Title:	Document Version:
Deliverable D4.1 **Head End Router HW and SW Specification**	1.5

Project Number:	Project Acronym:	Project Title:
IST-2001-37613	6POWER	IPv6, QoS & Power Line Integration

Contractual Delivery Date:	Actual Delivery Date:	Deliverable Type* - Security**:
31/03/2003	23/06/2003	R – PP

* Type: P - Prototype, R - Report, D - Demonstrator, O - Other
** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Responsible and Editor/Author:	Organization:	Contributing WP:
Jean-Mickael Guérin	6WIND	WP4

Authors (organizations):

Jordi Palet (Consulintel), Chano Gómez (DS2).

Abstract:

This deliverable presents hardware and software specification of Head End, the IP router equipment that interconnects the IP PLC network to external Internet.

Keywords:

IPv6, Multicast, PLC, QoS.

Revision History

The following table describes the main changes done in this document since its creation.

Revision	Date	Description	Author (Organization)
v1.0	10/03/2003	Document creation	Jean-Mickael Guérin (6WIND)
v1.1	28/05/2003	Add VLAN section	Jean-Mickael Guérin (6WIND)
v1.2	09/06/2003	Modified Multicast, QoS and VLAN sections	Chano Gómez (DS2)
v1.3	16/06/2003	Add abstract, executive summary and conclusion	Jean-Mickael Guerin (6WIND)
v1.4	23/06/2003	Updated to last template	Jean-Mickael Guerin (6WIND)
v1.5	23/06/2003	Final Review	Jordi Palet (Consulintel)

Executive Summary

The Head End (HE) is the equipment to interconnect the IP PLC network to the external Internet world.

This deliverable gathers hardware and software specification to build the Head End Router that includes basic and mandatory features for the project, according the results of WP2 “Integration of IPv6 Advanced Services over Power Lines” and WP3 “Network Architecture Design and Implementation”.

These include IPv6 autoconfiguration, quality of service and IPv6 multicast.

Table of Contents

Table of Figures

1. INTRODUCTION

WP4 aims at designing and developing the devices able to support a large IPv6 deployment over PLC. These developments include the Head End, the Home Gateway, a Set-Top Box and some adaptations for end devices. First deliverable D4.1 focus on hardware and software specifications of the Head End.

The Head End (HE) is the equipment to interconnect the IP PLC network to the external Internet world. HE can be functionally considered as a router with a dedicated modem interface card able to broadcast data on the downlink channel and to control the access on the uplink channel. The equipment to be used for the Head End Router development will be an IP access router.

Next figure gives an overview of IP over PLC architecture.

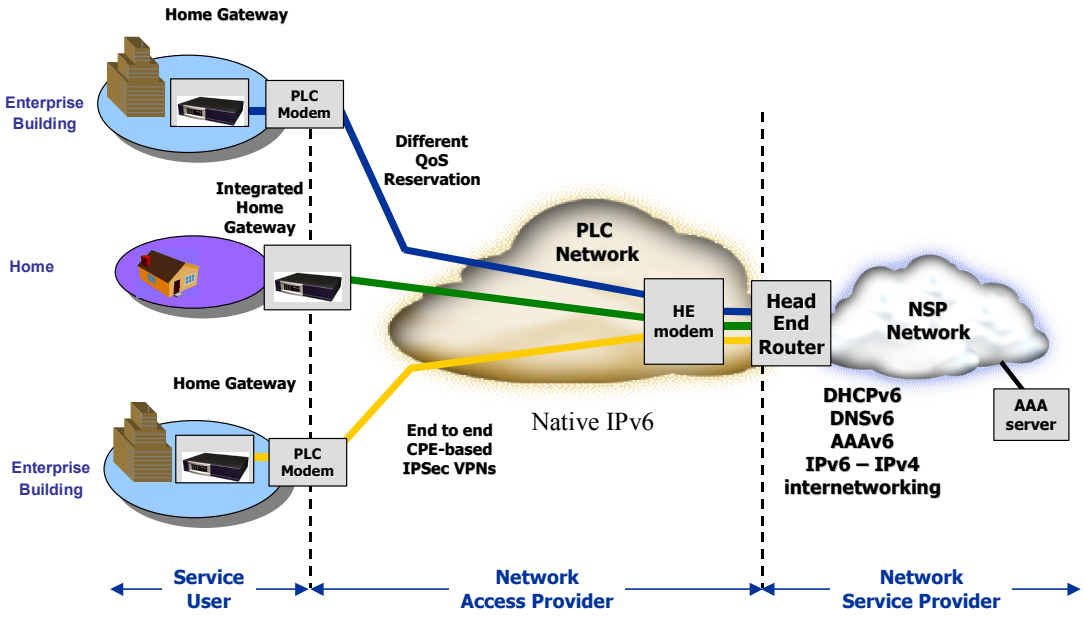


Figure 1-1: IP over PLC Architecture

2. HARDWARE SPECIFICATION

The IP router is based on the 6WINDGate 6200 hardware. The PLC modem is based on DS2 hardware. The router and the modem are interconnected through a Fast Ethernet cable.

2.1 Hardware Resources

Hardware	CPU	RAM
IP router	1 GHz	256 Mbytes
PLC modem	200 MHz	32 Mbytes

Figure 2-1: HW Resources

2.2 Hardware Interfaces

The PLC modem is shipped with DS2 chipset for PLC.

The IP router support 10/100 Fast Ethernet cards, 155 Mbps ATM interface.

Depending on the evolution of SixOS during the project, a Gigabit ethernet card could be available.

2.3 Hardware Performance

The PLC modem depends on the PLC chip set being used, that could be improved during the project lifetime.

High performance is needed to bring full capabilities to deliver high-end services. PLC modem support up to 45 Mbps. The IP router supports:

- Up to 2000 VPN tunnels.
- Up to 95 Mbps throughput.
- Up to 2000 simultaneous QoS flows.
- Up to 70 Mbps DES and 30 Mbps 3DES encryption.

3. SOFTWARE SPECIFICATION

The software will be mainly developed in C.

C++ for application development can be an option but we need to keep code size and resident memory of processes within reasonable limits to allow industrial applications in the future.

3.1 Software Architecture

3.1.1 Integrated Head End Architecture

According to deliverables [1] [2], the PLC driver developed by WP2 provides the encapsulation of ethernet frames into PLC frames. This leads to a generic Ethernet communication channel between the IP router entity and PLC modem entity.

3.1.2 Proof-of-Concept Prototype

Using a physical Ethernet link, the two logical entities can be split into a proof-of-concept prototype made of an IP router and a PLC bridge. Programming PLC modem from the IP router will be transparent about whether PLC modem is inside the IP router or outside in an independent PLC modem.

It brings the benefit of concurrent development of IP advanced functions and of new PLC capabilities.

3.2 Software Modules

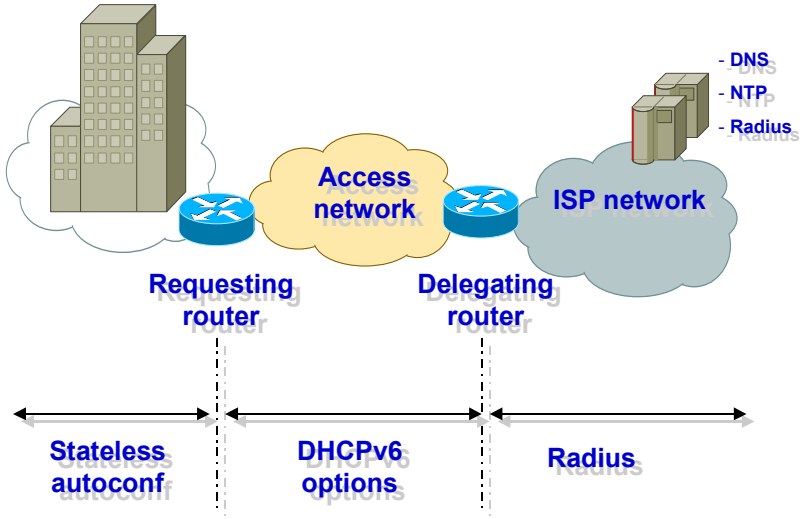
3.2.1 Autoconfiguration

3.2.1.1 IPv6 Stateless Autoconfiguration

IPv6 specification describes the stateless address configuration as a possible way to configure IPv6 addresses [3] [4] [5] [6]. This method relies on the IPv6 address structure. IPv6 addresses are made of a prefix network and of an interface identifier. Network prefixes are advertised on every link by routers while the interface identifier is built locally in the host from the MAC address of the card network. From these elements, every host can build its own IPv6 addresses.

3.2.1.2 DHCPv6 Prefix Delegation

DHCPv6 [8] [9] provides a mechanism to automate delegation of prefixes and to distribute IPv6 domain name servers [10]. The 6WINDGate implements a DHCPv6 client to get dynamically delegated prefix from a DHCPv6 server. The delegating router will be typically present at the point of presence in ISP network, and does not require knowledge about the topology of the links attached to the CPEs. A basic use case is assignment a prefix /48 to CPE that assign a subnet /64 from this delegated prefix to its LAN interfaces, and begin sending router advertisements.



The DHCPv6 client complies with the following IETF drafts:

- draft-ietf-dhc-dhcpv6-28 (Dynamic Host Configuration Protocol for IPv6).
- draft-ietf-dhc-dhcpv6-opt-prefix-delegation-02.txt (IPv6 Prefix Options for DHCPv6).

The development and update will need to take into account new revision of draft, specifically regarding to DHCPv6 prefix delegation option.

3.2.1.3 Domain Name Servers Advertisement

Thanks to the DHCPv6 DNS Configuration option [11], the delegating router can provide IPv6 addresses of name servers.

The requesting router then can act as DNS proxy for IPv6 (and may be IPv4) hosts. Depending on the way DNS addresses are discovered by IPv6 hosts, hosts can rely on CPE to perform name resolution.

3.2.1.4 Interaction with Radius Server

When the DHCPv6 server receives a solicitation from a client, it checks if he has a configuration for this client, watching to the configuration file parsing result. At this point it uses the client duid for identification.

If it finds one, it sends a reply to the client, containing the prefix he found in the configuration file.

If he can't find any, tries to authenticate the client using the radius server [7]. It checks among client duid, client MAC address, or client link-local address if a binding exists in the hosts file.

When a binding is matched the DHCPv6 server gets the associated login and password, and tries to authenticate to the radius server. If the authentication is valid, a new client configuration

(host_conf) is created using radius-acquired information, and added to the hosts configuration list (to avoid another authentication to the radius server, we keep this client configuration in memory). A new binding is added, and the server replies to the client, including the radius-acquired prefix. The prefix lifetime of this prefix is the default prefix lifetime, initialized to infinite but can be overwritten in the global configuration file.

3.2.1.5 Interaction with VLAN

VLAN stands for Virtual LAN and is useful to differentiate customers' traffic over a shared ethernet networks – ethernet frames are extended with a 16 bits tag, and a unique tag is assigned to each customer. According to the deliverable IPv6 over PLC, a logical Ethernet layer is used to carry data. This makes a PLC network an ethernet-like network on which use of VLAN can be benefit. Since the DHCPv6 server is localized in the Head End, it could take advantage of getting the VLAN tag from DHCPv6 requests from CPEs, and then selecting the adequate IPv6 prefix according to VLAN tag.

This functionality can be used in several ways, depending on the needs of the network operator: one possible scenario would reserving one VLAN for end-users' CPEs, another VLAN for network infrastructure equipment and another for VoIP traffic.

Another possibility would be connecting users in VLAN A to ISP A and users in VLAN to ISP B. This would be useful in deployment scenarios in which the electricity utility plays the role of a carriers' carrier, with third-party ISP providing Internet and Telephony connectivity.

3.2.2 Routing

IP router entity already supports dynamic routing protocols like RIPng, OSPFv3, BGP. There is no special development on routing protocols, but prefix delegation mechanism is tied to routing.

We must be able to inject delegated prefixes assigned to CPE into routing protocols. Delegated prefixes will be seen as specific routes tagged with a new tag 'DEP' among existent tags (RIP, OSPF, etc.). This will permit to configure routing protocols with the "redistribute" command,- using "redistribute dep" will inject any assigned prefixes to customers into dynamic routing protocols.

3.2.3 QoS

QoS implemenation is based on DIFFSERV specifications.

3.2.3.1 Diffserv Architectural Model Overview

The DiffServ architecture is described in "RFC 2475 Architecture for Differentiated Services" [16].

The DiffServ architecture goal is to implement a scalable service differentiation in the Internet. A "service" defines some significant characteristics of packet transmission in one direction across a set of paths within a network. These characteristics may be specified in terms of throughput, delay, jitter, and/or loss, or may otherwise be specified in terms of some relative priority of access to network resources.

The DiffServ architecture is composed of a number of elements implemented in network nodes, including:

- Per-hop forwarding behaviors (PHB) [14].
- Packet classification functions.

Traffic conditioning functions including:

- Metering.
- Marking.
- Shaping.
- Policing.

This architecture achieves scalability by implementing complex classification and conditioning functions at the network edges only, and by applying per-hop behaviors to traffic aggregates that have been appropriately marked using the Differentiated Services field [15] in the IP packet header.

The architecture maintains a distinction between:

- The service provided to a traffic aggregate.
- The conditioning functions and PHB used to realize services.
- The DS field value (DS code-point) used to mark packets to select a PHB.
- The particular node implementation mechanisms that realize a PHB.

This architecture only provides service differentiation in one direction of traffic flow and is therefore asymmetric.

3.2.3.2 Diffserv Architectural Model

The DiffServ architecture is based on a simple model where traffic entering a network is classified and possibly conditioned at the network edges, and assigned to different behavior aggregates. Each behavior aggregate is identified by a single DS code-point. Within the core of the network, packets are forwarded according to the PHB associated with the DS code-point.

DiffServ Domain

A DiffServ domain is a contiguous set of DiffServ nodes that operate with a common service provisioning policy and set of PHB groups implemented on each node. A DiffServ domain has a well-defined boundary consisting of DiffServ boundary nodes that classify and possibly condition ingress traffic to ensure that packets that transit the domain are appropriately marked to select a PHB from one of the PHB groups supported within the domain. Nodes within the DiffServ domain select the forwarding behavior for packets based on their DS code-point, mapping that value to one of the supported PHB using either the recommended code-point to PHB mapping or a locally customized mapping.

Inclusion of non-DiffServ-compliant nodes within a DiffServ domain may result in unpredictable performance and may impede the ability to satisfy Service Level Agreements.

A DiffServ domain normally consists of one or more networks under the same administration, for example, an organization Intranet or an ISP network. The domain administration is responsible for ensuring that adequate resources are provisioned and/or reserved to support the Service Level Agreements offered by the domain.

DiffServ Region

The DiffServ region is a set of one or more contiguous DiffServ domains. DiffServ regions are capable of supporting differentiated services along paths that span the domains within the region. The DiffServ domains in the DiffServ region may support different PHB groups internally and different code-point to PHB mappings.

Traffic Classification and Conditioning

The packet classification policy identifies the subset of traffic that may receive a differentiated service by being conditioned and/or mapped to one or more behavior aggregates (by DS code-point remarking) within the DiffServ domain.

Packet classifiers select packets in a traffic stream based on the content of some portion of the packet header.

Two types of classifiers are defined. The BA (Behavior Aggregate) classifier classifies packets based on the DS code-point only. The MF (Multi-Field) classifier selects packets based on the value of a combination of one or more header fields, such as source address, destination address, DS field, protocol ID, source port and destination port numbers, and other information such as incoming interface.

Traffic conditioning performs metering, shaping, policing and/or remarking in order to ensure that the traffic entering the DiffServ domain conforms to the rules specified by the service provider. The extent of traffic conditioning required is dependent on the specifics of the service offering, and may range from simple code-point re-marking to complex policing and shaping operations.

Traffic conditioners are usually located within DiffServ ingress and egress boundary nodes, but may also be located in nodes within the interior of a DiffServ domain, or within a non-DiffServ-capable domain.

Per-hop Behaviors (PHB)

A PHB is a description of the externally observable forwarding behavior of a DiffServ node applied to a particular DiffServ behavior aggregate.

PHB are implemented in nodes by means of some buffer management and packet scheduling mechanisms. PHB are defined in terms of behavior characteristics relevant to service provisioning policies, and not in terms of specific implementation mechanisms. In general, a variety of implementation mechanisms may be suitable for implementing a particular PHB group. Furthermore, it is likely that more than one PHB group may be implemented on a node and utilized within a domain.

DS Field Definition

A replacement header field, called the DS field, has been defined. It is intended to supersede the existing definitions of the IPv4 TOS field and the IPv6 Traffic Class field. The DS field is described in RFC 2474.[15]

Six bits of the DS field are used as a code-point (DSCP) to select the PHB a packet experiences at each node. A two-bit currently unused (CU) field is reserved. DiffServ-compliant nodes, when determining the PHB, to apply to a received packet, ignore values of CU bits.

The DS field structure is presented below:

- DSCP: Differentiated Services Code-Point.
- CU: Currently Unused.

In a DSCP value notation 'xxxxxx' (where 'x' may equal '0' or '1'), the left-most bit signifies bit 0 of the DS field (as shown above), and the right-most bit signifies bit 5.

DiffServ nodes select PHBs by matching against the entire 6-bit DSCP field, e.g., by treating the value of the field as a table index that is used to select a particular packet handling mechanism, which has been implemented in that device. The value of the CU field must be ignored by PHB selection. The DSCP field is defined as an unstructured field to facilitate the definition of future PHBs.

The mapping of DSCP to PHBs must be configurable. A DiffServ compliant node must support the logical equivalent of a configurable mapping table from code-points to PHB. PHB specifications must include a recommended default code-point that has to be unique for code-points in the standard space.

The structure of the DS field shown above is incompatible with the existing definition of the IPv4 TOS field in RFC 791. The presumption is that DiffServ domains protect themselves by deploying re-marking boundary nodes, as should networks do, using the RFC 791 precedence designations.

Nodes may rewrite the DS field as required to provide a desired local or end-to-end service. Specifications of DS field translations at DiffServ boundaries are the subject of Service Level Agreements between providers and users. Standardized PHBs allow providers to build their services from a well-known set of packet forwarding treatments that can be expected to be present in the equipment of many vendors.

Historical Code-Points Definitions

DS fields have a limited backwards compatibility with current practice. Backwards compatibility is addressed in two ways. First, there are PHBs that are already in widespread use (e.g., those satisfying the IPv4 Precedence queuing requirements specified in RFC 1812), and their continued use in DiffServ-compliant nodes is permitted. Second, there are some code-points that correspond to historical use of the IP Precedence field and these code-points are reserved to map to PHB, though the specific DiffServ PHB mapped to by those code-points may have additional specifications.

No attempt is made to maintain backwards compatibility with TOS bits of the IPv4 TOS octet, as defined in RFC 791.

Default PHB

A "default" PHB must be available in a DiffServ compliant node. This is the common, best-effort forwarding behavior available in existing routers. When no other agreements are in place, it is assumed that packets belong to this aggregate. This permits senders that are not “DiffServ-aware” to continue to use the network in the same manner as today.

The recommended code-point for the Default PHB is the bit pattern '000000'.

Today and Future IP Precedence Field Use

The IETF wishes to maintain some form of backwards compatibility with present uses of the IP Precedence Field: bits 0-2 of the IPv4 TOS field.

Routers exist that use the IP Precedence field to select different PHB treatments in a similar way to the use proposed here for the DSCP field. Thus, simple prototype DiffServ architecture can be quickly deployed by appropriately configuring these routers. Furthermore, IP systems today understand the location of the IP Precedence field, thus, if these bits are used in a similar manner as DiffServ-compliant equipment is deployed, significant failures are not likely to occur during early deployment. In other words, strict DS-compliance does not need to be ubiquitous even within a single service provider's network if bits 0-2 of the DSCP field are employed in a manner similar to, or subsuming, the deployed uses of the IP Precedence field.

The IETF has defined code-points 'xxx000' as the Class Selector code-points, where PHB selected by these code-points must meet the Class Selector PHB Requirements described below. This is done to preserve a useful level of backwards compatibility with current uses of the IP Precedence field in the Internet, without unduly limiting future flexibility.

In addition, code-point '000000' is used as the Default PHB value for the Internet and, as such, is not configurable. The remaining non-zero Class Selector code-points are configurable only to the extent that they map to PHBs that meet the requirements below.

PHB groups whose specifications satisfy the following requirements are referred to as Class Selector Compliant PHBs:

- A Class Selector code-point with a larger numerical value than another Class Selector code-point has a higher relative order while a Class Selector code-point with a smaller numerical value than another Class Selector code-point is said to have a lower relative order. The set of PHBs mapped by the Class Selector code-points must yield at least two independently forwarded classes of traffic. PHBs selected by a Class Selector code-point should give packets a probability of timely forwarding that is not lower than that given to packets marked with a Class Selector code-point of lower relative order, under reasonable operating conditions and traffic loads. A discarded packet is considered to be an extreme case of untimely forwarding. In addition, PHB selected by code-points '11x000' must give packets a preferential forwarding treatment by comparison to the PHB selected by code-point '000000', in order to preserve the common usage of IP Precedence values '110' and '111' for routing traffic.
- Further, PHBs selected by distinct Class Selector code-points should be independently forwarded; that is, packets marked with different Class Selector code-points may be re-ordered. A network node may enforce limits on the amount of the node's resources that can be utilized by each of these PHBs.
- Class Selector PHB Requirements on code-point '000000' are compatible with those listed for the Default PHB.

Expedited Forwarding PHB (EF PHB)

EF PHB is described in RFC 3246 [18].

EF PHB can be used to build a low loss, low latency, low jitter, guaranteed bandwidth, end-to-end service through DiffServ domains. Such a service appears to the endpoints like a point-to-point connection or a "virtual leased line".

Loss, latency and jitter are all due to the queues traffic experiences while transiting the network. Therefore providing low loss, latency and jitter for some traffic aggregate means ensuring that the aggregate sees no (or very small) queues. Queues arise when (short-term) traffic arrival rate exceeds departure rate at some node. Thus a service that ensures no queues for some aggregate is equivalent to bounding rates such that, at every transit node, the aggregate's maximum arrival rate is less than that aggregate's minimum departure rate.

Creating such a service represents two steps:

- Configuring nodes so that the aggregate has a well-defined minimum departure rate. ("Well-defined" means independent of the dynamic state of the node, in particular independent of the intensity of other traffic at the node).
- Conditioning the aggregate (via policing and shaping) so that its arrival rate at any node is always less than that node's configured minimum departure rate.

EF PHB provides the first part of the service. Network boundary traffic conditioners described in RFC 2475 provide the second part.

EF PHB is not a mandatory part of the DiffServ architecture, i.e., a node is not required to implement EF PHB in order to be considered DiffServ-compliant.

EF PHB is defined as a forwarding treatment for a particular DiffServ aggregate where the departure rate of the aggregate's packets from any DiffServ node must equal or exceed a configurable rate. EF traffic should receive this rate independently of the intensity of any other traffic attempting to transit the node. It should average at least the configured rate when measured over any time interval equal to or longer than the time it takes to send an output link MTU sized packet at the configured rate.

The configured minimum rate must be settable by a network administrator (using whatever mechanism the node supports for non-volatile configuration).

If EF PHB is implemented by a mechanism that allows unlimited pre-emption of other traffic (e.g., a priority queue), the implementation must include some means to limit the damage EF traffic could inflict on other traffic (e.g., a token bucket rate limiter). Traffic that exceeds this limit must be discarded. This maximum EF rate, and burst size if appropriate, must be settable by a network administrator (using whatever mechanism the node supports for non-volatile configuration). Minimum and maximum rates may be the same and configured by a single parameter.

Code-point 101110 is recommended for EF PHB.

Assured Forwarding PHB (AF PHB)

AF PHB is described in RFC 2597 [17].

There is a demand to provide assured forwarding of IP packets over the Internet. In a typical application, a company uses the Internet to interconnect its geographically distributed sites and wants an assurance that IP packets within this intranet are forwarded with high probability as long as the aggregate traffic from each site does not exceed the subscribed information rate (profile). It is desirable that a site may exceed the subscribed profile with the understanding that the excess traffic is not delivered with as high probability as the traffic that is within the profile. It is also important that the network does not reorder packets that belong to the same microflow, no matter if they are in or out of the profile.

An AF PHB group is a means for a DiffServ domain provider to offer different levels of forwarding assurances for IP packets received from a DiffServ domain customer.

Four AF classes are defined, where each AF class in each DiffServ node allocates a certain amount of forwarding resources (buffer space and bandwidth). The customer or the DiffServ domain provider to one or more of these AF classes, according to the services that the customer has subscribed to, assigns IP packets that wish to use the services provided by the AF PHB GROUP.

Within each AF class, IP packets are marked (again by the customer or the DiffServ domain provider) with one of three possible drop precedence values. In case of congestion, the drop precedence of a packet determines the relative importance of the packet within the AF class. A congested DiffServ node tries to protect packets with a lower drop precedence value from being lost, by preferably discarding packets with a higher drop precedence value.

In a DiffServ node, the forwarding assurance level of an IP packet thus depends on:

- How many forwarding resources have been allocated to the AF class the packet belongs to.
- The current load of the AF class, and in case of congestion, within the class.
- The drop precedence of the packet.

The correct behavior of the system in the DiffServ domain relies on the traffic conditioning actions carried by edge devices at the domain boundaries. If edge devices make sure that the core nodes are moderately loaded by packets with the lowest drop precedence's, the AF class can then offer a high level of assurance for packets that are within the subscribed profile as well as assurance for the excess traffic of the lowest levels of drop precedence.

An AF PHB group provides forwarding of IP packets in N independent AF classes. Within each AF class, an IP packet is assigned one of M different levels of drop precedence. An IP packet that belongs to an AF class i and has drop precedence j is marked with the AF code-point AFij, where 1 <= i <= N and 1 <= j <= M.

Currently, four classes (N=4) with three levels of drop precedence in each class (M=3) are defined for general use. More AF classes or levels of drop precedence may be defined for local use.

A DiffServ node should implement all four general use AF classes. Packets in one AF class must be forwarded independently from packets in another AF class, i.e., a DiffServ node must not aggregate two or more AF classes together.

A DiffServ node must allocate a configurable, minimum amount of forwarding resources (buffer space and bandwidth) to each implemented AF class. Each class should be serviced so as to achieve the configured service rate (bandwidth) over both small and large time scales.

An AF class may also be configurable to receive more forwarding resources than the minimum when excess resources are available either from other AF classes or from other PHB groups.

Within an AF class, a DiffServ node must make a decision according to the drop precedence values on which packets will be forwarded and which packets will be discarded. Note that this requirement can be fulfilled without needing to neither dequeue nor discard already-queued packets.

Within each AF class, a DiffServ node must accept all three-drop precedence code-points and yield at least two different levels of loss probability. In some networks, particularly in enterprise networks, where transient congestion is a rare and brief occurrence, it may be reasonable for a DiffServ node to implement only two different levels of loss probability per AF class. While this may suffice for some networks, three different levels of loss probability should be supported in DiffServ domains where congestion is a common occurrence.

If a DiffServ node only implements two different levels of loss probability for an AF class x, the code-point AFx1 must yield the lower loss probability and the code-points AFx2 and AFx3 must yield the higher loss probability.

A DiffServ node must not reorder AF packets of the same micro flow when these AF packets belong to the same AF class regardless of their drop precedence. There are no quantifiable timing requirements (delay or delay variation) associated with the forwarding of AF packets.

The AF PHB group may be used to implement both end-to-end and domain edge-to-domain edge services.

Recommended code-points for the four general use AF classes are given below. These code-points do not overlap with any other general use PHB groups.

The AF code-points recommended values are summarized in the following table:

Drop precedence	Class 1	Class 2	Class 3	Class 4
Low	001010 (AF11)	010010 (AF21)	011010 (AF31)	100010 (AF41)
Medium	001100 (AF12)	010100 (AF22)	011100 (AF32)	100100 (AF42)
High	001110 (AF13)	010110 (AF23)	011110 (AF33)	100110 (AF43)

Figure 3-1: Drop Precedence Table

Drop Policies

Several drop policies may be applied to a class of traffic.

- Taildrop.

When using the taildrop policy, incoming packets are dropped when the queue is full.

- Taildrop with Injection.

Taildrop with injection in best-effort queue means that when the queue is full, incoming packets are put in best-effort queue. When the best-effort queue is full itself, packets are dropped.

- RED.

The Random Early Drop (RED) policy randomly drops or marks packets when the average queue length exceeds a minimum threshold (qmin). The drop probability increases with increasing average queue length up to a maximal dropping probability (pmax). When the average queue length reaches an upper threshold (qmax), all packets are dropped. The average queue length is calculated with an Exponential Weighted Moving Average (EWMA) defined with the averaging parameter P:

$Qt = Qt-1 + P . (Mt - Qt-1)$

Mt: Measured queue length at time t

Qt: Average queue length at time t

P: Averaging parameter

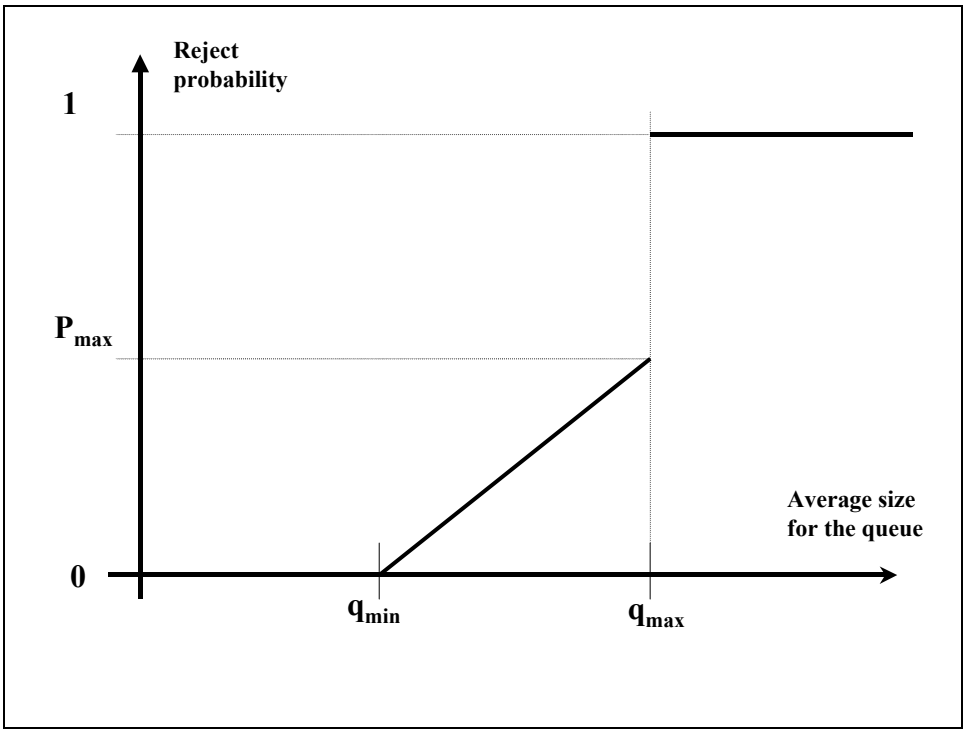


Figure 3-2: Reject Probability Evolution

WRED

The algorithm used by 6WIND software to implement an AF class is a WRED (Weighted Random Early Drop) algorithm.

A RED algorithm is implemented for each drop precedence (refer to Figure 3-1) with different qmin, qmax and Pmax parameters.fd.

3.2.3.3 QoS Parameters and Mapping

Mapping of IP-layer QoS parameters and PLC-ñayer QoS parameters is done at the PLC driver. The mapping is done as described in Deliverables D2.2 and D2.3: The content of the DSCP field is used for detecting traffic with QoS requirements (in our application, VoIP) and for reconfiguring resource allocation policies accordingly.

3.2.4 IPv6 Multicast

3.2.4.1 Multicast Parameters for PLC Modem

Multicast optimization will be done at PLC layer thanks to PLC modem. As explained, in deliverables D2.2 and D2.3, the current specification of the IPv6-over-PLC interface does not support native multicast; multicast frames are retransmitted to every other node in the network using unicast transmission.

Anyway, this is transparent to the IP router, which does not know how the PLC modem handles multicast transmission.

Once multicast support is added in the next revision of the IPv6-over-PLC specification, this design will transparently make use of it.

3.2.4.2 Multicast Routing

The Head End will connect IPv6 Multicast networks by running dynamic multicast protocol.

Multicast is based on the concept of a group. An arbitrary group of receivers expresses an interest in receiving a particular data stream. This group does not have any physical or geographical boundaries; the hosts can be located anywhere on the Internet. Applications that take advantage of multicast include videoconferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news.

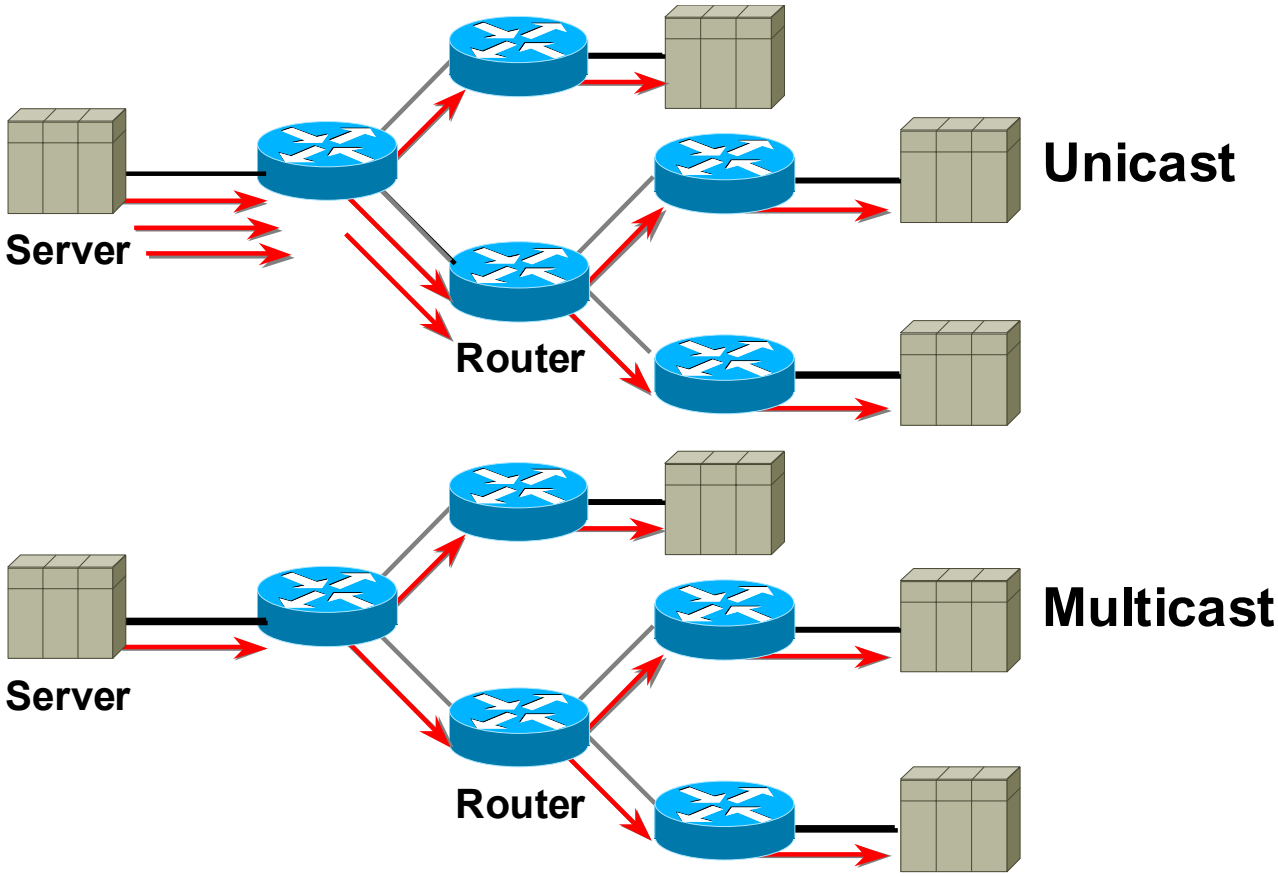


Figure 3-3: Unicast versus Multicast Routing

Hosts that are interested in receiving data flowing to a particular group must join the group using MLD (Multicast Listener Discovery). Hosts must be a member of the group to receive the data stream. The Multicast Listener Discovery (MLD) protocol manages the membership of hosts and routers in multicast groups. IPv6 multicast routers use MLD to learn, for each of their attached physical networks, which groups have interested listeners. Each router maintains a list of host multicast addresses that have listeners for each subnet, as well as a timer for each address. However, the router does not need to know the address of the listeners, just the address of the hosts. The router provides addresses to the multicast routing protocol it uses; this ensures that

multicast packets are delivered to all subnets where there are interested listeners. In this way, MLD is used as the transport for multicast Protocol Independent Multicast (PIM).

In multicast routing, the source is sending traffic to an arbitrary group of hosts represented by a multicast group address. IPv6 multicast addresses are reserved and assigned from the Format Prefix 1111 1111 (0xFF).

The multicast router must determine which direction is upstream (toward the source) and which direction (or directions) is downstream. If there are multiple downstream paths, the router replicates the packet and forwards the traffic down the appropriate downstream paths—which is not necessarily all paths. This concept of forwarding multicast traffic away from the source, rather than to the receiver, is called *reverse path forwarding*.

PIM gets its name from the fact that it is IP routing protocol-independent. PIM can leverage whichever unicast routing protocols are used to populate the unicast routing table, including EIGRP, OSPF, BGP, or static routes. PIM uses this unicast routing information to perform the multicast forwarding function, so it is IP protocol-independent. Although PIM is called a multicast routing protocol, it actually uses the unicast routing table to perform the reverse path forwarding (RPF) check function instead of building up a completely independent multicast routing table. PIM does not send and receive multicast routing updates between routers like other routing protocols do. There are three different orientations of flooding process.

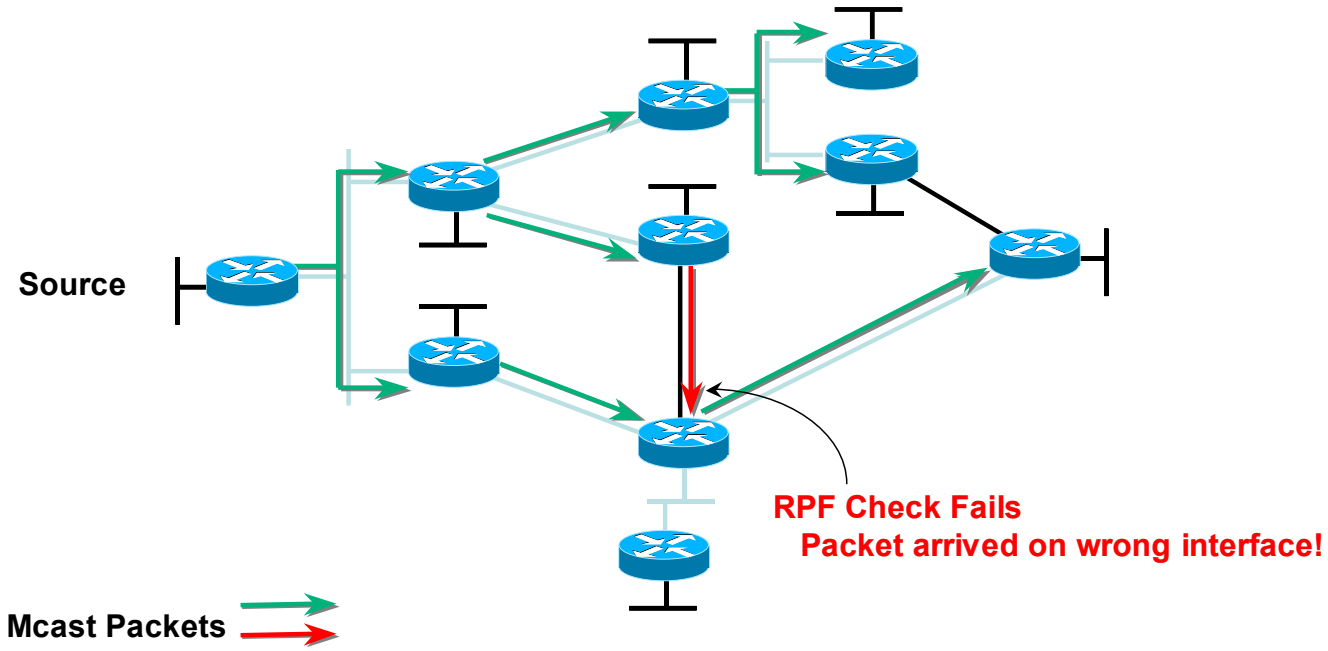


Figure 3-4: PIM

PIM Dense Mode (PIM-DM) uses a push model to flood multicast traffic to every corner of the network. This is a brute-force method for delivering data to the receivers, but in certain applications, this might be an efficient mechanism if there are active receivers on every subnet in the network.

PIM-DM initially floods multicast traffic throughout the network. Routers that do not have any downstream neighbors prune back the unwanted traffic. This process repeats every 3 minutes

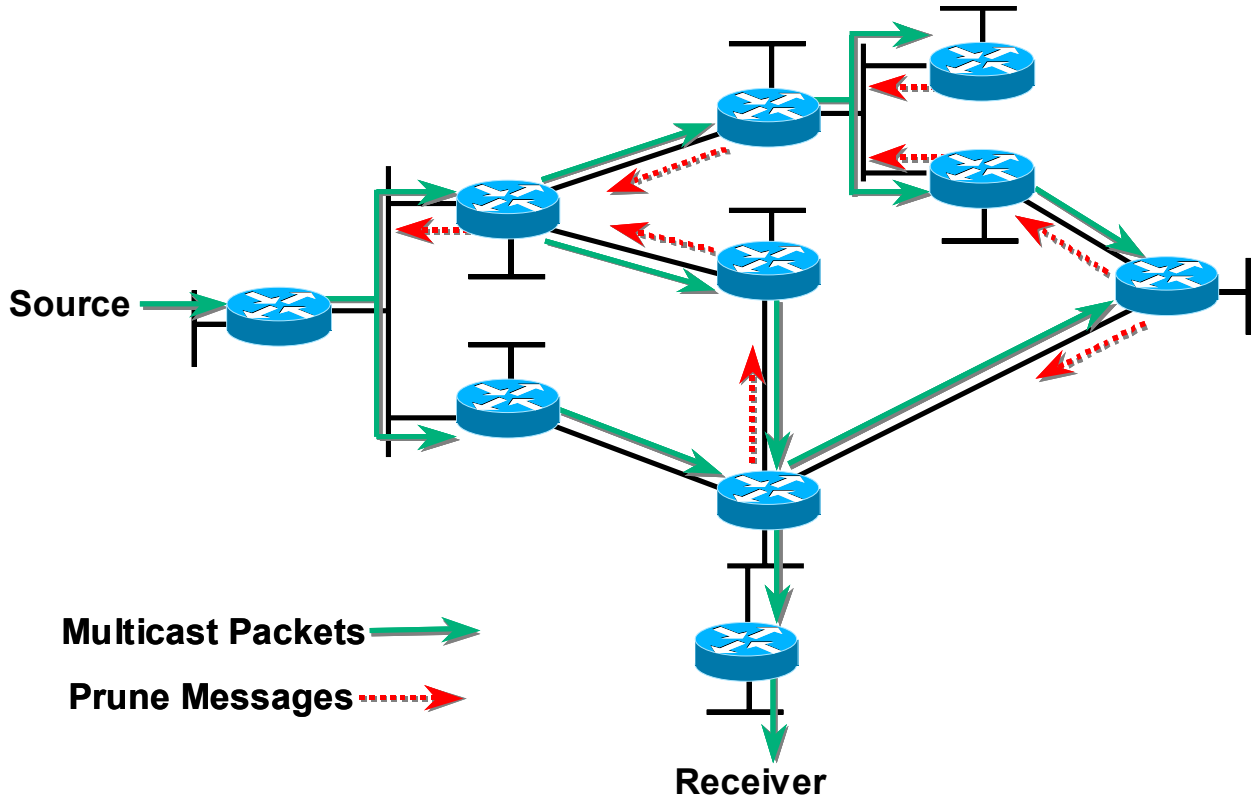


Figure 3-5: PIM-DM

PIM Sparse Mode (PIM-SM) uses a pull model to deliver multicast traffic. Only networks that have active receivers that have explicitly requested the data will be forwarded the traffic.

PIM-SM uses a shared tree to distribute the information about active sources. Depending on the configuration options, the traffic can remain on the shared tree or switch over to an optimized source distribution tree. The traffic starts to flow down the shared tree, and then routers along the path determine whether there is a better path to the source. If a better, more direct path exists, the designated router (the router closest to the receiver) will send a join message toward the source and then reroute the traffic along this path.

PIM-SM has the concept of an Rendezvous Point, since it uses shared trees, least initially. The RP must be administratively configured in the network. Sources register with the RP, and then data is forwarded down the shared tree to the receivers. If the shared tree is not an optimal path between the source and the receiver, the routers dynamically create a source tree and stop traffic from flowing down the shared tree.

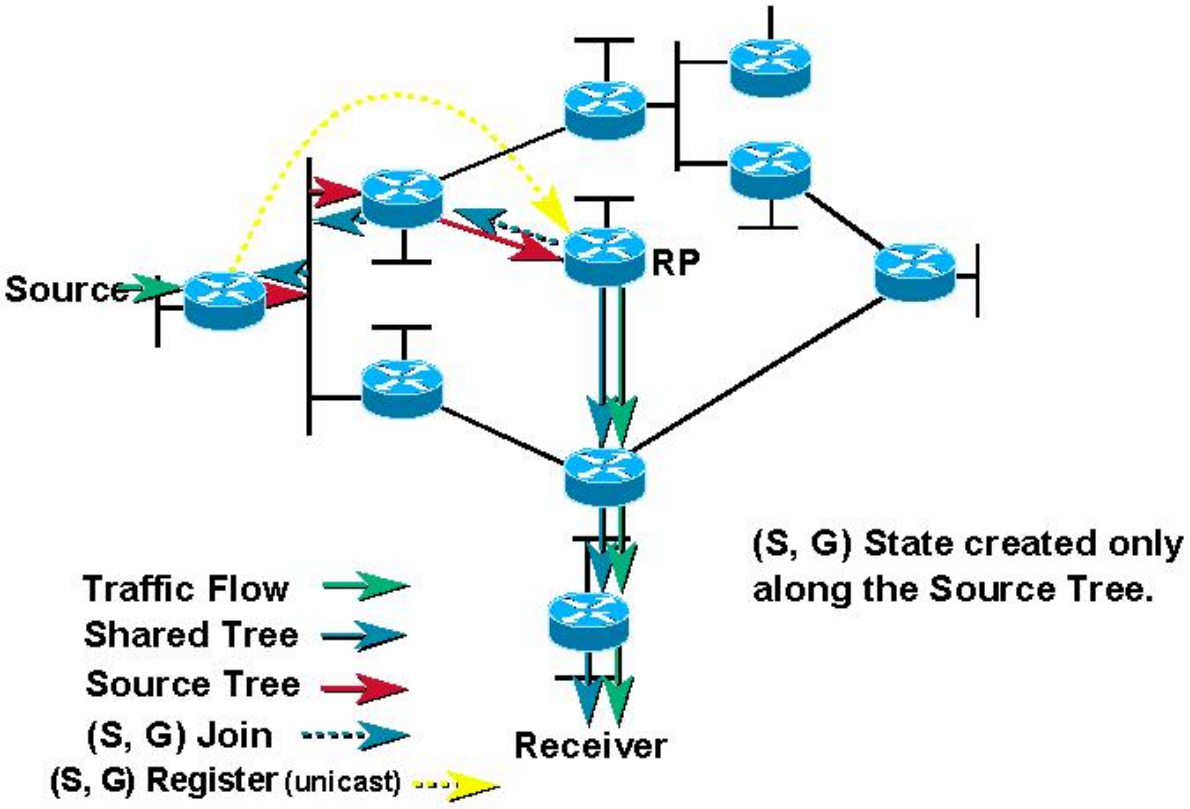


Figure 3-6: PIM-SM

In PIM-SSM (Source Specific Multicast), delivery of datagrams is based on (S, G) channels. Traffic for one (S, G) channel consists of datagrams with an IP unicast source address S and the multicast group address G as the IP destination address. Systems will receive this traffic by becoming members of the (S, G) channel. No signaling is required to become a source. However, in SSM, receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources. In other words, receivers can receive traffic only from (S, G) channels that they are subscribed to, so a RP is not necessary.

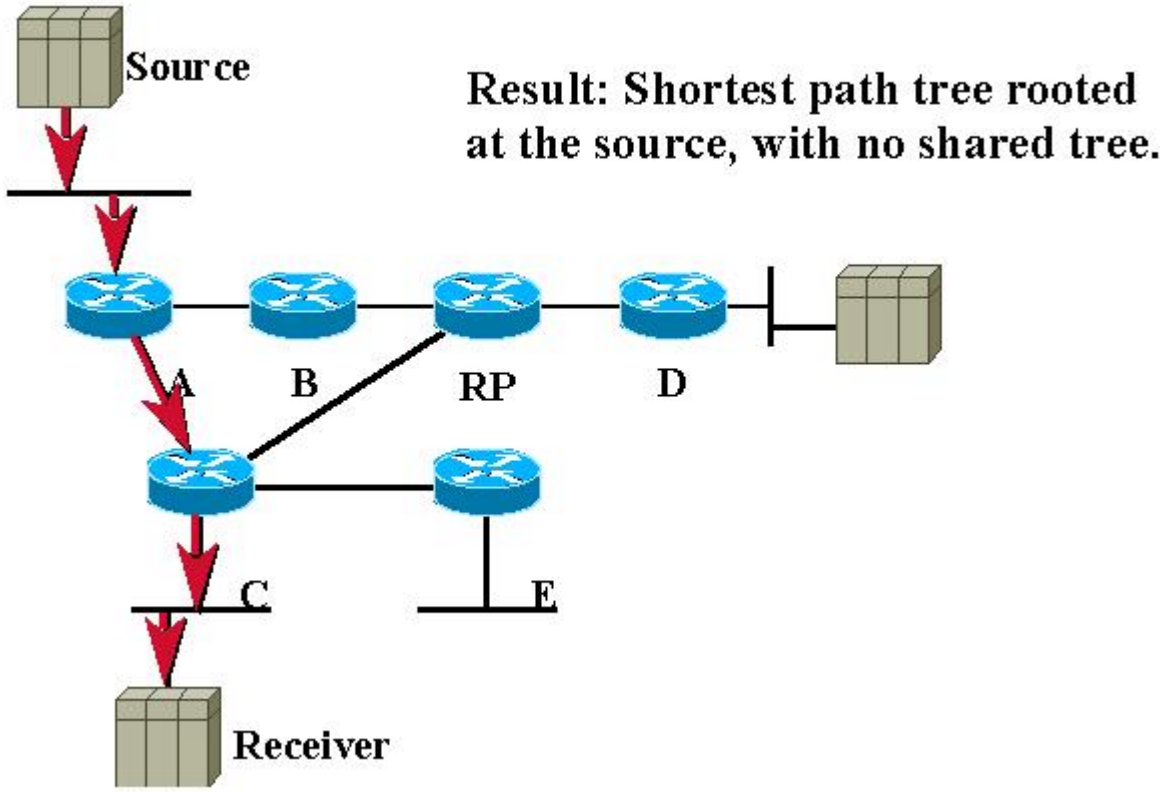


Figure 3-7: PIM-SSM

PIM-SM [13] is available in advanced configuration on 6WINDGate and follows draft-ietf-pim-sm-v2-new-05.

3.2.5 Management Architecture

Figure 3-8 describes the IP router network management architecture. There are four solutions to configure the device.

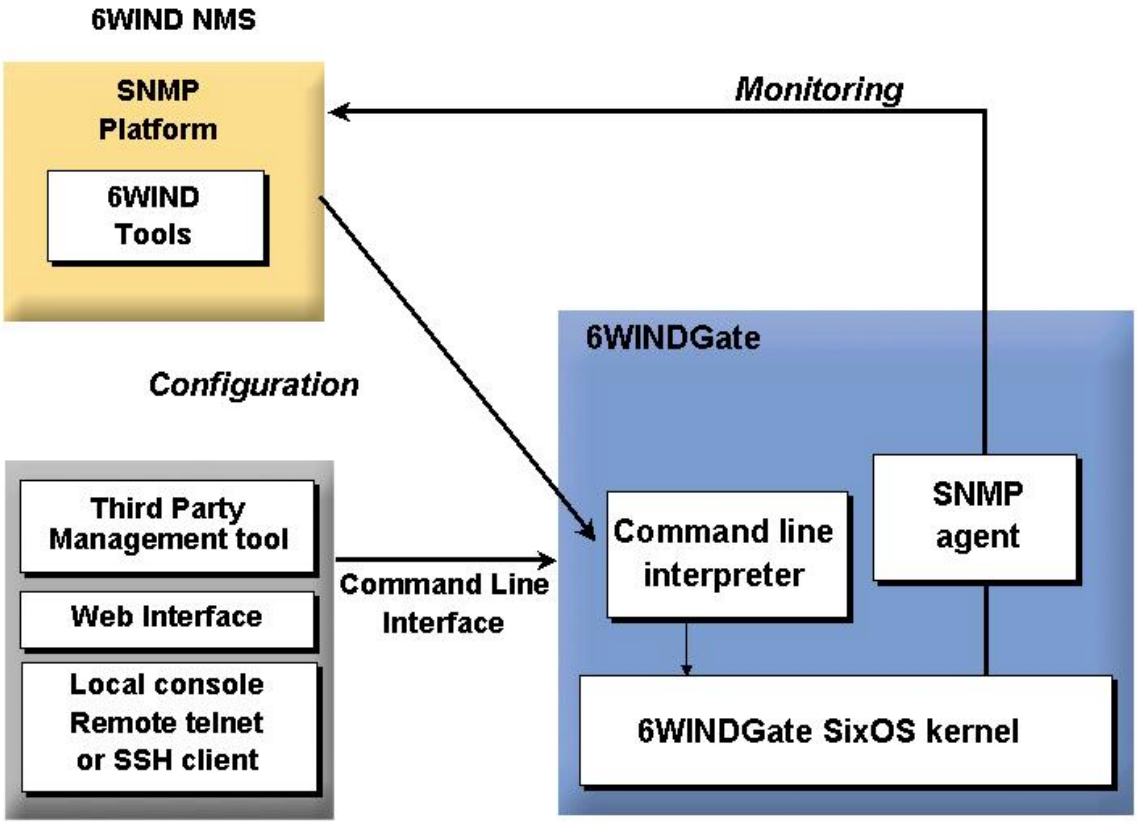


Figure 3-8: Management Architecture

The CLI installed on the equipment is the basic interface to configure the 6WINDGate using a local console, a remote telnet or a SSH client. This CLI will be extended with PLC dedicated commands in order to program PLC modem.

An IPv4/IPv6 SNMP agent [12] is implemented in IP router and so can be extended to support eventual PLC specific configuration.

4. SUMMARY AND CONCLUSIONS

Hardware and software specification of Head End have been described.

Proof-of-concept prototype and IP features have been defined including autoconfiguration, QoS, Multicast and Managenement.

Some advanced IP functionalities are based on IETF works on progress and will need to be improved along with changes of draft. Also experience acquired during the project will help to improved IETF standards and PLC specific work.

5. REFERENCES

[1] "Report on IPv6-over-PLC relevant issues", Deliverable 2.1, 6POWER Project (IST-2001-37613), September 2002.

[2] "IPv6-over-PLC interface specification", Deliverable 2.2, 6POWER Project (IST-2001-37613), May 2003

[3] Droms, R. "Internet Protocol, Version 6 (IPv6) Specification", Deering, S. and R. Hinden, RFC 2460, December 1998

[4] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[5] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[6] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[7] Aboba, B., Zorn, G. and D. Mitton, "RADIUS and IPv6", RFC 3162, August 2001

[8] Droms, R., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", draft-ietf-dhc-dhcpv6-28 (work in progress), November 2002.

[9] Miyakawa, S., "Requirements for IPv6 prefix delegation", draft-ietf-ipv6-prefix-delegation-requirement-03 (work in progress), March 2003.

[10] Troan, O., Droms, R. "IPv6 Prefix Options for DHCPv6", draft-ietf-dhc-dhcpv6-opt-prefix-delegation-04.txt (work in progress), June 2003.

[11] Droms, R., "DNS Configuration options for DHCPv6", draft-ietf-dhc-dhcpv6-opt-dnsconfig-03.txt (work in progress) February 28, 2003

[12] M. MacFaden, D. Partain, J. Saperia, W. Tackabury, "Configuring Networks and Devices with Simple Network Management Protocol (SNMP)", RFC 3512, April 2003

[13] W. Fenner, M. Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM) », draft-ietf-pim-sm-v2-new-07.txt (work in progress), March 2003

[14] D. Grossman, « New Terminology and Clarifications for Diffserv", RFC 3260, April 2002

[15] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[16] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998

[17] Heinanen, J., Baker, F., Weiss, W. and J. Wrocklawski, "Assured Forwarding PHB Group", RFC 2597, June 1999

[18] Davie, B., Charny, A., Baker, F., Bennett, J.C.R., Benson, K., Le Boudec, J., Chiu, A., Courtney, W., Cavari, S., Firoiu, V., Kalmanek, C., Ramakrishnam, K. and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)", RFC 3246, March 2002