IP and Integrated Services

Author: Saleem N. Bhatti

E-mail: S.Bhatti@cs.ucl.ac.uk

Telephone: +44 171 419 3249

Fax: +44 171 387 1397

Affiliation: Computer Science Department, University College London

Address: Computer Science Department
University College London
Gower Street
London
WC1E 6BT
UK

1. Introduction

During the 1990's, applications have become increasingly reliant on the use of the Internet protocols to provide data communications facilities. The use of the Internet protocols seems likely to increase at an extremely rapid rate and the Internet Protocol (IP) will be the dominant data communications protocol in the next decade. IP is being used for a huge variety of "traditional" applications, including e-mail, file transfer and other general non-real-time communication. However, IP is now being used for real-time applications that have quality of service (QoS) sensitive data flows[1]. Applications such as conferencing, telephony – voice-over-IP (VoIP) – as well as streaming audio and video are being developed using Internet protocols. The Internet and IP was never designed to handle such traffic and so the Internet community must evolve the network and enhance the Internet protocols in order to cater for the needs of these new and demanding applications. Users wish to have access to a whole plethora of telecommunication and data communication services via the Internet; users wish to access an **Integrated Services Network (ISN)**.

In this chapter, we consider the evolution of the changes that will be occurring in the Internet to support the ever-increasing demand of applications that populate it, and look at how to evolve the Internet to an ISN. We examine the requirements for provision of QoS awareness and QoS mechanisms within the network, as well as looking at the trends for the future.

In this chapter, we give an overview of a set of technologies that must work together and evolve together in order to allow Integrated Services provision over IP. Some of these technologies (e.g. RSVP, IP multicast) are describe in other chapters. Sections 2 and 3 consider the current state of the art and likely near-term developments for service deployment. Section 4 considers technology that is likely to be deployed in the mid-term while Section 5 looks at some longer-term technology issues.

2. Integrated Services

People today wish to use the Internet for many different applications. Some of these applications already exist on specific network technologies, e.g. voice on POTS, data on X.25, etc. As the ability to use a more diverse range of applications becomes available from the desktop to an increasing number of people, there is a higher demand for these applications. To supply the demand and provide access to such a diverse range of applications, it would be impractical to maintain access to each of the application-specific networks for each user. So, over the past two decades or so, there has been a move to provide a single integrated services network that can support the provision of any and all of these applications, e.g. N-ISDN (narrowband-ISDN), B-ISDN (broadband-ISDN). Although, in principle, such a network should be able to provide very good QoS guarantees, the notion of a single, ubiquitous (sub)-network technology is not realistic (and in fact today's Internet services are provided across networks that consist of many different technologies). The Internet protocols are widely available, generally easy to use, have well-defined software APIs, and can operate on many network technologies. Consequently the Internet is being seen as a means for allowing access to integrated services [DT97].

Internet users have increasing demands to use a range of multimedia applications with QoS sensitive data flows. All these applications may require different QoS guarantees to be provided by the underlying network. An e-mail application can make do with a best-effort network service. Interactive or real-time voice and video applications require (some or all of) delay, jitter, loss and throughput guarantees in order to function. Web access can make do with a best-effort service, but typically requires low delay, and may require high throughput depending on the content being accessed. The Internet was never designed to cope with such a sophisticated demand for services [Cla88] [RFC1958]. Today's Internet is built upon many different underlying network technologies, of different age, capability and complexity. Most of these technologies are unable to cope with such QoS demands. Also, the Internet protocols themselves are not designed to support the wide range of QoS profiles required by the huge plethora of current (and future) applications. This deficiency is currently being addressed by the IETF INTSERV (Integrated Services) WG[2] [RFC1633]. The explosive growth in the use of the Internet has resulted in much of the network being heavily loaded or overloaded. So there is a need to allow controlled use of resources.

In [CSZ92], the authors speak of the Internet evolving to an integrated services packet network (ISPN), and identify four key components for an Integrated Services architecture for the Internet:

---

[1] A **flow** is a stream of semantically related packets which may have special QoS requirements, e.g. an audio stream or a video stream.

[2] http://www.ietf.org/html.charters/intserv-charter.html

- **service-level:** the nature of the commitment made, e.g. the INTSERV WG has defined **guaranteed** and **controlled-load** service-levels (these are discussed later) and a set of control parameters to describe traffic patterns
- **service interface:** a set of parameters passed between the application and the network in order to invoke a particular QoS service-level, i.e. some sort of signalling protocol plus a set of parameter definitions
- **admission control:** for establishing whether or not a service commitment can be honoured before allowing the flow to proceed
- **scheduling mechanisms within the network:** the network must be able to handle packets in accordance with the QoS service requested

The simple description of the interactions between these components is as follows:

1. a **service-level** is defined (e.g. within an administrative domain or, with global scope, by the Internet community). The definition of the service-level includes all the service semantics; descriptions of how packets should be treated within the network, how the application should inject traffic into the network as well as how the service should be policed. Knowledge of the service semantics must be available within routers and within applications
2. an application makes a request for service invocation using the **service interface** and a **signalling protocol**. The invocation information includes specific information about the traffic characteristics required for the flow, e.g. data rate. The network will indicate if the service invocation was successful or not, and may also inform the application if there is a service violation, either by the application's use of the service, or if there is a network failure
3. before the service invocation can succeed, the network must determine if it has enough resources to accept the service invocation. This is the job of **admission control** that uses the information in the service invocation, plus knowledge about the other service requests it is currently supporting, and determines if it can accept the new request. The admission control function will also be responsible for policing the use of the service, making sure that applications do not use more resources than they have requested. This will typically be implemented within the routers

4. once a service invocation has been accepted, the network must employ mechanisms that ensure that the packets within the flow receive the service that has been requested for that flow. This requires the use of **scheduling mechanisms** and **queue management** for flows within the routers

We examine how the realisation of these key components have developed during this decade and discuss how research and development may progress in the next decade in order to move the Internet towards an ISN.

2.1. QoS service definitions and service invocation

The IETF INTSERV WG has proposed an architecture for evolving the Internet to an ISN. To support the architecture, INTSERV have produced a set of specifications for specific QoS service-levels based on a general network service specification template [RFC2216] and some general QoS parameters [RFC2215]. The template allows the definition of how network elements should treat traffic flows. With the present IP service enumerated as **best-effort**, currently, two service-level specifications are defined:

- **controlled-load** service [RFC2211]: the behaviour for a network element required to offer a service that approximates the QoS received from an unloaded, best-effort network
- **guaranteed** service [RFC2212]: the behaviour for a network element required to deliver guaranteed throughput and delay for a flow

Also specified is how to use a signalling protocol, RSVP [RSVP], to allow the use of these two services to be signalled through the network [RFC2210]. INTSERV also define SNMPv2 extensions [RFC2213] [RFC2214] to allow remote monitoring and management of network elements that support these network services. Part of the INTSERV work is the definition of an architecture for a QoS Manager (QM) entity that co-ordinates flow activities and resource usage at the end system [INTSERVQM]. Note that this architecture requires that the network elements and applications have semantic knowledge about the service-levels for the application flows, as specified in the service templates.

RSVP is used by applications to make a **resource reservation**, by asking the network to provide a defined quality of service for a flow. The reservation request consists of a *FlowSpec* identifying the traffic characteristics and service-level required. One part of the *FlowSpec* is a *TSpec*, a description of the traffic characteristic required for the reservation. So it is possible for the same traffic characteristic to be used with different service levels. This difference in QoS service-level could, for example, act as a way for offering cost differentials on the use of a particular application or service.

To invoke a particular service, the application uses a signalling protocol, RSVP, for a particular communication session (which may consist of one or more flows). To make a resource reservation, an appropriate *FlowSpec* is used along with session IP destination address, the protocol number in the IP packet and – optionally – the destination port number in the service invocation. The reservation procedure is as follows. The sender transmits a *Path* message advertising the session QoS requirements towards the destination IP address. All RSVP routers forwarding the *Path* message hold **soft-state** – information about the resource reservation required – until one of the following happens: a *PathTear* is sent from the sender cancelling the reservation, a *Resv* message is transmitted from a receiver effectively confirming the reservation, or the soft-state times-out. A *Resv* message from a receiver is sent back along the same route as the *Path* message[3], establishing the reservation and then the application starts sending data packets. *Path* and *Resv* messages are sent by the sender and receiver, respectively, during the lifetime of the session to refresh the soft-state and maintain the reservation. A *PathTear* or *ResvTear* message explicitly tears down the reservation and allows resources to be freed. It is possible for the reservation to be changed dynamically during the lifetime of the session. RSVP can be used for unicast or multicast sessions.

RSVP allows the reservation to be made using filters that control how the reservation is applied. A **fixed-filter** (FF) is used to make a distinct reservation (can not be shared by other flows along the path) with an explicit sender selection criteria (similar to a closed user group in telephony). A **shared-explicit** (SE) filter is used to request a reservation that is a union of all the requirements of the senders but still with explicit user selection. A **wildcard-filter** (WF) is a shared reservation with an open sender selection, i.e. an open group. FF would typically be used for unicast reservations or reservations for a lecture type multicast session. FF or SE would be used for closed user groups, e.g. virtual meeting rooms. SE and WF would be used for open multicast groups, e.g. public seminars or conferencing.

Note that RSVP:

- provides end-to-end signalling (between applications)
- RSVP sets up unidirectional reservations

- is specific to one session
- requires the applications and the network to be RSVP and INTSERV aware

2.2. QoS service provision

In the last subsection we have considered how the QoS services are defined and how they are invoked. We now consider how they are implemented in the network. In fact, the INTSERV WG does not mandate any particular algorithms or mechanisms for the provision of a particular service. The INTSERV WG defines the behaviour required in the network elements and only suggests ways in which this behaviour might be implemented at the current time, albeit with reference to existing implementation techniques. The philosophy behind this approach is that as technology matures or new, better technology is produced, it can be used to provide the same service as long as the service behaviour is honoured.

In the wide area, the behaviour implemented depends on the underlying network technology (the bearer service used by IP) to provide the links between the routers, and the behaviour of the routers connecting the various IP links. The bearer technology may be asynchronous transfer mode (ATM), frame relay (FR) or various point-to-point technologies, e.g. SONET/SDH (synchronous optical network/synchronous digital hierarchy). In some cases, it may be possible to exploit the properties of the underlying technology in order to achieve some traffic engineering goals at the IP level, e.g. use of ATM victual circuits for traffic segregation. Work is in progress within the Integrated Services for Specific Link layers (ISSLL) WG[4] and the Interworking over Non-Broadcast Multiple Access networks (ION) WG[5] of the IETF in order to provide solutions that are specific to certain bearer technologies. In particular, at the time of writing, the ISSLL WG work on mappings of INTSERV onto ATM was approaching publication. This includes the mapping of RSVP onto ATM signalling.

The ISSLL WG is also tasked with examining the provision of IP Integrated Services within an IEEE 802 based LAN environment, and this is currently work in progress. The IEEE has recently extended the definition of IEEE 802.1D to support priority classes for traffic (the work was carried out under a working group that was

---

[3] It is assumed that routes are symmetrical and relatively stable, but this is not always true in the wide area [Pax98a].

[4] http://www.ietf.org/html.charters/issll-charter.html

[5] http://www.ietf.org/html.charters/ion-charter.html

originally labelled IEEE 802.1p). There is work in progress within the Internet community to map INTSERV and Differentiated Services (DIFFSERV – see below) onto such mechanisms.

However, the main issue concerning Integrated Services provision is the handling of the individual packet that make up a flow in order to honour the QoS requirements of that flow. The router has a non-trivial forwarding process for each packet:

- classify the packet in order to identify its QoS requirements (classification)
- determine when the packet should be forwarded (scheduling)
- manage the output queues under congestion conditions (queue management)

Note these activities are logically distinct from the *routing* functions that all routers must be able to perform in order to determine in which direction to forward a packet (i.e. which output interface should be used). Several schemes have been developed within the Internet community for performing classification, scheduling and queue management tasks, and they are currently undergoing experimentation and development. The most popular mechanisms establish a class-based hierarchy that allows sharing of resources in some way, for example sharing of the link capacity. The mechanisms are refined in order to incorporate scheduling mechanisms that ensure that packets are transmitted within a given time frame. The mechanisms are based around analysis presented in [PG93] [PG94], which defines a model that allows fair sharing of resources. However, the realisation of this model is subject to some practical constraints in implementation, due in part to the computational complexity of the algorithms involved. Three models currently receiving attention within the Internet Community are weighted fair queuing (WFQ) [DKS90], class-based queuing (CBQ) [FJ95] [WGCJF95] and worst-case fair weighted fair queuing (WF^2Q+) [BZ96]. Note that “fair” does not necessarily imply “equal”, and all of the techniques that have been developed allow for different users[6] to have different shares of resources. For example, CBQ was designed to allow sharing of link capacity within a class-based hierarchy, and in Figure 1, we see an example showing the link capacity as a root node in a tree at 100%. Organisations X, Y and Z that share the link are assigned 40%, 30% and 30% of the link capacity, respectively. Within their own allocations of capacity, the organisations can choose to partition the available capacity further by creating sub-classes within the tree. Organisation X decides to allocate 30% to real-time traffic and 10% to all non-real-time traffic. Within the real-time allocation, X decides to allocate capacity to individual applications. Organisation Y also divides its allocation into real-time and non-real-time, but with a different share of the available link capacity. Organisation Z decides not to further refine its allocation of link capacity. The percentages indicate the minimum share of the link capacity a node in the tree will receive. Child nodes effectively take capacity from their parent node allocation. If some sibling nodes are not using their full allocations, other siblings that might be overshooting their own allocation are allowed to “borrow” capacity by interacting with the parent node. With an appropriate scheduling mechanism, this allows support for QoS sensitive flows. Classifications in Figure 1 could be made per application, per flow, per IP source address, etc., as dictated by the policy chosen by the individual organisations in conjunction with their Internet connectivity provider.

WFQ, CBQ and WF^{2}Q+ have different capabilities and different levels of computational complexity, depending on the policy used to define the granularity of the flow, and the exact nature of the resource sharing implemented. However, they all have their advantages and drawbacks and, at the time of writing, are still generally in experimental use, though products will soon be appearing incorporating these schemes.

When a resource reservation is invoked, one of the functions that may be applied is **admission control**. Given a suitable description of the resource reservation requirements for a flow, admission control determines whether or not it is currently possible to provide the service required for the flow. This also requires knowledge of other flows that are currently sharing any resources along the network path of the flow. The nature of the admission control algorithm is dependent on the type of service that is being invoked; controlled-load service admission control will be handled in a different manner from guaranteed service admission control. Where a network element supports both controlled-load and guaranteed services for different flows, careful engineering must ensure that the service commitments undertaken by the network element are maintained. There is work in progress within the IETF to address admission control, and schemes have been proposed by the research community (e.g. [BFMM94] [JDSZ95]).

3. Differentiated services

We have said that resource reservation with RSVP is a useful mechanism for applications with QoS sensitive data flows. However, as IP cannot rely on any particular network technology-specific mechanisms, RSVP uses a

---

[6] The definition of “user” can be at different granularities, e.g. per IP address prefix, per IP source address, per application, etc.

soft-state technique with a two-pass protocol. We summarise the main problems with RSVP below [SB95] [WGS97]:

1. during reservation establishment if the first pass of each of two separate reservation requests are sent through the same network element, where one request is a super-set of the other, the lesser one may be rejected (depending on the resources available), even if the greater one eventually fails to complete (of course it is possible to re-try)
2. if the first pass does succeed, the router must then hold a considerable amount of state for each receiver that wants to join the flow (e.g. in a multicast conference)
3. the routers must communicate with receivers to refresh soft-state, generating extra traffic, otherwise the reservation will time out
4. complete heterogeneity is not supported, i.e. in a conference everyone must share the same service-level (e.g. guaranteed or controlled-load), though heterogeneity within the service-level is supported
5. if there are router failures along the path of the reservation, this results in IP route changes, so the RSVP reservation fails and the communication carries on at best-effort service, with the other routers still holding the original reservation until an explicit tear-down or the reservation times out or the reservation can be re-established along the new path
6. the applications must be made RSVP aware, which is a non-trivial goal to realise for the many current and legacy applications that already exist, including multimedia applications with QoS sensitive flows

Resource reservation could be expensive on router resources and adaptation capability is still required within the application to cope with reservation failures or lack of end-to-end resource reservation capability. Indeed, the Internet community has acknowledged the shortcomings of RSVP, especially with respect to scaleability, and it is now recommended for use only in restricted network environments [RFC2208]. Such concerns about resource reservation have directed the Internet community to consider alternatives; specifically differentiated services [DIFFSERV]. Without resource reservation, we require some mechanisms to allow service differentiation within the network, but also we require a more flexible and **dynamic adaptation** capability within the application.

3.1. Service differentiation

The IETF DIFFSERV (Differentiated Services) WG[7] takes a different view of using network resources to that of the INTSERV WG. At the time of writing, this work is still at very early stages, so there are several schemes being discussed. The general model is to define a class-based system where packets are effectively marked with a well-known label. This label identifies the aggregate service-level the packet will receive much like a letter can be marked as registered, first class or second class delivery. This is a much coarser granularity of service, but reflects a well understood service model used in other commercial areas. The DIFFSERV model is different to RSVP. A key distinction of the DIFFSERV model is that it is geared to a business model of operation, based on administrative bounds, with services allocated to users or user groups. Whereas RSVP can act on a per-flow basis, the DIFFSERV classes may be used by many flows. Any packets within the same class must share resources with all other packets in that class, e.g. a particular organisation could request a Premium (low delay) quality with an Assured (low loss) service-level for all their packets at a given data rate from their provider. The packets are treated on a per-hop basis by *traffic conditioners*, routers that determine the way a packet should be treated based on a policy that is selected by examining the value of the class marking of the packet. The policy could be applied to all the traffic from a single user (or user group), and could be set up when subscription to the service is requested, or on a configurable profile basis. The DIFFSERV mechanisms would typically be implemented within the network itself, without requiring runtime interaction from the end-system or the user, so are particularly attractive as a means of setting up tiered services, each with a different price to the customer.

The RSVP mechanism seeks to introduce well-defined, end-to-end, per-flow QoS guarantees by use of a sophisticated signalling procedure. The DIFFSERV work seeks to provide a "virtual pipe" with given properties in which the user may require adaptation capability or further traffic control if there are multiple flows competing for the same "virtual pipe" capacity. Additionally, the DIFFSERV architecture means that different instances of the same application throughout the Internet could receive different QoS, so the application needs to be adaptable.

The service itself will be defined in terms of a **service level agreement (SLA)** that embodies the contract between the service user and service provider. The policy implemented by the SLA may include issues other than QoS that must be met, e.g. security, time-of-day constraints, etc. Figure 2 highlights the main difference

---

[7] http://www.ietf.org/html.charters/diffserv-charter.html

between INTSERV and DIFFSERV scope. INTSERV tries to provide, per application, end-to-end resource reservation. DIFFSERV aims to provide a SLA-based contract between service networks. One very attractive feature of DIFFSERV is that it can be introduced into existing networks in a piecewise manner, without having to modify current or legacy applications. The packets leaving a network are marked for DIFFSERV handling by DIFFSERV-capable routers that sit at administrative boundaries. Therefore, only the routers need to be updated and the applications themselves can remain unchanged. (However, this does not preclude individual hosts or individual applications being DIFFSERV-aware and marking packets accordingly as they leave the host.) The DIFFSERV-capable routers could be at the edge of the customer network or part of the provider's network. If the DIFFSERV-marking is performed within the customer network, then policing is required at the ingress router at the provider network in order to ensure that customer does not try to use more resources than allowed by the SLA.

3.2. Providing differentiated services

The DIFFSERV work is aimed at providing a way of setting up QoS using policy statements that form part of a service level agreement between service user and service provider. The policy may use several packet header fields to classify the packet, but the classification marking is a simple identifier (currently a single byte) within the packet header. The classification is by way of a special value for a single header field, the **DS (differentiated services) byte**, which will be used in place of the ToS (Type of Service) field in IPv4 packets or the traffic-class field in IPv6 packets. The DS byte will have the same syntax and semantics in both IPv4 and IPv6. There are likely to be some global values – **DS codepoints** – agreed for the DS field within the IETF but the intention is that the exact policy governing the interpretation of the DS codepoints and the handling of the packets is subject to some locally agreed SLA. SLAs could exist between customer and Internet Service Provider (ISP) as well as between ISPs. The DS codepoints are used to identify packets that should have the same aggregate **per-hob behaviour (PHB)** with respect to how they are treated by individual network elements. The PHB definitions and the DS codepoints used may differ between ISPs, so there will be need for translation mechanisms between ISPs.

The meaning of the DS codepoints and the content of SLAs are established at subscription time and although there will be scope for change by agreement between customer and provider, the kind of dynamic and flexible resource reservation that is described above for using RSVP is not envisaged for DIFFSERV.

The mechanisms for classification of packets and handling of packets within the network can be the same as for INTSERV – WFQ, CBQ and WF^2Q+ could be used. The big gain is that the end-to-end signalling and the maintenance of per-flow soft-state within the routers that is required with RSVP is no longer required. This makes DIFFSERV easier to deploy and more scaleable than using RSVP and INTSERV services. However, this does not mean that INTSERV and DIFFSERV services are mutually exclusive. Indeed, it is likely that DIFFSERV SLAs will be set-up between customer and provider for general use, and then RSVP-based per-flow reservations may be used for certain applications as required, e.g. for instance an important video conference within an organisation. This concept is shown in Figure 3. The DIFFSERV capability provides the aggregate service to the provider while individual applications with special needs can use RSVP to set-up INTSERV reservations within this aggregate "pipe", as required.

Note that while INTSERV is based on the notion of *receiver* generated control messages for confirming the resource reservation, DIFFSERV requires that the ISPs for the *receiver and the sender*, have a way of allowing the PHB definition to be honoured *across the network*. This requires co-operation between many ISPs. So, it is expected that the DIFFSERV facilities will be used initially to offer individual customers of single ISPs the ability to establish virtual private network (VPN) scenarios, with the network of that single provider enabling the wide-area connectivity. Of course, individual ISPs (or backbone providers) may form peering agreements to enable wide-area connectivity based on DIFFSERV. Such connectivity could be used to provide enterprise Intranet services, as well as conferencing, group-working and software distribution based on use of IP-multicast across the VPN.

4. Performance enhancements for IP

With the evolution to Integrated Services provision with IP, one thing is certain: the amount of IP traffic will increase so the networks must be able to handle this increased load. Over the past few years, particular emphasis has been placed on developing techniques to allow increased performance of IP-based networks. It should be noted that performance issues are not necessarily the same as QoS issues. Performance issues are concerned with getting packets from A to B across the network as fast as possible. QoS issues are concerned with making sure that as packets traverse the network, they receive appropriate handling at the routers to ensure that QoS performance criteria (such as delay, jitter, data rate etc.) are met. Nevertheless, as IP traffic increases so the networks must be able to handle large volumes of IP packets else the QoS criteria may not be met.

In this section we examine three of the issues affecting performance that may impact QoS – the use of IP over high-speed bearer services, enabling fast forwarding mechanisms within the network, and the evolution of IP routers.

4.1. High speed bearer services

A suitable sub-network technology to provide integrated services capability might be asynchronous transfer mode (ATM). ATM is itself designed to be an integrated service bearer, so IP and ATM might be seen as competing technologies. However, the evolution seems to be that there will probably be few "native" ATM applications, but many IP applications already exist and many more are being created. So there has been much activity within the Internet community to make IP work effectively over ATM networks [RFC1821]. Basic connectivity mechanisms for IP over ATM have been proposed by the IETF [RFC2225], however other solutions, not designed specifically for IP but with the advantage of providing supporting for other layer 3 protocols, have been proposed by the ATM Forum[8] – LAN Emulation [LANEv2] and Multi-Protocol Over ATM [MPOA]. These technologies all provide an encapsulation mechanism for IP and a set of rules for establishing ATM-level connectivity for the transportation of IP packets.

In general, there is a need to allow IP to be carried in a whole range of non-broadcast multiple access (NBMA) scenarios including ATM, frame-relay and other point-to-point technologies, and this need is being addressed by the ION WG within the IETF. However, in certain backbone scenarios, the use of ATM is seen as an overhead for carrying IP, and many network operators are now investigating carrying IP packets directly in SONET/SDH frames. Indeed, an encapsulation method for IP in SONET/SDH [RFC1619], IP over ISDN [RFC1618], IP in Frame Relay [RFC1973] have all existed from some time, based on the use of the standard Point-to-Point Protocol (PPP) [RFC1661].

However, these mechanisms say nothing of how high performance can be achieved with IP. In general, there are likely to be relatively few problems with allowing IP to be carried within a particular network bearer service – IP works over anything.

The mapping of IP onto any lower layer should be as simple as possible so that protocol overhead does not become a performance bottleneck. An example of protocol inefficiency is seen in the protocol stack of

[8] http://www.atmforum.com/

[RFC2225] for Classical IP over ATM (CIPA). The main goal for CIPA is connectivity, and Figure 4 shows the protocol stack used to attain connectivity with CIPA. So, an IP packet must be framed in a LLC frame, then within an AAL5 protocol data unit and then shredded into ATM cells. This process must be reversed at every IP-level routing hop within the ATM network (to reform the IP packet) and then the IP packet must be re-encapsulated with the same process if forwarding onto another ATM interface from the router. CIPA does not allow direct ATM-level communication between IP-nodes at the ATM-level if they are on *different IP sub-networks*, even if the are on the *same ATM network*.

So the main issue for performance is to find an efficient forwarding scheme for transporting IP packets over NBMA network connections.

4.2. Fast forwarding mechanisms

The problem of making fast forwarding decisions is inherent in IP networking. A description of the task is quite simple – to move a packet from an input port to an output port as fast as possible. To make a forwarding decision for an IP packet the following steps take place at a router:

1. a packet arrives at input port and the packet may need to be buffered
2. the router must read the destination address of the packet
3. based on the destination address, the router selects candidate routing table entries, and for each candidate entry, saves the next hop address, the address mask for the address and the output port for that entry
4. after all the candidate entries have been found an entry must be selected by using the longest prefix match using the routing entry address mask and the destination address in the packet
5. when the appropriate candidate entry has been selected, the packet is placed on the appropriate output queue

Steps 3 and 4 in this process may require the consideration of other information such as routing metrics, policy-based routing, security information, etc. In general, this may slow down the forwarding process, although clever caching and recent developments in table lookups can help (more on this below). Where there are many packets in a flow, it seems that this process need only be executed once as all packets for the flow will be subject to the same forwarding decision. This is the main principle behind **multi-protocol label switching (MPLS)**. MPLS expedites the forwarding process by using simple, fixed-length labels to identify packets within a flow. The labels may be set up using network management tools or other administrative measures or may be generated dynamically as a flow is detected. The label acts as a selector, just like a virtual circuit identifier (VCI), with only local significance, to allow "switching" of IP packets based on labels rather than on IP address information. This is sometimes called short-cut routing or cut-through routing, reflecting the fact that the concatenation of the locally generated labels along a path describe the route for a packet along that path. As the labels are of a short, fixed length (currently 20-bits), they are easy to look up using tables. Note that this is not a new *routing* mechanism – it is simply a way of making *forwarding* decisions easier and in fact still uses standard IP routing information and relies on standard IP routing protocols to establish the forwarding table. The label sits as part of a short (32-bit) shim-header between the link-layer header and the IP-header. In fact, as the name suggests, MPLS is designed to work for any layer 3 packet switched protocol not just IP, but most of the effort is currently around the implementation of IP solutions. Use of labels in this way introduces its own requirements: labels must be generated, distributed and maintained throughout the network. The MPLS technology is work in progress and covers all aspects of label distribution and handling.

Different vendors have already produced products that use different flavours of cut-through routing to exploit sub-network specific technology features in place of the generic label of MPLS e.g. use of ATM VCI/VPI (virtual circuit identifiers/virtual path identifiers) to switch ATM cell streams containing IP packets.

4.3. Smarter, not just faster

Making networks faster means more than just increasing the line speed connecting the routers, it means improving the performance of the routers themselves [KS98] [KLS98]. Current technology is already at the point where memory access speeds and the execution of the software algorithms implementing the forwarding code within routers is becoming the bottleneck. The design and implementation of router hardware and software is now an art and can determine the overall performance of a network more than the line speed of the individual links. In dealing with the provision for INTSERV/DIFFSERV mechanisms, router manufacturers and the research community are working hard, with some success, to produce "smarter" routers with better software, and not just running existing routing/forwarding software on faster hardware platforms.

Also, for high-speed packet processing, routers must be able to process and classify packets at line speed, i.e. there should be no queuing before packets have been classified, lest this delay contribute to the violation of the QoS requirements for the flow to which this packet belongs. Additionally, routers must not rely on any knowledge of possible traffic patterns as history shows the traffic patterns are hard to model and predict [PF95] [PF97] [Pax98b].

There is much progress in devising new algorithms for performing fast routing table look-ups [BCDP97] [VTP97] and packet classification [LS98]. There are moves to try and integrate the hardware and software as much as possible and devise algorithms that are as simple as possible so that they can be implemented in hardware.

This aim of "hardware-friendliness" is also visible with the evolution of the IP protocol itself. In Figure 5 we can compare the IPv4 packet header [IPv4] and the (currently proposed) IPv6 header. We see that the latter is much simpler in nature. We see that the IPv6 header is much more amenable to hardware processing than the IPv4 header, and as fragmentation and re-assembly have now become an end-to-end issue in IPv6, this simplifies the router's task in handling IP packets. Additionally, IPv6 has (potentially) better support for QoS support by including a flow-label and the traffic-class field within the first word of the IP packet header, however the exact use of both these fields has not yet been fully defined.

So, current work suggests it is wise to take into account QoS and performance issues when considering hardware, software *and* protocol design and this trend seems set to continue and will be an important factor in promoting IP as an integrated services bearer.

5. Technology components for future IP Integrated Services

So far we have considered current research or development that is likely to become deployed within the next decade. In this section we consider three technology issues that are likely to impact Integrated Services in the coming decade, and may change the way in which applications and services are used and deployed.

5.1. Dynamically adaptable applications

We have discussed above that the original INTSERV work using RSVP may not scale and it is likely that DIFFSERV and INTSERV mechanisms will be used together. This has the potential to allow different users of the same applications to have different QoS. In general, the QoS experienced by a particular application instance may vary due to a number of factors:

- variations in network behaviour due to network traffic from other sources

- variations in network paths due to the behaviour of routing functions
- the application resides on a mobile host
- the (human) user selects different user preferences depending on the costs of a particular service or the QoS required for a particular use of an application

The first three of these are based on QoS observed by an application instance during operation. In that case, the application must be able to detect QoS changes and adapt its operation to match the QoS available at the current time. Additionally, differences in QoS may be because different users of the same application subscribe to different service-levels from their ISP. Most ISPs only currently provide a single best-effort service, but this is sure to change in the near future.

The final bullet in the list above is a distinct choice made by the user. For example, consider that the user has a video-telephony application. When that application is used to contact family, the user may select high-quality, full-screen video and high-quality audio, but when the same application is used to contact the office, the user may select slow-scan, small-size video and phone quality audio. So the application must be able to adapt in response to changes in the network QoS as well as change in user preferences. We can see that this adaptation is dynamic and involves changes in the application's configuration. Such changes in configuration are currently handled manually, and must rely on a knowledgeable user being able to determine the correct application configuration for a particular network QoS scenario. We need mechanisms that can provide summaries of QoS information that allow either the application to dynamically adapt (re-configure) itself automatically (taking into account network QoS factors as well as user preferences) or to least allow the user to make an informed decision using simple feedback to the user [BK98a]. There is currently some work in progress to design mechanism that can enable dynamic adaptation [BK98b].

5.2. Active networks

Another way to try and capture the adaptation capability required for Integrated Services provision is to make the networks themselves adaptable. Such networks could consist of active network components and network elements that are effectively programmable by the application and can adapt their behaviour in response to changes in the network QoS or changes in the application behaviour (the latter occurring due to interaction with the user). In [Cam97], [LLB97] and [CCH96] are discussed issues concerning the provision of adaptation capability within the network itself. In such situations the QoS requirements for the flow, including adaptation capability, are submitted to the network which must manage resources to maintain the service for the user.

In more general terms, active networks allow the deployment of new services into the network in an incremental fashion as required. Therefore, in theory, a new application with its own sophisticated QoS requirements could effectively "download" the code for the processing mechanisms for its packets to all the relevant network elements along its communication path as required [TW96] [WLG98]. The application need not worry about the availability or deployment of definitions such as those currently being specified by INTSERV or DIFFSERV.

In order to enable such active networking, we need to have common application programming interfaces (APIs) that allow applications to interact with the network components. The task of establishing such APIs has been undertaken by the IEEE application Programming Interfaces for Networks (PIN) Working Group (P1520)[9]. The goal of IEEE-PIN is to define a reference model and set of APIs to allow access to network elements. Making the network elements programmable allows a much more flexible and dynamic approach to deployment of service and facilities in the network. The approach is heavily based on a distributed systems model, with resources and network entities modelled as objects that are accessed via well-defined interfaces. The intention is that the hardware and software aspects of service development and deployment are separated to the extent that the service may have some level of independence from the hardware substrate. At the time of writing, this work is at a very early stage, with no draft standards produced.

In general, there is currently great momentum behind the idea that the network can be made more active, and platforms such as Java[10] and CORBA (Common Object Request Broker Architecture)[11] are seen as enabling technologies in this arena.

5.3. Security

One of the biggest issues raised by the use of the Internet for carrying media flows such as voice and video (as well as other more sensitive data, e.g. credit card numbers!) is security. IP has a security architecture [RFC1825] as well as some specific security standards defined that allow encryption of packets to provide privacy in communication [RFC1827] [RFC1829], as well as to allow per-packet authentication [RFC1826] [RFC1828].

---

[9] http://www.ieee-pin.org/

[10] http://www.javasoft.com/

There are some technical issues concerning the use of security, for example the performance loss when security mechanisms such as encryption are used. However, most of the major security issues are currently concerned with national and international political activities and the provision of trusted third parties (TTPs). Various governments around the world see that provision of the strong cryptographic techniques will make it almost impossible for them to monitor the communications of criminals. This has led to legislation where the use of strong cryptography is only permissible if security agencies or other authorised government agencies have the ability to access the encrypted information. The implementation of this that is being proposed is the use of TTPs that will (securely) store the cryptographic keys that are used and make them available to an authorised body as required. The argument against such a mechanism is that it is highly unlikely that the criminals will register their keys with the TTP and will continue to use the strong cryptographic techniques that are available. TTPs acting as certification authorities (CAs) are also required in order to provide verification of electronic credentials – signatures for electronic identifiers.

Other security issues arise when considering active networking. Who has the right to program the network elements? How does a network element know the code is safe? What would happen if a network “virus” were to “infect” active network components? The security requirements of active networking have yet to be clearly identified.

6. Summary

The Internet protocols and APIs are widely used to develop applications that have particular QoS requirements. However, the Internet was never designed to offer QoS guarantees for applications. There is a need to provide support for QoS sensitive applications within the Internet using additional mechanisms. There is work in progress within the INTSERV WG of the IETF to produce an Integrated Services model and to develop QoS mechanisms that can reserve resources for applications. However, the current developments, based around the use of RSVP for application-network-application signalling are not fully deployed and indeed it is considered that, at the current time, they may not scale to for use across whole of the Internet.

The DIFFSERV WG of the IETF proposes a different model based on the classifying packets according to specific QoS requirements and implementing special packet handling criteria based on this packet classification.

---

[11] http://www.omg.org/

The DIFFSERV approach is more coarse-grained than the INTSERV approach, and is based on providing administratively controlled, service differentiation rather than fine-grained, per-application, dynamically requested QoS. This will allow ISPs to offer a tiered service on a per-customer or per-application basis.

As increasing numbers of users and applications make use of the Internet, the core network must be capable of handling large amounts of traffic. In order to ease the congestion that is currently seen across the Internet, there is a need to have new protocols and mechanisms to provide performance enhancements in the network elements, and not just faster transmission capability. One of the major bottlenecks in the Internet is the capability of the routers. To support INTSERV and DIFFSERV, routers must be enhanced with controlled scheduling, classification, queue management and fast-forwarding mechanisms.

Applications need to be dynamically adaptable so that they can be easily re-configured (under user control) to make best use of the resources available to them in a particular situation. Networks themselves may become active and programmable to support the diverse range of applications, QoS options and user preferences that may be available. In order to allow protected real-time communication such as person-to-person voice and video flows and conferencing, security mechanisms will be required.

7. References

[BCDP97] A. Brodnik, S. Carlsson, M. Degermark, S. Pink, “Small Forwarding Tables for Fast Routing Lookups”, Proc. ACM SIGCOMM’97, pp3-14, Sep 1997

[BFMM94] A. Banerjea, D. Ferrari, B. A. Mah, M. Moran, “The tenet real-time protocol suite: Design, implementation, and experiences”, Technical Report TR-94-059, University of California at Berkeley, Berkeley, California, Nov. 1994.

[BK98a] S. N. Bhatti, G. Knight, “QoS Assurance vs. Dynamic Adaptability for Applications”, Proc. 8th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV’98), New Hall, Cambridge University, Cambridge, UK, 8-10 Jul 1998.

[BK98b] S. N. Bhatti, G. Knight, “Notes on a QoS information model for making adaptation decisions”, Proc. 4th International Workshop on High Performance Protocol Architectures

(HIPPARCH'98), University College London, London, UK, 15-16 Jun 1998.

[BZ96] J. C. R. Bennett, H. Zhang, "Hierarchical Packet Fair Queue Algorithms", Proc. Acm SIGCOMM'96, pp143-156, Sep 1996.

[Cam97] A. T. Campbell, "Mobiware: QoS-Aware Middleware for Mobile Multimedia Networking", Proc. IFIP 7th International Conference on High Performance Networking, White Plains, New York, Apr 1997.

[CCH96] A. Campbell, G. Coulson, D. Hutchison, "Supporting Adaptive Flows in Quality of Service Architecture", ACM Multimedia Systems Journal, May 1996

[Cla88] D. D. Clark, "The Design Philosophy of the DARPA Internet Protocols", Proc. ACM SIGCOMM'88, pp106-114, Aug 1988.

[CSZ92] D. D. Clark, S. Shenker, L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", Proc. ACM SIGCOMM'92, pp14-26, Aug 1992.

[DIFFSERV] K. Nichols, S. Blake (Eds), "Differentiated Services Operational Model and Definitions", IETF DIFFSERV WG, work-in-progress, Feb 1998.

[DKS90] A. Demers, S. Keshav, S. Shenker, "Analysis and simulation of a fair queuing algorithm", Internetworking Research and Experience, vol. 1, pp3-26, Jan 1990.

[DT97] M. Decina, V. Trecordi, "Convergence of Telecommunications and Computing to Networking Models for Integrated Services and Applications", Proceedings of the IEEE, vol. 85 no. 12, Dec 1997.

[FJ95] S. Floyd, V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks", IEEE/ACM Transactions on Networking, Vol. 3 No. 4, pp365-386, Aug 1995.

[INTSERVQM] D. Clark, "The Quality Management Interface", slides presented at the 31st IETF meeting, Jan 1995.

[IPv4] J. Postel, "Internet Protocol", RFC791, Sep 1981

[JDSZ95] S. Jamin, P. Danzig, S. Shenker, L. Zhang, "A Measurement-based Admission Control

Algorithm for Integrated Services Packet Networks", Proc. ACM SIGCOMM'95, pp2-13, Sep 1995.

[KLS98] V. P. Kumar, T. V. Lakshman, D, Stiliadis, "Beyond Best Effort: Architectures for the Differentiated Services of Tomorrow's Internet", IEEE Communications, no. 5, vol. 36, pp151-164, May 1998

[KS98] S. Keshav, R. Sharma, "Issues and Trends in Router Design", IEEE Communications, no. 5, vol. 36, pp144-151, May 1998

[LANEv2] "LANE v2.0 LUNI Interface", ATM Forum, af-lane-0084.000, Jul 1997.

[LLB97] S. Lu, K.-W. Lee, V. Bharghavan, "Adaptive Service in Mobile Computing Environments", in Building QoS into Distributed Systems, (A. Campbell, K. Nahrstedt, Eds), pp25-36, [Chapman & Hall] 1997

[LS98] T. V. Lakshman, D. Stiliadis, "Packet Classification Algorithms for Gigabit Internet Routers", Proc. ACM SIGCOMM'98, Sep 1998

[MPOA] "Multi-Protocol Over ATM Specification v1.0", ATM Forum, af-mpoa-0087.000, Jul 1997.

[Pax97a] V. Paxson, "End-to-End Routing Behavior in the Internet", IEEE/ACM Transactions on Networking, vol. 5 no. 5, pp601-615, Oct. 1997.

[Pax97b] V. Paxson , "End-to-End Internet Packet Dynamics", Proc. ACM SIGCOMM'97, pp139-152, Sep 1997

[PF95] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling", IEEE/ACM Transactions on Networking, vol. 3 no. 3, pp226-244, June 1995.

[PF97] V. Paxson, S. Floyd, "Why we don't know how to simulate the Internet", Proc. 1997 Winter Simulation Conference

[PG93] A. Parekh, R. Gallagher, "A Generalised Processor Sharing Approach to Flow Control in Integrated in Integrated Services Networks – The Single Node Case", ACM/IEEE Transactions on Networking, vol. 1 no. 3 1993.

[PG94] A. Parekh, R. Gallagher, “A Generalised Processor Sharing Approach to Flow Control in Integrated in Integrated Services Networks – The Mutiple Node Case”, ACM/IEEE Transactions on Networking, vol. 2 no. 2 1994.

[RFC1483] J. Heinanen, “Multiprotocol Encapsulation over ATM Adaptation Layer 5”, RFC1483, Jul 1993.

[RFC1618] W. Simpson, “PPP over ISDN”, RFC1618, May 1994.

[RFC1619] W. Simpson, “PPP over SONET/SDH”, RFC1619, May 1994.

[RFC1633] B. Braden, D. Clark, S. Shenker, “Integrated Services in the Internet Architecture: An Overview”, RFC1633, Jun 1994.

[RFC1661] W. Simpson (Ed), “The Point-to-Point Protocol (PPP)”, RFC1661 Jul 1994.

[RFC1821] Borden, Crawley, Davie, Batsell, “Integration of Real-time Services in an IP-ATM Network Architecture”, RFC1821, Aug 1995.

[RFC1825] R. Atkinson, “Security Architecture for the Internet Protocol”, RFC1825, Aug 1995.

[RFC1826] R. Atkinson, “IP Authentication Header”, RFC1826, Aug 1995.

[RFC1827] R. Atkinson, “IP Encapsulating Security Payload (ESP)”, RFC1827, Aug 1995.

[RFC1828] P. Metzger, W. Simpson,“IP Authentication using Keyed MD5”, RFC1828, Aug 1995.

[RFC1829] P. Karn, P. Metzger, W. Simpson, “The ESP DES-CBC Transform”, RFC1829, Aug 1995.

[RFC1958] B. Carpenter, “Architectural Principles of the Internet”, RFC1958, Jun 1996.

[RFC1973] W. Simpson, “PPP in Frame Relay”, RFC1973, Jun 1996.

[RFC2208] A. Mankin, F. Baker, B. Braden, S. Bradner, M. O’Dell, A. Romanow, A. Weinrib, L. Zhang, “Resource ReSerVation Protocol (RSVP) – Version 1 Applicability Statement Some Guidelines on Deployment”, RFC2208, Sep 1997.

[RFC2210] J. Wroclawski, “The Use of RSVP with IETF Integrated Services”, RFC2210, Sep 1997.

[RFC2211] J. Wroclawski, “Specification of the Controlled-Load Network Element Service”, RFC2211, Sep 1997.

[RFC2212] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", RFC2212, Sep 1997.

[RFC2213] F. Baker, J. Krawczyk, A. Sastry, "Integrated Services Management Information Base using SMIv2", RFC2213 Sep 1997.

[RFC2214] F. Baker, J. Krawczyk, A. Sastry, "Integrated Services Management Information Base Guaranteed Service Extensions using SMIv2", RFC2214, Sep 1997.

[RFC2215] S. Shenker, J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC2215, Sep 1997.

[RFC2216] S. Shenker, J. Wroclawski, "Network Element Service Specification Template", RFC2216, Sep 1997.

[RFC2225] M. Laubach, J. Halpern, "Classical IP and ARP over ATM", RFC2225, Apr 1998.

[RSVP] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification", RFC2205, Sep 1997.

[SB95] S. Shenker, L. Breslau, "Two Issues in reservation Establishment", Proc ACM SIGCOMM'95, pp14-26, Sep 1995.

[She95] S. Shenker, "Fundamental Design Issues for the Future Internet", IEEE Journal of Selected areas in Communication, no.13, pp1141-1149, 1995.

[TW96] D. L. Tennenhouse, D. J. Wetherall, "Towards an Active Networking Architecture", ACM Computer Communications Review, no. 2 vol. 26, Apr 1996

[VTP97] G. Varghese, J. Turner, B. Plattner, "Scalable High Speed IP Routing Table Lookups", Proc. ACM SIGCOMM'97, pp25-36, Sep 1997

[WGCJF95] I. Wakeman, A. Ghosh, J. Crowcroft, V. Jacobson, S. Floyd, "Implementing Real Time Packet Forwarding Policies using Streams", Proc. USENIX'95, New Orleans, Louisiana, USA pp71-82, Jan 1995.

[WGS97] L. Wolf, C. Gridwodz, R. Steinmetz, "Multimedia Communication", Proceedings of the IEEE, vol. 85 no. 12, pp-1915-1933, Dec 1997.

[WLG98] D. Wetherall, U. Legedza, J. Guttag, “Introducing New Internet Services: Why and How”, IEEE Network, no. 3 vol. 12, May/Jun 1998

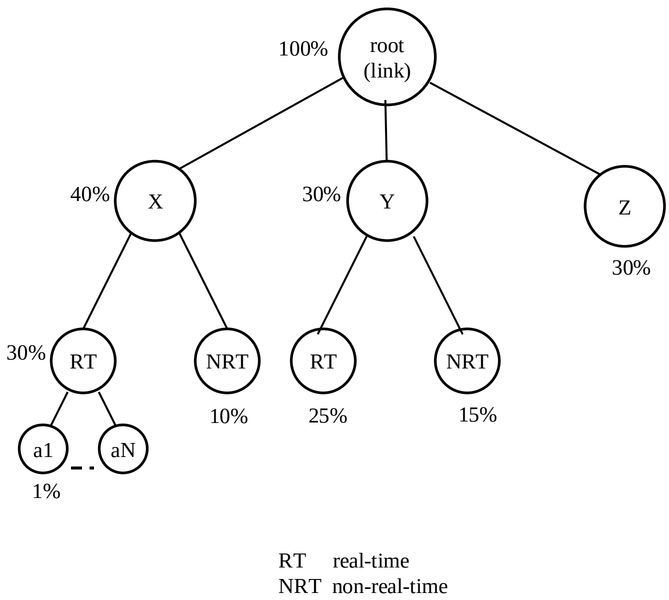


Figure 1: Example class hierarchy for link-sharing

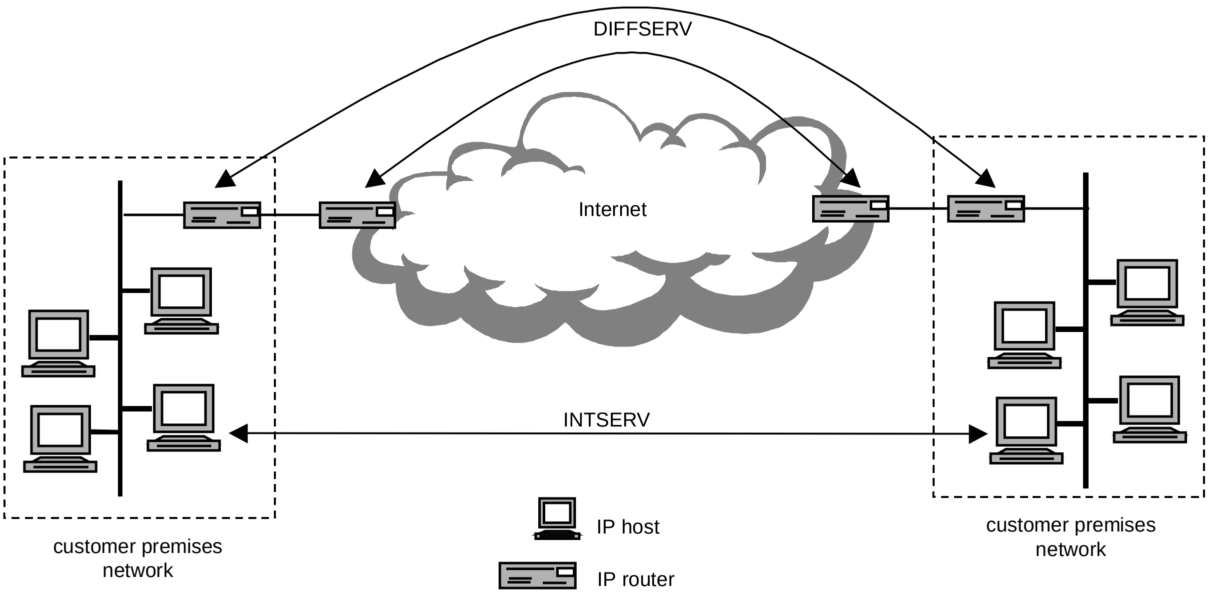


Figure 2: Scope of INTSERV and DIFFSERV

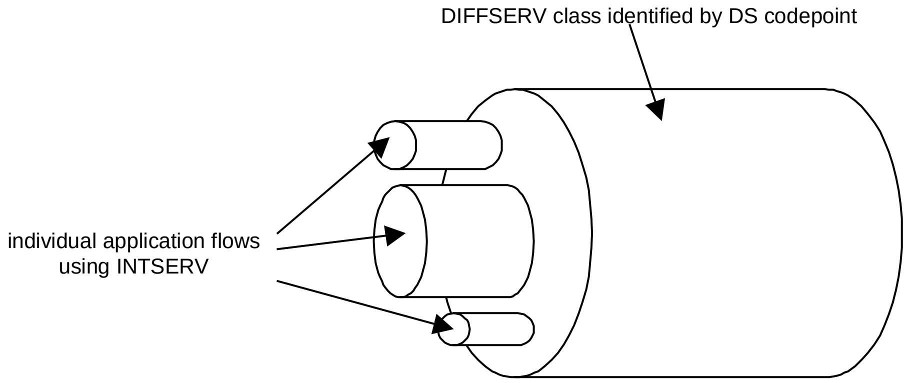
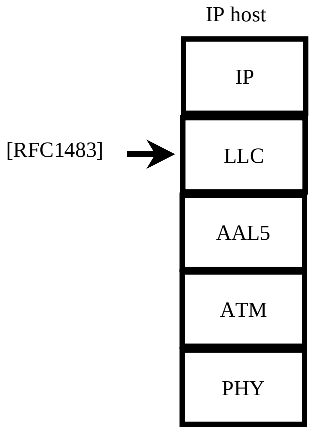


Figure 3: Conceptual view of INTSERV reservations within a DIFFSERV class



AAL	ATM adaptation layer
ATM	asynchronous transfer mode
IP	Internet Protocol
LLC	logical link control
PHY	physical layer

Figure 4: Protocol stack for Classical IP over ATM (CIPA) [RFC2225]

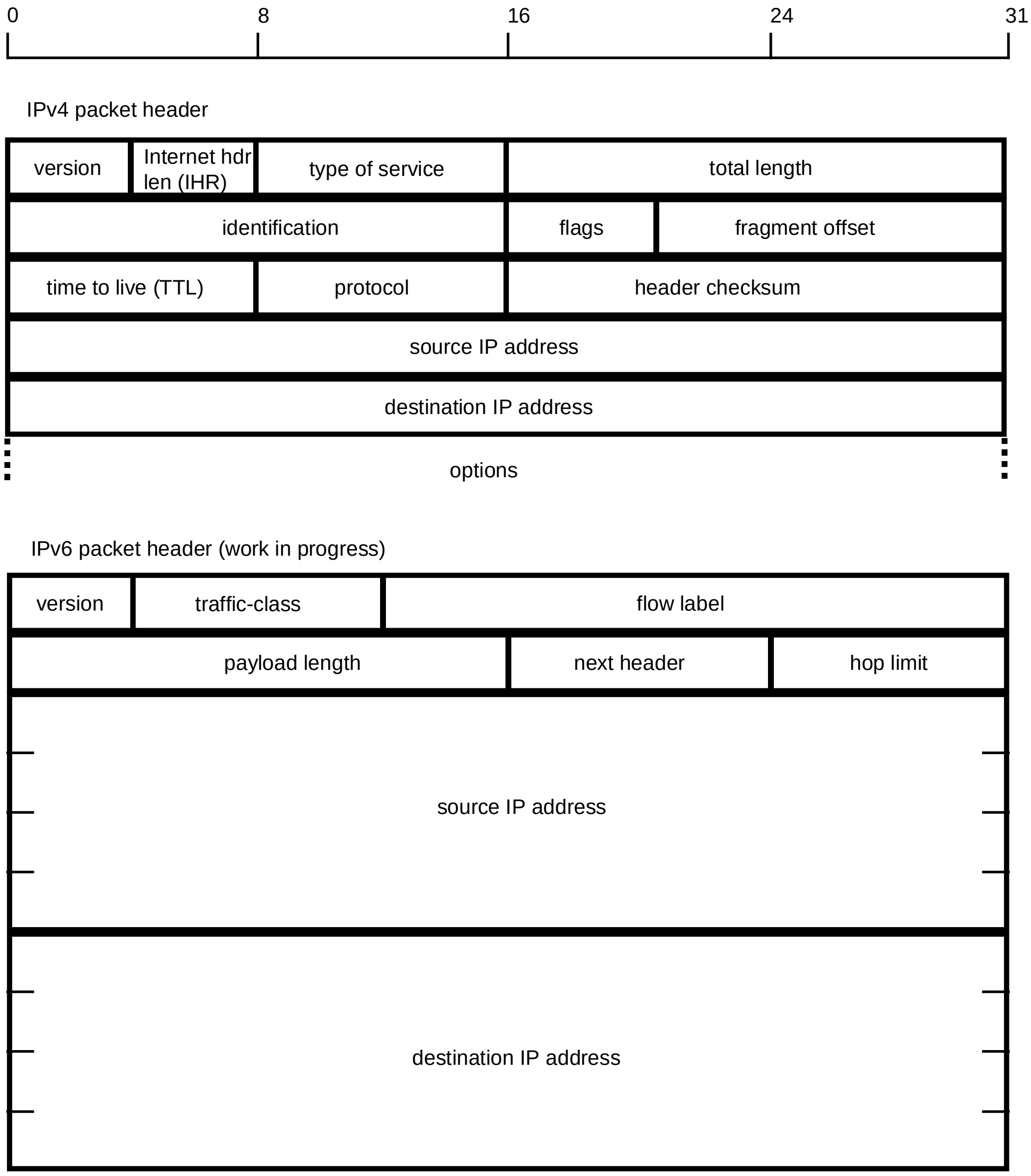


Figure 5: IPv4 packet header (top) and IPv6 packet header (bottom)