IMPROVING QUALITY OF SERVICE IN THE INTERNET

François Fluckiger
CERN, Geneva, Switzerland

Abstract
The Internet transport technology was designed to be robust, resilient to link or node outages, and with no single point of failure. The resulting connectionless system supports what is called a "best effort datagram delivery service", the performance of which is often greatly unpredictable. To improve the predictability of IP-based networks, several Quality of Service technologies have been designed over the past decade. The first one, RSVP, based on reservation of resources, is operational but has several major deficiencies, such as scalability difficulties. However, associated to other more recent technologies -RSVP aggregation, Diffserv and MPLS- the combination may result into an appropriate solution for improving Quality of Service guarantees in a scalable way.

1. BACK TO BASICS: THE INTERNET TRANSPORT SERVICE

One of the chief reasons for the success of the Internet lies in its *transport* technology. What is so special with the Internet transport technology? Unlike all the other wide area network technologies which predated it –such as the Switched Telephone Network, X.25 ATM or Frame Relay -, the Internet is not connection-oriented. This means that no prior connection between the two communicating systems –we say the "*end-systems*"- is required before a first fragment of information –the *packet*, also called the *datagram*- can be sent. As a result, all packets, which are treated independently by the network, must carry the full address of the destination system.

Packets may also take differing *routes* to reach the destination, though in practice, this only happens in case of link or node failure. If they take differing routes, they may be delivered in a miss-ordered way, as a given packet may take over another one on the different path. More importantly, packets may be lost, for reasons we examine later. If a packet is lost, the end-systems are not informed, and they have to detect such loses by themselves –as long as it matters for them to know about data losses. What the network guarantees is that it will do "*its best*" to deliver these datagrams. This is why the resulting Internet transport service is called a *best-effort datagram delivery service*.

Each of these base techniques, *connection-less* and *connection-oriented* have their own merits and drawbacks.

- In a *connection-less* network, there is no delay needed to set up a connection between the two communicating systems, before a first data packet can be sent. This is by the way, a fundamental feature of the Internet transport technology which was key in the support of the main Internet application: The World Wide Web. Indeed, imagine if, when you click on a link, a hard connection –that is a sort of telephone call- had be set over the network before the first packet of your request be sent to the destination (well, in practice, there is a first handshake called the TCP connection, but this is very fast as it only involves the two end-systems, but not the underlying network). This would be by far too slow. There would be no web!
- Another major advantage is that the routing is more dynamic as packets are independent,

and the network may easily adapt to changing conditions without impacting the service, as packet reroutes may remain unnoticed. Thus, the network appears as more resilient to failures.

Conversely, connection-oriented networks have also advantages.

- Connection-oriented networks, being aware of the requests before conversations actually start, by means of the *connection set up* procedure, may better predict the traffic load.
- If the traffic is more predictable, *resources* may be *reserved* more easily.
- If not enough resources are available, then the connection is refused. This is called the "busy signal", the functionality by which the network tells the user "*Sorry, I can't accept you call for the moment, please try later*". This is what we experience with the telephone network, the oldest connection-oriented service. This process is called "*Call Admission Control*" (*CAC*) or also "*Capacity Admission Control*" because what matters for deciding whether a new request for connection is granted is the available capacity of the network.
- It is therefore easier for connection-oriented networks to guarantee the quality of service of the communications they support.

2. WHY IMPROVING QUALITY OF SERVICE?

The objective of the efforts undertaken since the beginning of the 90s about the Internet quality of service is to improve the predictability of the service. Indeed, the historical "best effort datagram service" results in a somewhat unpredictable behaviour. There are multiple reasons why this has become no longer desirable.

- Users may wish to set up *Virtual Private Networks* (*VPN*) over the shared Internet with a guaranteed Quality of Service, such as the bandwidth of the pipes between sites part of the VPN. For example, imagine a company with one head-quarter and three branches, all four connected to the same *Internet Service Provider* (*ISP*) - to simplify the case. The VPN is to be made of three links between the branches and the head-quarter, each with a guaranteed bandwidth of say, 1 Megabit/second. Can we do that with the current Internet, that is with regular routers? No, we can't. We need something more than the "best effort datagram service"
- Users which connect to an ISP at a given access speed may wish to have a secure aggregate bandwidth out of this access link, irrespective of the destination of their traffic. For example, a company connecting to an ISP at 1.5 Megabit/second, and thus paying for that access speed, may wish to be guaranteed at least 2/3 of this access bandwidth for all its outgoing traffic, wherever it goes. Again, we can't do this with the "best effort datagram service".
- More and more multimedia applications use the Internet, in particular, audio and video streams. These streams usually need a minimum bit rate, below which it makes no sense to try and send the audio or video traffic. These are requirements that do not apply to aggregates of traffic as in the above case, but to point-to-point flows between two end-systems.

3. SERVICE DISCRIMINATION

Thus, the efforts for improving the *Quality of Service* (*QoS*) guarantees aim at moving away from the historical model of traffic where all packets are handled with the same priority by the network. By abandoning the pure egalitarian treatment of the datagrams, the new Quality of Service techniques create discrimination between packets. This is called *service discrimination*.

Service discrimination does not create any resource by itself –we do not get more bit rate on a link because some packets have higher priorities- therefore, it is not solving all problems of Quality of

Service. If a network, or a portion of a network (a link), has not enough capacity, service discrimination will not help for all the traffic. However it will help for some. Indeed, the objective of service discrimination is to give better service to some traffic. But this is done at the expense of giving a worse service to the rest. Hopefully, this only occurs in *times of congestion*.

In passing, this move from an egalitarian world, where all packets were equal, to a world where some packets are "more equal than others", a world where discrimination is legalized or organized, was viewed by some as orthogonal to the evolution of our society. I will leave this discussion to the judgement of the reader …

4. INTEGRATED SERVICES

The first substantial work on Quality of Service in the Internet started in the early 90s in the framework of what was called the *Integrated Services* (*IS*) model. The first release was made in 93.

The Integrated Services model is based on the statement that a single *class* of packets is no longer sufficient, and that new classes with higher priorities are needed, in the same way as we have the economy, business and often first class with airlines. How many new classes were needed? The Integrated Services model opted for two new classes of packets, resulting in a total of three possible classes in the new discriminated Internet world:

- The *best effort service class* (*BE*)

  This is the default class

- The *controlled-load service class* (*CS*)

  There, if the sender respects a certain traffic profile (that is a certain bit rate) for a given flow, then the network promises to behave as though it was *unloaded*, but without *quantitative* guarantees in particular of the latencies of the packets.

- The *guaranteed service class* (*GS*)

  There, packets are promised to be delivered within a *firmly bounded delay*. This is for special applications with very stringent time delivery requirements.

5. RESOURCE RESERVATIONS

The guiding principles of the Integrated Services model are the following:

- **Resource reservation is necessary**

  To improve the guarantees, the key resources needed in the network must be *reserved* in some way.

- **Reservations operate on flows**

  A *flow* is a stream of packets between one source and one destination (note that the destination may be a unique destination in the usual cases, but also a multiple destination, in case of a multicast flow; but this is comment for the specialists). For every flow that needs to benefit from either the *CS* or the *GS* service, reservations need be made.

- **Routers have to maintain flow-specific states**

  By *state*, we mean a block of memory in the router where information about the flow and its requirements are stored: the service class (*CS* or *GS*), the bit rate to guarantee for that flow, the conditions for delay if applicable, …

- **Dynamic Reservations need a signaling (set-up) protocol:**

This protocol has been specified and is called *Resource Reservation Protocol*, or *RSVP*

6. HOW TO EXPRESS QUALITY OF SERVICE

For a given flow, the *parameters* for determining the Quality of Service belong to two groups. The two groups reflect a *contract* between the user of the network on the one hand -or more exactly the user *sending* the traffic- and the network on the other hand.

- **Traffic parameters**

  The first group specifies what the flow *traffic pattern* is to be: parameters include the *sustained bit rate*, and the *burst size*, that is the tolerance that exists for the sender to exceed for short periods of time the sustained bit rate -as long as over longer periods, the average bit rate remains within the limit of the sustained bit rate.

  The sender promises in the contract to respect this traffic profile.

- **Quality parameters**

  These parameters specify the quality the network promises in turn to guarantee if the sender respects its agreed traffic profile. There are two things the network can possibly promise:

  - *latencies*, that is guarantees on the delay it takes for packets to traverse the network; this is called the *transit delay*
  - *loss ratio*: that the maximum proportion of packets not delivered against the total number of packets sent.

7. RSVP GUIDING PRINCIPLES

The *Resource Reservation Protocol, RSVP*, is the mechanism defined by the Integrated Services for reserving resources in the network. It is called a *signaling* protocol, because its aim is to *signal* to the network that a given flow is going to require certain guarantees for latencies and loss ratio, if the flow respects a certain bit rate. RSVP is based on a number of guiding principles.

- **RSVP is to co-exist with regular datagram service**

  Any router which supports RSVP, also supports the regular best effort datagram service

- **RSVP does not set hard connections**

  Instead, the connections are said to be "soft", and we explain this concept a bit later.

- **The amount of reserved resources is recipient driven**

  That is, the sender will only *propose* a certain traffic profile. But this is the receiver system which will *decide* how much of this proposed bit rate it can accommodate. Indeed, a frequent case may be when the sender is a fast powerful server system and the receiver a slow desktop device. The capabilities of the receiver to process the received data, for example to decompress a video stream, may be much lower than that of the sending server.

- **Reservations are unidirectional**

  If an application requires bi-directional reservations (such as in telephony, which is of course different from viewing a movie over the Internet, which is one-way only application), two reservations will have to be made, one in every direction.

8. RSVP PROTOCOL MECHANISMS

The RSVP protocol is only concerned with conveying information -along a path followed by a flow- to routers so that they can reserve the resources they need. For this, the protocol uses special RSVP *control packets*, which are packets which will be recognized by *RSVP-aware routers* and treated as such. The two main RSVP control packets are the *PATH* message and the *RESV* (reservation) message. The mechanisms for reserving resources for a given flow between a sender and a receiver are as follows:

- The “PATH” control messages are to be sent periodically by the sender

  This message carries the traffic parameters and the details of the requested service class. It has to be repeated periodically (thus, the sender keeps "saying" to the network "*I still need this quality of service for that particular flow*").
- The “PATH” control messages establishes an RSVP state in the intermediary routers
- The receiver replies with a “RESV” message, according to its capabilities
- The “RESV” message reserves resources, if available, in routers on the route back.
- I not enough resource are available at a given router, the router generates an error control message to the end-systems.
- If “PATH” is not repeated after time-out, then the resources are released
- “PATH” and “RESV” messages are carried by ordinary best-effort datagrams

9. WHICH RESOURCES ARE RESERVED

One aspect of RSVP which may be surprising at first sight is that the protocol which aims at reserving resource does not define what those resources are. And there is a good reason for this. The "resources”, that is "what is important" for a router or for a given transmission medium between two routers to guarantee a certain performance is fully implementation dependent. In particular, RSVP makes no assumption on the internals of routers.

That being said, in practice, with today’s routers and transmission links, reservations generally apply to two types of resources:

- a slice of the link bandwidth
- a fraction of the buffers from the buffer pool of the routers

Note that *reservation* is different from *allocation*. Reserving means that a given amount has been secured for a flow, but the exact and precise resource is not allocated (yet). This is similar to reservation in public transportation networks: reserving a flight ticket is different from getting the seat allocated.

10. PATH STABILITY

A difficult problem with RSVP lies with the fact that resources are reserved for a given flow over a given path. This path is the one followed by the initial PATH and RESV messages. However, it may turn out that the route followed by the packets change after the reservation has been made. This happens in particular if the route used for the first reservation was a long path, because a shorter path between the two end-systems was unavailable at that time (due to a router or link failure). When the shortest path is back to availability, the data will naturally take this shortest route because this is how the regular Internet routing works: packets always try and take the shortest route.

We then have the reservations made on the longer initial route and the data flowing on a new shortest route where no reservation has been made. Fortunately, the situation may clear itself after a while, because PATH messages are repeated (roughly every minute). The next repeated message will then take the shortest route and reserve resources on this new route.

This is unfortunately not a complete solution, because it may turn out that there are not enough resources available on the shortest route. Then, the two users experience a situation where, without knowing why, they move from a good quality transmission to a bad one, simply because rerouting took place. What we need for a more complete solution is a means of having more stability for reserved paths, but also for finding routes, possibly longer, but which do have the required resources when the shortest route can not satisfy a given request. Such a routing technique for not only finding a short route but also one with enough resources to satisfy a certain quality of service is called *Quality of Service-based routing*.

11. RSVP OVERHEAD FOR DATA PACKETS

Another difficulty with RSVP lies in the *overhead* created in every router by the need to analyze every packet in order to know which priority it has, or more exactly which *service class* it belongs to (Best effort, Controlled-load, Guaranteed).

The process of deciding which service class an incoming packet belongs to is called *classification*. In the case of RSVP, how can we know the class? Is there any mark somewhere in the packet header, any field than can be rapidly and easily examined and that would tell us*: "It is class 1, 2 or 3*"? The answer is no: the class is not explicitly stated in the data packets when using RSVP only.

This is by examining the pair of source and destination addresses (and possibly other fields such a the protocol type) that characterizes every flow and by looking at a table to check whether a reservation has been made for that particular pair, that the router knows the class of the packet. This technique, which rely on the examination of *multiple fields* (at least two) is called *multi-field classification*. Multi-field classification, which is to be performed on every packet creates serious overhead to RSVP routers.

12. RSVP SCALABILITY

The overhead of packet classification is one of the drawback of RSVP and raises a scalability issue.

Another scalability difficulty lies in the fact that *states* -that is, a block of memory that stores static and dynamic information about the reservation- has to be created and maintained for every pair of sender and receiver that has to enjoy an improved service. If this may be appropriate for small scale Intranets with a limited number of concurrent RSVP reservations, the need to maintain states per individual flow is unsuitable for large scale Internets. This was the argument from opponents to RSVP, and this triggered the development of another, complementary technology called *DiffServ* that we see later.

13. SUMMARY OF RSVP DEFICIENCIES, AND WAYS FOR OVERCOMING THEM

We have seen that RSVP, though being an effective mechanism for reserving resources has three shortcomings:

1. **The per-flow states to maintain in routers**

   This entails a scalability problem for large scale Internets

2. **The multi-field classification on data packets**

   The overhead generated by the need to analyze every packet by examining multiple fields in order to determine the service class of that packet

3. **The instability of routes and the lack of quality of service-based routing**

   The fact that resources are reserved along routes and that we cannot guarantee the data traffic will follow these routes.

We see in the next sections how three rather recent technologies can be associated to RSVP so as to overcome most of those deficiencies. These technologies are:

- *RSVP aggregation*, to overcome problem 1
- *Diffserv*, to overcome problem 2
- *MPLS*, to overcome problem 3

14. RSVP AGGREGATION

RSVP Aggregation is a recent technique not yet stabilized at the time of this writing but very promising. The aim is to drastically reduce the number of states to be maintained in the network. This is a technology to be used within the core of the large scale Internet networks, not a technology for Intranets or limited scale Internets. The idea is to replace the reservations made per individual flow in the plain RSVP, by reservations made for aggregates of flows. Which aggregates?

Simply consider the core of a network and imagine that this core uses RSVP aggregation, that is, all the routers in the core understand the RSVP aggregation protocol. This core is made of a number of routers, some of them being *edge routers* -that is, having connection to the "non-RSVP aggregation" periphery- and others being *interior routers* only. The idea is that for any pair of edge routers, a *single* reservation will be made for all the flows that enter the core through one router of the pair and that exit the core through the other router of the pair. Thus, all those flows with the same *ingress* and *egress* router will share the same reserved path, and the amount of resources reserved over that path will match the sum of the individual requirements (not necessarily exactly).

As a result, if there are N edge routers surrounding the RSVP aggregation core, we will have (N^2–N) paths, therefore states, to maintain (remember, the reservation are unidirectional, thus we need two paths per pair in practice). This may be a big number but anyway lower than the total number of individual flows aggregated within the reserved paths.

15. DIFFSERV

Diffserv, which stands for *Differentiated Services*, is another recent technique aiming at overcoming the problem of heavy classification -that is the process for routers of knowing which service class a packet belongs to. The idea it to "mark" the packets with and indication of their priority in order to avoid having routers examining multiple fields. This mark is called a "*differentiated mark*", or *a Diffserv Code Point* (*DSCP*) and serves to map to a *differentiated treatment* to be applied to the packet. For a fast classification, the "mark" must be:

- of fixed length
- located at the beginning of the packet
- in a fixed position
- to be used as a direct pointer to find out what the differentiated treatment is to be.

The use of the mark is similar to that of RSVP aggregation: this is a technique which assumes that there is a core in the network which is "*Diffserv-capable*", that is, made of routers which understand the Diffserv marks and know how to exploit them for efficiently determining the packet priority. At the *edge* of this Diffserv core, the edge routers must be provisioned with the appropriate instructions to mark the packets (e.g. based on identification of flows such as source and destination addresses)

Therefore, by combining in the core of the network, RSVP aggregation and Diffserv marking, we can overcome two of the deficiencies of the plain RSVP:

- RSVP aggregation reduces drastically the number of reservations to maintain
- Diffserv removes the overhead of CPU-consuming classification by providing a simple, fast way of knowing the packet priority.

The final technology we briefly see, *MPLS*, will help us with the third difficulty, the route stability and the need for finding routes which also satisfy some quality of service constraints.

16. MPLS

MPLS stands for *Multi-Protocol Label Switching*. This is not a Quality of Service technique per se. This is a *forwarding* technology, a new approach to build network nodes which are neither pure IP routers nor pure switches, rather *hybrid* objects which try and combine the good points of both systems. The objective of MPLS is to build network nodes which can forward packets fast, hopefully at "hardware speed". More exactly, the technology provides a mechanism for making a fast *forwarding decision*, that is deciding to which outgoing link an incoming packet should be sent. To this end, MPLS uses a *logical reference*, called a *label*, which is inserted somewhere in the header of the packet. MPLS routers are provided with forwarding tables that map the incoming label to an outgoing link. The mechanism is analogous to the switching done in ATM or in X.25 switches.

MPLS, as RSVP aggregation and Diffserv, uses the *core* and *edge* principle. The MPLS technology is used within an "*MPLS-capable*" core, a mesh of MPLS routers. When entering the core, all packets taking the same route receive the same label, which is then used within the core by routers to take the forwarding decision, instead of the destination address as done outside the MPLS core.

This technique is not per se a solution to the remaining difficulty of the plain RSVP: the route stability and the Quality of Service-based Routing. However, it turns out that the paths followed by all the packet which carry the same label (called *LSPs*, *Label Switched Paths*) may be *constrained* to have a certain *stability* (for example, only change in case of failure), or to match certain *performance* characteristics such as securing a certain bit rate or transit delay. The later facility, *called constraint-based routing* is nothing else than a form of Quality of Service-based Routing.

17. CONCLUSION

The combination of three techniques to be used in the core of large scale Internet:

- RSVP aggregation for reserving resources, yet limiting the number of states to maintain,
- Diffserv for a fast and easy classification of the data packets into service classes,
- and MPLS for providing appropriate route stability and Quality of Service-based Routing

is likely to provide in the future a solution which supports Quality of Service and is reasonably scalable.

18. BIBLIOGRAPHY

Understanding Network Multimedia, François Fluckiger, Prentice Hall,
ISBN: 0-13-190992-4

Quality of Service in IP Networks, Grenville Armitage, MacMillan
Technical Publishing, ISBN 1-57870-189-9

MPLS, Technology and Applications, Brice Davie and Yakov Rekhter, Morgan
Kaufann

Interconnection Second Edition, Radia Perlman, Addison-Weisley, ISBN:
0-201-63448-1