Internet Draft
Expires November 1997
draft-ietf-issll-802-01.txt

Mick Seaman
3Com
Andrew Smith
Extreme Networks
Eric Crawley
Gigapacket Networks
June 1997

Integrated Services over IEEE 802.1D/802.1p Networks

Status of this Memo

Abstract

This document describes the support of IETF Integrated Services over LANs built from IEEE 802 network segments which may be interconnected by draft standard IEEE P802.1p switches.

It describes the practical capabilities and limitations of this technology for supporting Controlled Load [8] and Guaranteed Service [9] using the inherent capabilities of the relevant 802 technologies [5],[6] etc. and the proposed 802.1p queuing features in switches. IEEE P802.1p [2] is a superset of the existing IEEE 802.1D bridging specification. This document provides a functional model for the layer 3 to layer 2 and user-to-network dialogue which supports admission control and defines requirements for interoperability between switches. The special case of such networks where the sender and receiver are located on the same segment is also discussed.

This scheme expands on the ISSLL over 802 LANs framework described in

[7]. It makes reference to an admission control signaling protocol developed by the ISSLL WG which is known as the "Subnet Bandwidth Manager". This is an extension to the IETF's RSVP protocol [4] and is described in a separate document [10].

1. Introduction

The IEEE 802.1 Interworking Task Group is currently enhancing the basic MAC Service provided in Bridged Local Area Networks (aka "switched LANs"). As a supplement to the original IEEE MAC Bridges standard [1], the update P802.1p [2] proposes differential traffic class queuing and access to media on the basis of a "user_priority" signaled in frames.

In this document we
* review the meaning and use of user_priority in LANs and the frame forwarding capabilities of a standard LAN switch.
* examine alternatives for identifying layer 2 traffic flows for admission control.
* review the options available for policing traffic flows.
* derive requirements for consistent traffic class handling in a network of switches and use these requirements to discuss queue handling alternatives for 802.1p and the way in which these meet administrative and interoperability goals.
* consider the benefits and limitations of this switched-based approach, contrasting it with full router based RSVP implementation in terms of complexity, utilisation of transmission resources and administrative controls.

The model used is outlined in the "framework document" [7] which in summary:
* partitions the admission control process into two separable operations:
* an interaction between the user of the integrated service and the local network elements ("provision of the service" in the terms of 802.1D) to confirm the availability of transmission resources for traffic to be introduced.
* selection of an appropriate user_priority for that traffic on the basis of the service and service parameters to be supported.
* distinguishes between the user to network interface above and the mechanisms used by the switches ("support of the service"). These include communication between the switches (network to network

signaling).
* describes a simple architecture for the provision and support of these services, broken down into components with functional and interface descriptions:
* a single "user" component: a layer-3 to layer-2 negotiation and

translation component for both sending and receiving, with interfaces to other components residing in the station.
* processes residing in a bridge/switch to handle admission control and mapping requests, including proposals for actual traffic mappings to user_priority values.
* identifies a need for a signaling protocol to carry admission control requests between devices.

It will be noted that this document is written from the pragmatic viewpoint that there will be a widely deployed network technology and we are evaluating it for its ability to support some or all of the defined IETF integrated services: this approach is intended to ensure development of a system which can provide useful new capabilities in existing (and soon to be deployed) network infrastructures.

2. Goals and Assumptions

It is assumed that typical subnetworks that are concerned about quality-of-service will be"switch-rich": that is to say most communication between end stations using integrated services support will pass through at least one switch. The mechanisms and protocols described will be trivially extensible to communicating systems on the same shared media, but it is important not to allow problem generalisation to complicate the practical application that we target: the access characteristics of Ethernet and Token-Ring LANs are forcing a trend to switch-rich topologies along with MAC enhancements to ensure access predictability on half-duplex switch to switch links.

Note that we illustrate most examples in this document using RSVP as an "upper-layer" QoS signaling protocol but there are actually no real dependencies on this protocol: RSVP could be replaced by some other dynamic protocol or else the requests could be made by network management or other policy entities. In any event, no extra modifications to the RSVP protocol are assumed.

There may be a heterogeneous mixture of switches with different capabilities, all compliant with IEEE 802.1p, but implementing queuing and forwarding mechanisms in a range from simple 2-queue per port, strict priority, up to more complex multi-queue (maybe even one per-flow) WFQ or other algorithms.

The problem is broken down into smaller independent pieces: this may lead to sub-optimal usage of the network resources but we contend that such benefits are often equivalent to very small improvements in network efficiency in a LAN environment. Therefore, it is a goal that the switches in the network operate using a much simpler set of information than the RSVP engine in a router. In particular, it is assumed that such

switches do not need to implement per-flow queuing and policing (although they might do so).

It is a fundamental assumption of the int-serv model that flows are isolated from each other throughout their transit across a network. Intermediate queueing nodes are expected to police the traffic to ensure that it conforms to the pre-agreed traffic flow specification. In the architecture proposed here for mapping to layer-2, we diverge from that assumption in the interests of simplicity: the policing function is assumed to be implemented in the transmit schedulers of the layer-3 devices (end stations, routers). In the LAN environments envisioned, it is reasonable to assume that end stations are "trusted" to adhere to their agreed contracts at the inputs to the network and that we can afford to over-allocate resources at admission -control time to compensate for the inevitable extra jitter/bunching introduced by the switched network itself.

These divergences have some implications on the receiver heterogeneity that can be supported and the statistical multiplexing gains that might have been exploited, especially for Controlled Load flows.

3. User Priority and Frame Forwarding in IEEE 802 Networks

3.1 General IEEE 802 Service Model

User_priority is a value associated with the transmission and reception of all frames in the IEEE 802 service model: it is supplied by the sender which is using the MAC service. It is provided along with the

data to a receiver using the MAC service. It may or may not be actually carried over the network: Token- Ring/802.5 carries this value (encoded in its FC octet), basic Ethernet/802.3 does not. 802.1p defines a way to carry this value over the network in a consistent way on Ethernet, Token Ring, FDDI or other MAC-layer media using an extended frame format. The usage of user_priority is summarised below but is more fully described in section 2.5 of 802.1D [1] and 802.1p [2] "Support of the Internal Layer Service by Specific MAC Procedures" and readers are referred to these documents for further information.

If the "user_priority" is carried explicitly in packets, its utility is as a simple label in the data stream enabling packets in different classes to be discriminated easily by downstream nodes without their having to parse the packet in more detail.

Apart from making the job of desktop or wiring-closet switches easier, an explicit field means they do not have to change hardware or software as the rules for classifying packets evolve (e.g. based on new protocols or new policies). More sophisticated layer-3 switches, perhaps deployed

towards the core of a network, can provide added value here by performing the classification more accurately and, hence, utilising network resources more efficiently or providing better protection of flows from one another: this appears to be a good economic choice since there are likely to be very many more desktop/wiring closet switches in a network than switches requiring layer-3 functionality.

The IEEE 802 specifications make no assumptions about how user_priority is to be used by end stations or by the network. In particular it can only be considered a "priority" in a loose sense: although the current 802.1p draft defines static priority queuing as the default mode of operation of switches that implement multiple queues (user_priority is defined as a 3-bit quantity so strict priority queueing would give value 7 = high priority, 0 = low priority). The general switch algorithm is as follows: packets are placed onto a particular queue based on the received user_priority (from the packet if a 802.1p header or 802.5 network was used, invented according to some local policy if not). The selection of queue is based on a mapping from user_priority [0,1,2,3,4,5,6 or 7] onto the number of available queues.Note that switches may implement any number of queues from 1 upwards and it may not be visible externally, except through any advertised switch parameters and the its admission control behaviour, which user_priority

values get mapped to the same vs. Different queues internally.Other algorithms that a switch might implement might include e.g. weighted fair queueuing, round robin.

In particular, IEEE makes no recommendations about how a sender should select the value for user_priority: one of the main purposes of this current document is to propose such usage rules and how to communicate the semantics of the values between switches, end- stations and routers. For the remainder of this document we use the term "traffic class" when discussing the treatment of packets with one of the user_priority values.

3.2 Ethernet/802.3

There is no explicit traffic class or user_priority field carried in Ethernet packets. This means that user_priority must be regenerated at a downstream receiver or switch according to some defaults or by parsing further into higher-layer protocol fields in the packet. Alternatively, the IEEE 802.1Q encapsulation [11] may be used which provides an explicit traffic class field on top of an basic MAC format.

For the different IP packet encapsulations used over Ethernet/802.3, it will be necessary to adjust any admission- control calculations according to the framing and to the padding requirements:

Encapsulation	Framing Overhead bytes/pkt	IP MTU bytes
IP EtherType (ip_len<=46 bytes)	64-ip_len	1500
(1500>=ip_len>=46 bytes)	18	1500
IP EtherType over 802.1p/Q (ip_len<=42)	64-ip_len	1500*
(1500>=ip_len>=42 bytes)	22	1500*
IP EtherType over LLC/SNAP (ip_len<=40)	64-ip_len	1492
(1500>=ip_len>=40 bytes)	24	1492

* note that the draft IEEE 802.1Q specification exceeds the IEEE 802.3 maximum packet length values by 4 bytes.

3.3 Token-Ring/802.5

The token ring standard [6] provides a priority mechanism that can be used to control both the queuing of packets for transmission and the access of packets to the shared media. The priority mechanisms are implemented using bits within the Access Control (AC) and the Frame Control (FC) fields of a LLC frame. The first three bits of the AC field, the Token Priority bits, together with the last three bits of the AC field, the Reservation bits, regulate which stations get access to the ring. The last three bits of the FC field of an LLC frame, the User Priority bits, are obtained from the higher layer in the user_priority parameter when it requests transmission of a packet. This parameter also establishes the Access Priority used by the MAC. The user_priority value is conveyed end-to-end by the User Priority bits in the FC field and is typically preserved through Token-Ring bridges of all types. In all cases, 0 is the lowest priority.

Token-Ring also uses a concept of Reserved Priority: this relates to the value of priority which a station uses to reserve the token for the next transmission on the ring. When a free token is circulating, only a station having an Access Priority greater than or equal to the Reserved Priority in the token will be allowed to seize the token for transmission. Readers are referred to [14] for further discussion of this topic.

A token ring station is theoretically capable of separately queuing each of the eight levels of requested user priority and then transmitting frames in order of priority. A station sets Reservation bits according to the user priority of frames that are queued for transmission in the highest priority queue. This allows the access mechanism to ensure that the frame with the highest priority throughout the entire ring will be transmitted before any lower priority frame. Annex I to the IEEE 802.5 token ring standard recommends that stations send/relay frames as

follows:

Application	user_priority
non-time-critical data	0
-	1
-	2
-	3
LAN management	4

time-sensitive data	5
real-time-critical data	6
MAC frames	7

To reduce frame jitter associated with high-priority traffic, the annex also recommends that only one frame be transmitted per token and that the maximum information field size be 4399 octets whenever delay-sensitive traffic is traversing the ring. Most existing implementations of token ring bridges forward all LLC frames with a default access priority of 4. Annex I recommends that bridges forward LLC frames that have a user priorities greater that 4 with a reservation equal to the user priority (although the draft IEEE P802.1p [2] permits network management override this behaviour). The capabilities provided by token ring's user and reservation priorities and by IEEE 802.1p can provide effective support for Integrated Services flows that request QoS using RSVP. These mechanisms can provide, with few or no additions to the token ring architecture, bandwidth guarantees with the network flow control necessary to support such guarantees.

For the different IP packet encapsulations used over Token Ring/802.5, it will be necessary to adjust any admission-control calculations according to the framing requirements:

Encapsulation	Framing Overhead bytes/pkt	IP MTU bytes
IP EtherType over 802.1p/Q	29	4370*
IP EtherType over LLC/SNAP	25	4370*

*the suggested MTU from RFC 1042 [13] is 4464 bytes but there are issues related to discovering what the maximum supported MTU between any two points both within and between Token Ring subnets. We recommend here an MTU consistent with the 802.5 Annex I recommendation.

4. Integrated services through layer-2 switches

4.1 Summary of switch characteristics

For the sake of illustration, we divide layer-2 bridges/switches into several categories, based on the level of sophistication of their QoS

and software protocol capabilities: these categories are not intended to

represent all possible implementation choices but, instead, to aid discussion of what QoS capabilities can be expected from a network made of these devices.

Class I - 802.1p priority queueuing between traffic classes.
- No multicast heterogeneity.
- 802.1p GARP/GMRP pruning of individual multicast addresses.

Class II As (I) plus:
- can map received user_priority on a per-input-port basis to some internal set of canonical values.
- can map internal canonical values onto transmitted user_priority on a per-output-port basis giving some limited form of multicast heterogeneity.
- maybe implements IGMP snooping for pruning.

Class III As (II) plus:
- per-flow classification
- maybe per-flow policing and/or reshaping
- WFQ or other transmit scheduling (probably not per-flow) 4.2 Queueing

Connectionless packet-based networks in general, and LAN-switched networks in particular, work today because of scaling choices in network provisioning. Consciously or (more usually) unconsciously, enough excess bandwidth and buffering is provisioned in the network to absorb the traffic sourced by higher-layer protocols or cause their transmission windows to run out, on a statistical basis, so that the network is only overloaded for a short duration and the average expected loading is less than 60% (usually much less).

With the advent of time-critical traffic such overprovisioning has become far less easy to achieve. Time critical frames may find themselves queued for annoyingly long periods of time behind temporary bursts of file transfer traffic, particularly at network bottleneck points, e.g. at the 100 Mb/s to 10 Mb/s transition that might occur between the riser to the wiring closet and the final link to the user from a desktop switch. In this case, however, if it is known (guaranteed by application design, merely expected on the basis of statistics, or just that this is all that the network guarantees to support) that the time critical traffic is a small fraction of the total bandwidth, it suffices to give it strict priority over the "normal" traffic. The worst case delay experienced by the time critical traffic is roughly the maximum transmission time of a maximum length non-time-critical frame - less than a millisecond for 10 Mb/s Ethernet, and well below an end to end budget based on human perception times.

When more than one "priority" service is to be offered by a network element e.g. it supports Controlled-Load as well as Guaranteed Service, the queuing discipline becomes more complex. In order to provide the required isolation between the service classes, it will probably be necessary to queue them separately. There is then an issue of how to service the queues - a combination of admission control and maybe weighted fair queuing may be required in such cases. As with the service specifications themselves, it is not the place for this document to specify queuing algorithms, merely to observe that the external behaviour meet the services' requirements.

4.3 Multicast Heterogeneity

IEEE 802.1D and 802.1p specify a basic model for multicast whereby a switch performs multicast routing decisions based on the destination address: this would produce a list of output ports to which the packet should be forwarded. In its default mode, such a switch would use any user_priority value in received packets to enqueue the packets at each output port. All of the classes of switch identified above can support this operation.

At layer-3, the int-serv model allows heterogeneous multicast flows where different branches of a tree can have different types of reservations for a given multicast destination, or even supports the notion that some trees will have some branches with reserved flows and some using best effort (default) service.

If a switch is selecting per-port output queues based only on the incoming user_priority, as described by 802.1p, it must treat all branches of all multicast sessions within that user_priority class with the same queuing mechanism: no heterogeneity is then possible.I If a switch were to implement a separate user_priority mapping at each output port, as described under "Class II switch" above, then some limited form of receiver heterogeneity can be supported e.g. forwarding of traffic as user_priority 4 on one branch where receivers have performed admission control reservations and as user_priority 0 on one where they have not. We assume that per-user_priority queuing without taking account of input or output ports is the minimum standard functionality for systems in a LAN environment (Class I switch, as defined above). More functional layer-2 switches or even layer-3 switches (a.k.a. routers) can be used if even more flexible forms of heterogeneity are considered necessary: their behaviour is well standardised.

4.4 Override of incoming user_priority

In some cases, a network administrator may not trust the user_priority

values contained in packets from a source and may which to map these into some more suitable set of values. Alternatively, due perhaps to

equipment limitations or transition periods, values may need to be mapped to/from different regions of a network.

Some switches may implement such a function on input that maps received user_priority into some internal set of values (this table is known in 802.1p as the "user_priority regeneration table"). These values can then be mapped using the output table described above onto outgoing user_priority values: these same mappings must also be used when applying admission control to requests that use the user_priority values (see e.g. [10]). More sophisticated approaches may also be envisioned where a device polices traffic flows and adjusts their onward user_priority based on their conformance to the admitted traffic flow specifications.

4.5 Remapping of non-conformant aggregated flows

One other topic under discussion in the int-serv context is how to handle the traffic for data flows from sources that are exceeding their currently agreed traffic contract with the network. An approach that shows much promise is to treat such traffic with "somewhat less than best effort" service in order to protect traffic that is normally given "best effort" service from having to back off (such traffic is often "adaptive" using TCP or other congestion control algorithms and it would be unfair to penalise it due to badly behaved traffic from reserved flows which are usually set up by non-adaptive applications).

A solution here might be to assign normal best effort traffic to one user_priority and to label excess non-conformant traffic as a "lower" user_priority. This topic is further discussed below.

5. Selecting traffic classes

One fundamental question is "who gets to decide what the classes mean and who gets access to them?" One approach would be for the meanings of the classes to be "well-known": we would then need to standardise a set of classes e.g. 1 = best effort, 2 = controlled- load, 3 = guaranteed (loose delay bound, high bandwidth), 4 = guaranteed (slightly tighter delay) etc. The values to encode in such a table in end stations, in

isolation from the network to which they are connected, is problematical: one approach could be to define one user_priority value per int-serv service and leave it at that (reserving the rest of the combinations for future traffic classes - there are sure to be plenty!).

We propose here a more flexible mapping: clients ask "the network" which user_priority traffic class to use for a given traffic flow, as categorised by its flow-spec and layer-2 endpoints. The network provides a value back to the requester which is appropriate to the current

network topology, load conditions, other admitted flows etc. The task of configuring switches with this mapping (e.g. through network management, a switch-switch protocol or via some network-wide QoS-mapping directory service) is an order of magnitude less complex than performing the same function in end stations. Also, when new services (or other network reconfigurations) are added to such a network, the network elements will typically be the ones to be upgraded with new queuing algorithms etc. and can be provided with new mappings at this time.

Given the need for a new session or "flow" requiring some QoS support, a client then needs answers to the following questions:

1. which traffic class do I add this flow to?
 The client needs to know how to label the packets of the flow as it places them into the network.

2. who do I ask/tell?
 The proposed model is that a client ask "the network" which user_priority traffic class to use for a given traffic flow. This has several benefits as compared to a model which allows clients to select a class for themselves.

3. how do I ask/tell them?
 A request/response protocol is needed between client and network: in fact, the request can be piggy-backed onto an admission control request and the response can be piggy-backed onto an admission control acknowledgment: this "one pass" assignment has the benefit of completing the admission control in a timely way and reducing the exposure to changing conditions which could occur if clients cached the knowledge for extensive periods.

The network (i.e. the first network element encountered downstream from

the client) must then answer the following questions:

1. which traffic class do I add this flow to?
 This is a packing problem, difficult to solve in general, but many simplifying assumptions can be made: presumably some simple form of allocation can be done without a more complex scheme able to dynamically shift flows around between classes.

2. which traffic class has worst-case parameters which meet the needs of this flow?
 This might be an ordering/comparison problem: which of two service classes is "better" than another? Again, we can make this tractable by observing that all of the current int-serv classes can be ranked (best effort <= Controlled Load <= Guaranteed Service) in a simple manner. If any classes are implemented in the future that cannot be simply ranked then the issue can be finessed by either a priori knowledge about what

classes are supported or by configuration.

and return the chosen user_priority value to the client.

Note that the client may be either an end station, router or a first switch which may be acting as a proxy for a client which does not participate in these protocols for whatever reason. Note also that a device e.g. a server or router, may choose to implement both the "client" as well as the "network" portion of this model so that it can select its own user_priority values: such an implementation would, however, be discouraged unless the device really does have a close tie-in with the network topology and resource allocation policies but would work in some cases where there is known over- provisioning of resources.

6. Flow Identification

Several previous proposals for int-serv over lower-layers have treated switches very much as a special case of routers: in particular, that switches along the data path will make packet handling decisions based on the RSVP flow and filter specifications and use them to classify the corresponding data packets. However, filtering to the per-flow level becomes cost-prohibitive with increasing switch speed: devices with such filtering capabilities are unlikely to have a very different implementation cost to IP routers, in which case we must question

whether a specification oriented toward switched networks is of any benefit at all.

This document proposes that "aggregated flow" identification based on user_priority be the minimum required of switches.

7. Reserving Network Resources - Admission Control

So far we have not discussed admission control. In fact, without admission control it is possible to scratchbuild a LAN network of some size capable of supporting real-time services, providing that the traffic fits within certain scaling constraints (relative link speeds, numbers of ports etc. - see below). This is not surprising since it is possible to run a fair approximation to real time services on small LANs today with no admission control or help from encoded priority bits.

Imagine a campus network providing dedicated 10 Mbps connections to each user. Each floor of each building supports up to 96 users, organized into groups of 24, with each group being supported by a 100 Mbps downlink to a basement switch which concentrates 5 floors (20 x 100 Mbps) and a data center (4 x 100 Mbps) to a 1 Gbps link to an 8 Gbps central campus switch, which in turn hooks 6 buildings together (with 2

x 1 Gbps full duplex links to support a corporate server farm). Such a network could support 1.5 Mb/s of voice/video from every user to any other user or (for half the population) the server farm, provided the video ran high priority: this gives 3000 users, all with desktop video conferencing running along with file transfer/email etc. In such a network RSVP's role would be limited to ensuring resource availability at the communicating end stations and for connection to the wide area.

In such a network, a discussion as to the best service policy to apply to high and low priority queues may prove academic: while it is true that "normal" traffic may be delayed by bunches of high priority frames, queuing theory tells us that the average queue occupancy in the high priority queue at any switch port will be somewhat less than 1 (with real user behaviour, i.e. not all watching video conferences all the time) it should be far less. A cheaper alternative to buying equipment with a fancy queue service policy may be to buy equipment with more bandwidth to lower the average link utilisation by a few per cent.

In practice a number of objections can be made to such a simple solution. There may be long established expensive equipment in the network which does not provide all the bandwidth required. There will be considerable concern over who is allowed to say what traffic is high priority. There may be a wish to give some form of "prioritised" service to crucial business applications, above that given to experimental video-conferencing. The task that faces us is to provide a degree of control without making that control so elaborate to implement that the control-oriented solution is not simply rejected in favor of providing yet more bandwidth, at a lower cost.

The proposed admission control mechanism requires a query-response interaction with the network returning a "YES/NO" answer and, if successful, a user_priority value with which to tag the data frames of this flow.

The relevant int-serv specifications describe the parameters which need to be considered when making an admission control decision at each node in the network path between sender and receiver. We discuss how to calculate these parameters for different network technologies below but we do not specify admission control algorithms or mechanisms as to how to progress the admission control process across the network. One such mechanism is described as SBM in [10].

Where there are multiple mechanisms in use for allocating resources e.g. some combination of SBM and network management, it will be necessary to ensure that network resources are partitioned amongst the different mechanisms in some way: this could be by configuration or maybe by having the mechanisms allocate from a common resource pool within any device.

---

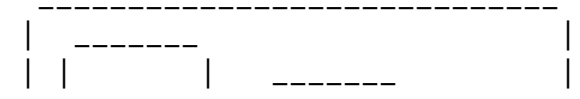
8. Mapping of integrated services to layer-2 in layer-3 devices

8.1 Layer-3 client

We assume the same client model as int-serv and RSVP where we use the term "client" to mean the entity handling QoS in the layer-3 device at each end of a layer-2 hop (e.g. end-station, router). The sending client itself is responsible for local admission control and scheduling packets onto its link in accordance with the service agreed. Just as in the int-serv model, this involves per-flow schedulers (a.k.a. shapers) in every such data source.

The client is running an RSVP process which presents a session establishment interface to applications, signals RSVP over the network, programs a scheduler and classifier in the driver and interfaces to a policy control module. In particular, RSVP also interfaces to a local admission control module: it is this entity that we focus on here.

The following diagram is taken from the RSVP specification [4]:



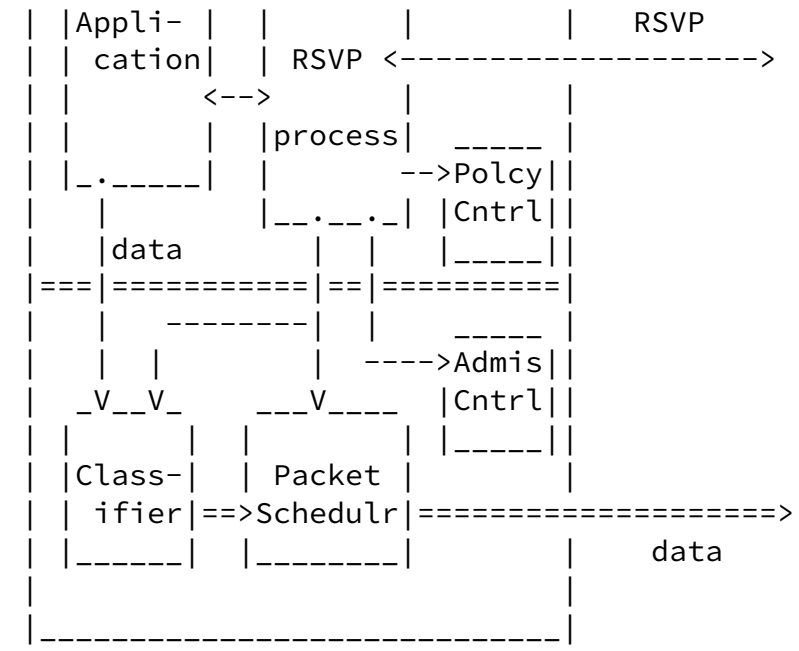


Figure 1 - RSVP in Sending Hosts

Note that we illustrate examples in this document using RSVP as the "upper-layer" signaling protocol but there are no actual dependencies on this protocol: RSVP could be replaced by some other dynamic protocol or else the requests could be made by network management or other policy entities.

8.2 Requests to layer-2

The local admission control entity within a client is responsible for mapping these layer-3 requests into layer-2 language.

The upper-layer entity requests from ISSLL:

"May I reserve for traffic with <traffic characteristic> with <performance requirements> from <here> to <there> and how should I label it?"

where
  <traffic characteristic> = Flow Spec, Tspec, Rspec (e.g. bandwidth, burstiness, MTU etc.)
  <performance requirements> = latency, jitter bounds etc.
  <here> = IP address(es)
  <there> = IP address(es) - may be multicast

8.3 Sender

The ISSLL functionality in the sender is illustrated below and may be summarised as:
* maps the endpoints of the conversation to layer-2 addresses in the LAN, so it can figure out what traffic is really going where (probably makes reference to the ARP protocol cache for unicast or an algorithmic mapping for multicast destinations).
* applies local admission control on outgoing link and driver
* formats a SBM request to the network with the mapped addresses and filter/flow specs
* receives response from the network and reports the YES/NO admission control answer back to the upper layer entity, along with any negotiated modifications to the session parameters.
* stores any resulting user_priority to be associated with this session in a "802 header" lookup table for use when sending any future data packets.

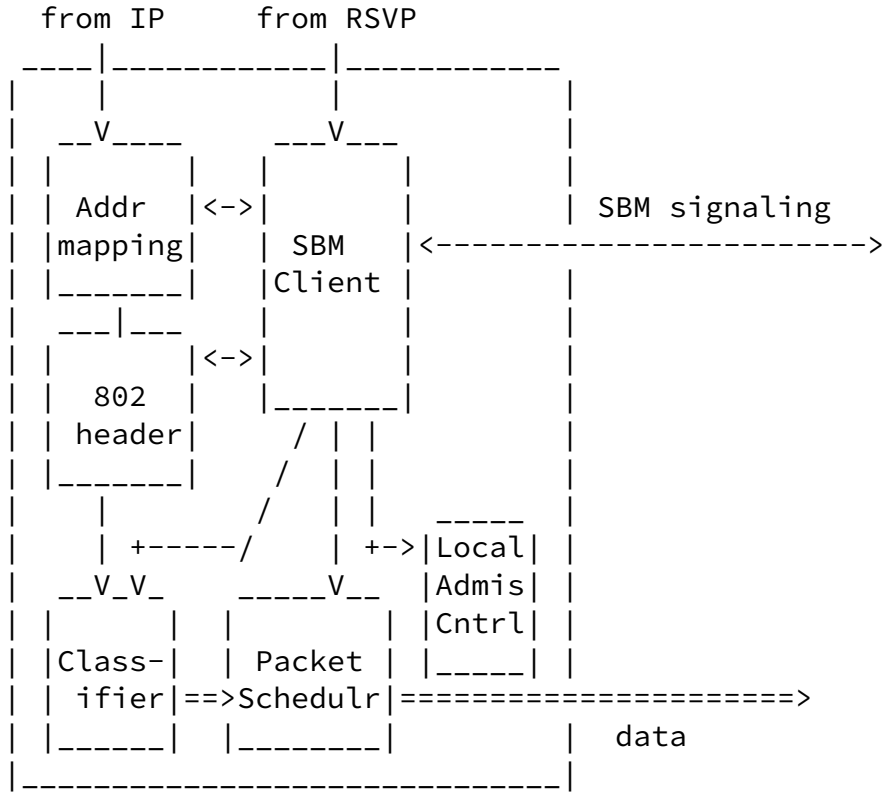


Figure 2 - ISSLL in End-station Sender

ISSLL manageable objects in the sender:
802 header table
Local admission control resource status
L2 additions to classifier/scheduler int-serv tables

8.4 Receiver

The ISSLL functionality in the receiver is a good deal simpler. It is summarised below and is illustrated by the following picture:
* handles any received SBM protocol indications.
* applies local admission control to see if a request can be supported with appropriate local receive resources.
* passes indications up to RSVP if OK.
* accepts confirmations from RSVP and relays them back via SBM signaling towards the requester.
* may program a receive classifier and scheduler, if any is used, to identify traffic classes of received packets and accord them appropriate treatment e.g. reserve some buffers for particular traffic classes.
* programs receiver to strip any 802 header information from received packets.

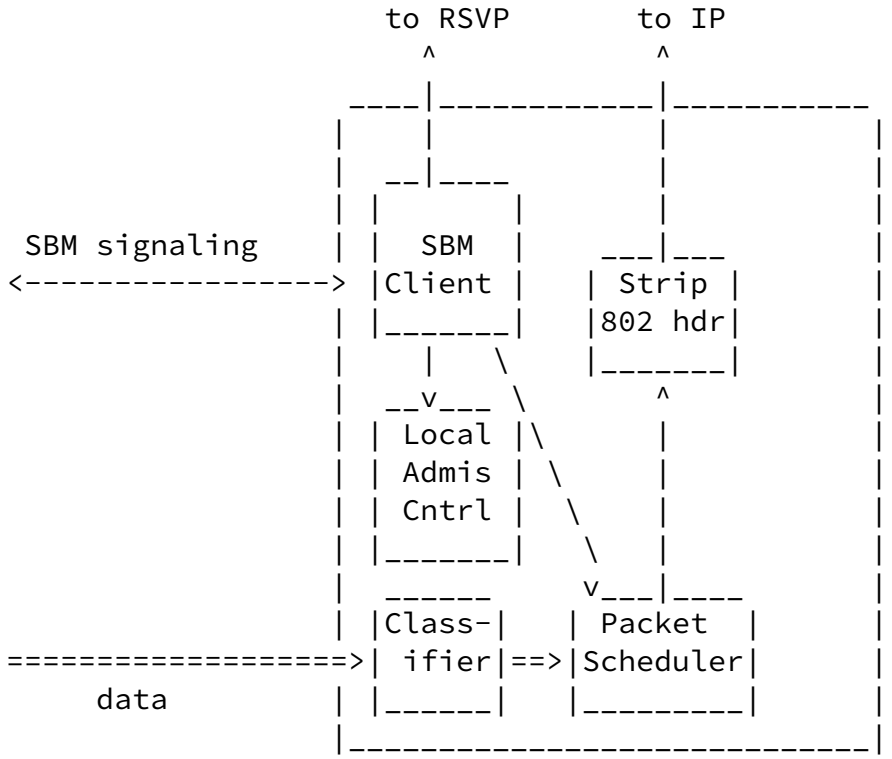


Figure 3 - ISSLL in End-station Receiver

9. Layer-2 Switch Functions

9.1 Switch Model

In this model of layer-2 switch behaviour, we define the following entities within the switch:

* Local admission control - one of these on each port accounts for the

available bandwidth on the link attached to that port. For half-duplex links, this involves taking account of the resources allocated to both transmit and receive flows. For full-duplex, the input port accountant's task is trivial.

* Input SBM module: one instance on each port, performs the "network" side of the signaling protocol for peering with clients or other switches. Also holds knowledge of the mappings of int-serv classes to user_priority.

* SBM propagation - relays requests that have passed admission control at the input port to the relevant output ports' SBM modules. This will require access to the switch's forwarding table (layer-2 "routing table" - cf. RSVP model) and port spanning-tree states.

* Output SBM module - forwards requests to the next layer-2 or -3 network hop.

* Classifier, Queueing and Scheduler - these functions are basically as described by the Forwarding Process of IEEE 802.1p (see section 3.7 of [2]). The Classifier module identifies the relevant QoS information from incoming packets and uses this, together with the normal bridge forwarding database, to decide to which output queue of which output port to enqueue the packet. In Class I switches, this information is the "regenerated user_priority" parameter which has already been decoded by the receiving MAC service and potentially re-mapped by the 802.1p forwarding process (see description in section 3.7.3 of [2]). This does not preclude more sophisticated classification rules which may be applied in more complex Class III switches e.g. matching on individual int-serv flows.

The Queueing and Scheduler module holds the output queues for ports and provides the algorithm for servicing the queues for transmission onto

the output link in order to provide the promised int-serv service. Switches will implement one or more output queues per port and all will implement at least a basic strict priority dequeueing algorithm as their default, in accordance with 802.1p.

* Ingress traffic class mapper and policing - as described in 802.1p section 3.7. This optional module may check on whether the data within traffic classes are conforming to the patterns currently agreed: switches may police this and discard or re-map packets. The default behaviour is to pass things through unchanged.

* Egress traffic class mapper - as described in 802.1p section 3.7. This optional module may apply re-mapping of traffic classes e.g. on a per-output port basis. The default behaviour is to pass things through unchanged.

These are shown by the following diagram which is a superset of the IEEE 802.1D/802.1p bridge model:

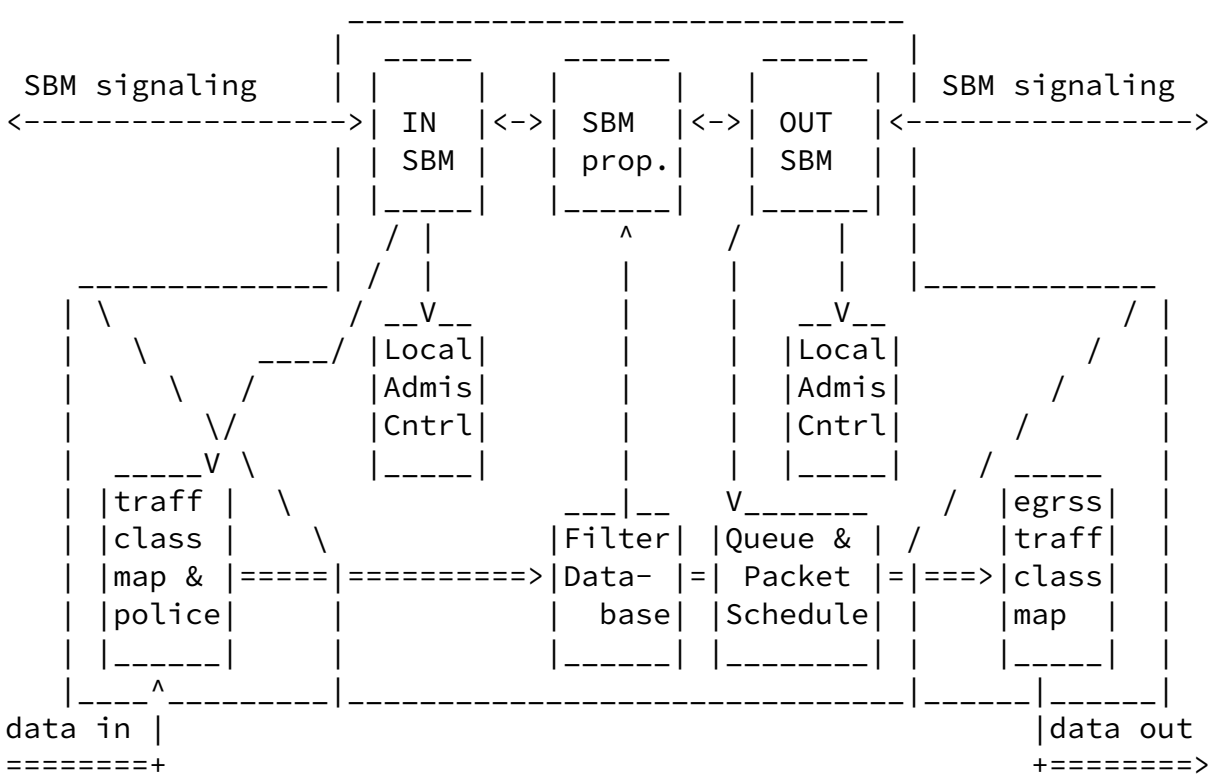


Figure 4 - ISSLL in Switches

9.2 Admission Control

On reception of an admission control request, a switch performs the following actions:
* ingress SBM module translates any received user_priority or else selects a layer-2 traffic class which appears compatible with the request and whose use does not violate any administrative policies in force. In effect, it matches up the requested service with those available in each of the user_priority classes and chooses the "best" one. It ensures that, if this reservation is successful, the selected value is passed back to the client.
* ingress SBM observes the current state of allocation of resources on the input port/link and then determines whether the new resource allocation from the mapped traffic class would be excessive. The request is passed to the reservation propagator if accepted so far.
* reservation propagator relays the request to the bandwidth accountants on each of the switch's outbound links to which this reservation would apply (implied interface to routing/forwarding database).
* egress bandwidth accountant observes the current state of allocation of queueing resources on its outbound port and bandwidth on the link itself and determines whether the new allocation would be excessive. Note that this is only the local decision of this switch hop: each

further layer-2 hop through the network gets a chance to veto the request as it passes along.
* the request, if accepted by this switch, is then passed on down the line on each output link selected. Any user_priority described in the forwarded request must be translated according to any egress mapping table.

* if accepted, the switch must notify the client of the user_priority to use for packets belonging to this flow. Note that this is a "provisional YES" - we assume an optimistic approach here: later switches can still say "NO" later.
* if this switch wishes to reject the request, it can do so by notifying the original client (by means of its layer-2 address).

10. Mappings from intserv service models to IEEE 802

It is assumed that admission control will be applied when deciding

whether or not to admit a new flow through a given network element and that a device sending onto a link will be proxying the parameters and admission control decisions on behalf of that link: this process will require the device to be able to determine (by estimation, measurement or calculation) several parameters. It is assumed that details of the potential flow are provided to the device by some means (e.g. a signaling protocol, network management). The service definition specifications themselves provide some implementation guidance as to how to calculate some of these quantities.

The accuracy of calculation of these parameters may not be very critical: indeed it is an assumption of this model's being used with relatively simple Class I switches that they merely provide values to describe the device and admit flows conservatively.

10.1 General characterisation parameters

There are some general parameters that a device will need to use and/or supply for all service types:
- Ingress link
- Egress links and their MTUs, framing overheads and minimum packet sizes (see media-specific information presented above).
- available path bandwidth: updated hop-by-hop by any device along the path of the flow.
- minimum latency

10.2 Parameters to implement Guaranteed Service

A network element must be able to determine the following parameters:

- Constant delay bound through this device (in addition to any value provided by "minimum latency" above) and up to the receiver at the next network element for the packets of this flow if it were to be admitted: this would include any access latency bound to the outgoing link as well as propagation delay across that link.
- Rate-proportional delay bound through this device and up to the receiver at the next network element for the packets of this flow if it were to be admitted.
- Receive resources that would need to be associated with this flow (e.g. buffering, bandwidth) if it were to be admitted and not suffer packet loss if it kept within its supplied Tspec/Rspec.
- Transmit resources that would need to be associated with this flow

(e.g. buffering, bandwidth, constant- and rate-proportional delay bounds) if it were to be admitted.

10.3 Parameters to implement Controlled Load

A network element must be able to determine the following parameters which can be extracted from [8]:

- Receive resources that would need to be associated with this flow (e.g. buffering) if it were to be admitted.
- Transmit resources that would need to be associated with this flow (e.g. buffering) if it were to be admitted.

10.4 Parameters to implement Best Effort

For a network element to implement best effort service there are no explicit parameters that need to be characterised.

10.5 Mapping to IEEE 802 user_priority

There are many options available for mapping aggregations of flows described by int-serv service models (Best Effort, Controlled Load, and Guaranteed are the services considered here) onto user_priority classes. There currently exists very little practical experience with particular mappings to help make a determination as to the "best" mapping. In that spirit, the following options are presented in order to stimulate experimentation in this area. Note, this does not dictate what mechanisms/algorithms a network element (e.g. an Ethernet switch) needs to perform to implement these mappings: this is an implementation choice and does not matter so long as the requirements for the particular service model are met. Having said that, we do explore below the ability of a switch implementing strict priority queueing to support some or all of the service types under discussion: this is worthwhile because this is likely to be the most widely deployed dequeueing algorithm in simple switches as it is the default specified in 802.1p.

In order to reduce the administrative problems , such a mapping table is held by *switches* (and routers if desired) but generally not by end-station hosts and is a read-write table. The values proposed below are defaults and can be overridden by management control so long as all switches agree to some extent (the required level of agreement requires

further analysis).

It is possible that some form of network-wide lookup service could be implemented that serviced requests from clients e.g. traffic_class = getQoSbyName("H.323 video") and notified switches of what sorts of traffic categories they were likely to encounter and how to allocate those requests into traffic classes: such mechanisms are for further study.

Proposal: A Simple Scheme

user_priority	Service
0	"less than" Best Effort
1	Best Effort
2	reserved
3	reserved
4	Controlled Load
5	Guaranteed Service, 100ms bound
6	Guaranteed Service, 10ms bound
7	reserved

In this proposal, all traffic that uses the controlled load service is mapped to a single 802.1p user_priority whilst that for guaranteed service is placed into one of two user_priority classes with different delay bounds. Unreserved best effort traffic is mapped to another.

The use of classes 4, 5 and 6 for Controlled Load and Guaranteed Service is somewhat arbitrary as long as they are increasing. Any two classes greater than Best Effort can be used as long as GS is "greater" than CL although those proposed here have the advantage that, for transit through 802.1p switches with only two-level strict priority queuing, they both get "high priority" treatment (the current 802.1p default split is 0-3 and 4-7 for a device with 2 queues). The choice of delay bound is also arbitrary but potentially very significant: this can lead to a much more efficient allocation of resources as well as greater (though still not very good) isolation between flows.

The "less than best effort" class might be useful for devices that wish to tag packets that are exceeding a committed network capacity and can be optionally discarded by a downstream device. Note, this is not *required* by any current int-serv models but is under study.

The advantage to this approach is that it puts some real delay bounds on

the Guaranteed Service without adding any additional complexity to the other services. It still ignores the amount of *bandwidth* available for each class. This should behave reasonably well as long as all traffic for CL and GS flows does not exceed any resource capacities in the device. Some isolation between very delay-critical GS and less critical GS flows is provided but there is still an overall assumption that flows will in general be well- behaved. In addition, this mapping still leaves room for future service models.

Expanding the number of classes for CL service is not as appealing since there is no need to map to a particular delay bound. There may be cases where an administrator might map CL onto more classes for particular bandwidths or policy levels. It may also be desirable to further subdivide CL traffic in cases where the itis frequently non-conformant for certain applications.

11. Network Topology Scenarios

11.1 Switched networks using priority scheduling algorithms

In general, the int-serv standards work has tried to avoid any specification of scheduling algorithms, instead relying on implementers to deduce appropriate algorithms from the service definitions and on users to apply measurable benchmarks to check for conformance. However, since one standards' body has chosen to specify a single default scheduling algorithm for switches [2], it seems appropriate to examine to some degree, how well this "implementation" might actually support some or all of the int-serv services.

If the mappings of Proposal A above are applied in a switch implementing strict priority queueing between the 8 traffic classes (7 = highest) then the result will be that all Guaranteed Service packets will be transmitted in preference to any other service. Controlled Load packets will be transmitted next, with everything else waiting until both of these queues are empty. If the admission control algorithms in use on the switch ensure that the sum of the "promised" bandwidth of all of the GS and CL sessions are never allowed to exceed the available link bandwidth then things are looking good.

11.2 Full-duplex switched networks

We have up to now ignored the MAC access protocol. On a full-duplex switched LAN (of either Ethernet or Token-Ring types - the MAC algorithm is, by definition, unimportant) this can be factored in to the characterisation parameters advertised by the device since the access latency is well controlled (jitter = one largest packet time). Some example characteristics (approximate):

Type	Speed	Max Pkt Length	Max Access Latency
Ethernet	10Mbps	1.2ms	1.2ms
	100Mbps	120us	120us
	1Gbps	12us	12us
Token-Ring	4Mbps	9ms	9ms
	16Mbps	9ms	9ms
FDDI	100Mbps	360us	8.4ms

These delays should be also be considered in the context of speed- of-light delays of e.g. ~400ns for typical 100m UTP links and ~7us for typical 2km multimode fibre links.

Therefore we see Full-Duplex switched network topologies as offering good QoS capabilities for both Controlled Load and Guaranteed Service.

11.3 Shared-media Ethernet networks

We have not mentioned the difficulty of dealing with allocation on a single shared CSMA/CD segment: as soon as any CSMA/CD algorithm is introduced then the ability to provide any form of Guaranteed Service is seriously compromised in the absence of any tight coupling between the multiple senders on the link. There are a number of reasons for not offering a better solution for this issue.

Firstly, we do not believe this is a truly solvable problem: it would seem to require a new MAC protocol. Those who are interested in solving this problem per se should probably be following the BLAM developments in 802.3 but we would be suspicious of the interoperability characteristics of a series of new software MACs running above the traditional 802.3 MAC.

Secondly, we are not convinced that it is really an interesting problem. While not everyone in the world is buying desktop switches today and there will be end stations living on repeated segments for some time to come, the number of switches is going up and the number of stations on repeated segments is going down. This trend is proceeding to the point that we may be happy with a solution which assumes that any network conversation requiring resource reservations will take place through at least one switch (be it layer-2 or layer-3). Put another way, the easiest QoS upgrade to a layer-2 network is to install segment

switching: only when has been done is it worthwhile to investigate more complex solutions involving admission control.

Thirdly, in the core of the network (as opposed to at the edges), there does not seem to be enough economic benefit for repeated segment solutions as opposed to switched solutions. While repeated solutions

may be 50% cheaper, their cost impact on the entire network is amortised across all of the edge ports. There may be special circumstances in the future (e.g. Gigabit buffered repeaters) but these have differing characteristics to existing CSMA/CD repeaters anyway.

Type	Speed	Max Pkt Length	Max Access Latency
Ethernet	10Mbps	1.2ms	unbounded
	100Mbps	120us	unbounded
	1Gbps	12us	unbounded

11.4 Half-duplex switched Ethernet networks

Many of the same arguments for sub-optimal support of Guaranteed Service apply to half-duplex switched Ethernet as to shared media: in essence, this topology is a medium that *is* shared between at least two senders contending for each packet transmission opportunity. Unless these are tightly coupled and cooperative then there is always the chance that the junk traffic of one will interfere with the other's important traffic. Such coupling would seem to need some form of modifications to the MAC protocol (see above).

Notwithstanding this, these topologies do seem to offer the chance to provide Controlled Load service: with the knowledge that there are only a small limited number (e.g. two) of potential senders that are both using prioritisation for their CL traffic (with admission control for those CL flows based on the knowledge of the number of potential senders) over best effort, the media access characteristics, whilst not deterministic in the true mathematical sense, are somewhat predictable. This is probably a close enough approximation to CL to be useful.

Type	Speed	Max Pkt Length	Max Access Latency

Ethernet	10Mbps	1.2ms	unbounded
	100Mbps	120us	unbounded
	1Gbps	12us	unbounded

11.5 Half-duplex and shared Token Ring networks

In a shared Token Ring network, the network access time for high priority traffic at any station is bounded and is given by (N+1)*THTmax, where N is the number of stations sending high priority traffic and THTmax is the maximum token holding time [14]. This assumes that network adapters have priority queues so that reservation of the token is done for traffic with the highest priority currently queued in the adapter. It is easy to see that access times can be improved by reducing N or

THTmax. The recommended default for THTmax is 10 ms [6]. N is an integer from 2 to 256 for a shared ring and 2 for a switched half duplex topology. A similar analysis applies for FDDI. Using default values gives:

Type	Speed	Max Pkt Length	Max Access Latency
Token-Ring	4/16Mbps shared	9ms	2570ms
	4/16Mbps switched	9ms	30ms
FDDI	100Mbps	360us	8ms

Given that access time is bounded, it is possible to provide an upper bound for end-to-end delays as required by Guaranteed Service assuming that traffic of this class uses the highest priority allowable for user traffic. The actual number of stations that send traffic mapped into the same traffic class as GS may vary over time but, from an admission control standpoint, this value is needed a priori. The admission control entity must therefore use a fixed value for N, which may be the total number of stations on the ring or some lower value if it is desired to keep the offered delay guarantees smaller. If the value of N used is lower than the total number of stations on the ring, admission control must ensure that the number of stations sending high priority traffic never exceeds this number. This approach allows admission control to estimate worst case access delays assuming that all of the N stations are sending high priority data even though, in most cases, this will mean that delays are significantly overestimated.

Assuming that Controlled Load flows use a traffic class lower than that used by GS, no upper-bound on access latency can be provided for CL flows. However, CL flows will receive better service than best effort flows.

Note that, on many existing shared token rings, bridges will transmit frames using an Access Priority (see section 3.3) value 4 irrespective of the user_priority carried in the frame control field of the frame. Therefore, existing bridges would need to be reconfigured or modified before the above access time bounds can actually be used.

12. Signaling protocol

The mechanisms described in this document make use of a signaling protocol for devices to communicate their admission control requests across the network: the service definitions to be provided by such a protocol are described below. The candidate IETF protocol for this

purpose is called "Subnet Bandwidth Manager" and is described in [10].

In all these cases, appropriate delete/cleanup mechanisms will also have to be provided for when sessions are torn down. All interactions are assumed to provide read as well as write capabilities.

12.1 Client service definitions

The following interfaces are identified from Figures 2 and 3:

SBM <-> Address mapping

This is a simple lookup function which may cause ARP protocol interactions, may be just a lookup of an existing ARP cache entry or may be an algorithmic mapping. The layer-2 addresses are needed by SBM for inclusion in its signaling messages to/from switches which avoids the switches having to perform the mapping and, hence, have knowledge of layer-3 information for the complete subnet:

```
l2_addr = map_address( ip_addr )
```

SBM <-> Session/802 header

This is for notifying the transmit path of how to associate user_priority values with the traffic of each outgoing session: the transmit path will provide the user_priority value when it requests a MAC-layer transmit operation for each packet (user_priority is one of the parameters defined by the IEEE 802 service model):

```
bind_802_header( sessionid, user_priority )
```

SBM <-> Classifier/Scheduler

This is for notifying transmit classifier/scheduler of additional layer-2 information associated with scheduling the transmission of a session's packets (may be unused in some cases):

```
bind_l2sessioninfo( sessionid, l2_header, traffic_class )
```

SBM <-> Local Admission Control

For applying local admission control for a session e.g. is there enough transmit bandwidth still uncommitted for this potential new session? Are there sufficient receive buffers? This should commit the necessary resources if OK: it will be necessary to release these resources if a later stage of the session setup process fails.


```
status = admit_l2txsession( Tspec, flowspec )
status = admit_l2rxsession( Rspec, flowspec )
```

SBM <-> RSVP - this is outlined above in section 8.2 and fully described in [10].

12.2 Switch service definitions

The following interfaces are identified from Figure 4:

SBM <-> Classifier

This is for notifying receive classifier of how to match up incoming layer-2 information with the associated traffic class: it may in some cases consist of a set of read-only default mappings:

```
bind_l2classifierinfo( l2_header, traffic_class )
```

SBM <-> Queue and Packet Scheduler

This is for notifying transmit scheduler of additional layer-2 information associated with a given traffic class (it may be unused in some cases):

```
bind_l2schedulerinfo( l2_header, traffic_class )
```

SBM <-> Local Admission Control

As for host above.

SBM <-> Traffic Class Map and Police

Optional configuration of any layer-2 policing function and/or user_priority remapping that might be implemented on input to a switch:

```
bind_l2classmapping( in_user_priority, remap_user_priority )
bind_l2policing( l2_header, traffic_characteristics )
```

SBM <-> Filtering Database

SBM propagation rules need access to the layer-2 forwarding database to determine where to forward SBM messages (analogous to RSRR interface in L3 RSVP):

```
output_portlist = lookup_l2dest( l2_addr )
```


13. Compatibility and Interoperability with existing equipment

Layer-2-only "standard" 802.1p switches will have to work together with routers and layer-3 switches. Wide deployment of such 802.1p switches is

envisaged, in a number of roles in the network. "Desktop switches" will provide dedicated 10/100 Mbps links to end stations at costs comparable/compatible with NICs/adapter cards. Very high speed core switches may act as central campus switching points for layer 3 devices. Real network deployments provide a wide range of examples today. The question is "what functionality beyond that of the basic 802.1D bridge should such 802.1p switches provide?". In the abstract the answer is "whatever they can do to broaden the applicability of the switching solution while still being economically distinct from the layer 3 switches in their cost of acquisition, speed/bandwidth, cost of ownership and administration". Broadening the applicability means both addressing the needs of new traffic types and building larger switched networks (or making larger portions of existing networks switched). Thus one could imagine a network in which every device (along a network path) was layer-3 capable/intrusive into the full data stream; or one in which only the edge devices were pure layer-2; or one in which every alternate device lacked layer-3 functionality; or most do - excluding some key control points such as router firewalls, for example. Whatever the mix, the solution has to interoperate with these layer-3 QoS-aware devices.

Of course, where int-serv flows pass through equipment which is ignorant of priority queuing and which places all packets through the same queuing/overload-dropping path, it is obvious that some of the characteristics of the flow get more difficult to support. Suitable courses of action in the cases where sufficient bandwidth or buffering is not available are of the form:

(a) buy more (and bigger) routers
(b) buy more capable switches
(c) rearrange the network topology: 802.1Q VLANs [11] may help here.
(d) buy more bandwidth

It would also be possible to pass more information between switches about the capabilities of their neighbours and to route around non-QoS-capable switches: such methods are for further study.

14. Justification

An obvious comment is that this is all too complex, it's what RSVP is doing already, why do we think we can do better by reinventing the solution to this problem at layer-2?

The key is that we do not have to tackle the full problem space of RSVP: there are a number of simple scenarios that cover a considerable proportion of the real situations that occur: all we have to do here is cover 99% of the territory at significantly lower cost and leave the other applications to full RSVP running in strategically positioned high-function switches or routers. This will allow a significant reduction in overall network cost (equipment and ownership). This approach does mean that we have to discuss real life situations instead of abstract topologies that "could happen".

Sometimes, for example, simple bandwidth configuration in a few switches e.g. to avoid overloading particular trunk links, can be used to overcome bottlenecks due to the network topology: if there are issues with overloading end station "last hops", RSVP in the end stations would exert the correct controls simply by examining local resources without much tie-in to the layer-2 topology. In this case there has been no need to resort to any form of complex topology computation and much complexity has been avoided.

In the more general case, there remains work to be done. This will need to be done against the background constraint that the changing of queue service policies and the addition of extra functionality to support new service disciplines will proceed at the rate of hardware product development cycles and advance implementations of new algorithms may be pursued reluctantly or without the necessary 20/20 foresight.

However, compared to the alternative of no traffic classes at all, there is substantial benefit in even the simplest of approaches (e.g. 2-4 queues with straight priority), so there is significant reward for doing something: wide acceptance of that "something" probably means that even the simplest queue service disciplines will be provided for.

15. References

[1] ISO/IEC 10038, ANSI/IEEE Std 802.1D-1993 "MAC Bridges"

[2] "Supplement to MAC Bridges: Traffic Class Expediting and Dynamic Multicast Filtering", May 1997, IEEE P802.1p/D6

[3] "Integrated Services in the Internet Architecture: an Overview" RFC1633, June 1994

[4] "Resource Reservation Protocol (RSVP) - Version 1 Functional

Specification", Internet Draft, June 1997

<draft-ietf-rsvp-spec-16.[ps,txt]>

[5] "Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications" ANSI/IEEE Std 802.3-1985.

[6] "Token-Ring Access Method and Physical Layer Specifications" ANSI/IEEE Std 802.5-1995

[7] "A Framework for Providing Integrated Services Over Shared and Switched LAN Technologies", Internet Draft, May 1997 <draft-ietf-issll-is802-framework-02>

[8] "Specification of the Controlled-Load Network Element Service", Internet Draft, May 1997, <draft-ietf-intserv-ctrl-load-svc-05.txt>

[9] "Specification of Guaranteed Quality of Service", Internet Draft, February 1997, <draft-ietf-intserv-guaranteed-svc-07.txt>

[10] "SBM (Subnet Bandwidth Manager): A Proposal for Admission Control over Ethernet", Internet Draft, June 1997 <draft-yavatkar-sbm-ethernet-04>

[11] "Draft Standard for Virtual Bridged Local Area Networks", May 1997, IEEE P802.1Q/D6

[12] "General Characterization Parameters for Integrated Service Network Elements", Internet Draft, November 1996 <draft-ietf-intserv-charac-02.txt>

[13] "A Standard for the Transmission of IP Datagrams over IEEE 802 Networks", RFC 1042, February 1988

[14] "The Use of Priorities on Token-Ring Networks for Multimedia Traffic", C. Bisdikian, B. V. Patel, F. Schaffa and M. Willebeek-LeMair, IEEE Network, Nov/Dec 1995.

16. Security Considerations

There are no known security issues over and above those inherent in the Integrated Services architecture and the network technologies referenced by this document.

17. Acknowledgments

This document draws heavily on the work of the ISSLL WG of the IETF and the IEEE P802.1 Interworking Task Group. In particular, it includes previous work on Token-Ring by Anoop Ghanwani, Wayne Pace and Vijay Srinivasan.

18. Authors' addresses

Mick Seaman
3Com Corp.
5400 Bayfront Plaza
Santa Clara CA 95052-8145
USA
+1 (408) 764 5000
mick_seaman@3com.com

Andrew Smith
Extreme Networks
10460 Bandley Drive
Cupertino CA 95014
USA
+1 (408) 863 2821
andrew@extremenetworks.com

Eric Crawley
Gigapacket Networks
25 Porter Rd.
Littleton MA 01460
USA
+1 (508) 486 0665
esc@gigapacket.com