is notified. This completes the transaction and status is changed to 'rejected', and a reason given for rejection.

The transaction manager will keep sending the rollback until it receives the acknowledgement (or will notify some human administrator).

In case any service along the path does not respond a timeout mechanism will be triggered on the transaction manager and a rollback process will be started automatically. Once all domains have rolled back their reservations the transaction manager is notified. This completes the transaction and status is changed to 'rejected', and a reason given for rejection.

The client can periodically query the status of the request. Once the transaction is complete the status will be updated from 'pending' to 'accepted' or 'rejected'.

If the reservation is submitted to a domain (home domain) but it starts in another (source) domain, the request will be forwarded to the start domain. Even though the

reservation starts in the source domain, the home domain will act as the transaction manager. In all other respects AMPS behaves as described above.

## **Network Information Service sub-system**

The Network Information Service sub-system uses the Network Information database to calculate what route a given flow will take across the network. To do this the database must have a record of all the links in the network, and it must know each links metric (NIS currently only works for networks using link-state IGPs), and the NIS must be told (by a client, using the NIS's PathFinder interface) the start and end points of the flow. The NIS will then return to the client (using the same PathFinder interface), the link by link path the specified flow will take.

Various methods may be used to populate and maintain the topology, such as, for example, collecting and parsing router configurations. The NIS has the following main tables:

- a. Routers
- b. Physical interfaces
- c. Layer 2 logical interfaces
- d. Layer 3 logical interfaces
- e. Layer 3 links
- f. Adjacent domains

For possible future use there are also tables for LSPs and a routing table. In version 1 and 2 of AMPS the population of NIS is done using non-robust, bespoke (network dependent) methods. This is acceptable for the initial, limited pilot phase, but is wholly inappropriate for a production-quality system. Before a more general deployment of AMPS is possible, the NIS must be enhanced so that it can be populated ( and subsequently verified) using standard methods (ideally SNMP) and a minimum amount of configuration on the part of the operator.

## **Configuration sub-system**

The configuration sub-system (under development) takes the information associated with an accepted reservation (ingress interface, source and destination IP addresses and capacity) and creates a router configuration written in a generic language. A vendor-specific driver module is then used to translate this generic language into a specific router configuration (e.g. JUNOS or IOS). In version 1 of the Provisioning System NOC staff will use the generated config to manually edit the routers' configurations (currently only JUNOS configurations are generated – Cisco configs would need to be manually derived from the JUNOS text). In future versions it will be possible for the Configuration subsystem itself to apply and remove the configs directly, at the correct time. This of course will reduce the required lead-time.