Resource Management in Differentiated Services Networks

Andrew Simmonds and Priyadarsi Nanda
Department of Computer Systems, Faculty of Information Technology, University of Technology, Sydney, Australia. e-mail: <simmonds, pnanda>@it.uts.edu.au

Abstract: The Differentiated services architecture (diffserv) proposed by the Internet Engineering Task Force (IETF) [1] provides service differentiation in the Internet in an efficient and scalable manner. The central idea of diffserv is that the Type Of Service field (TOS) in the IPv4 header can be used to prioritize traffic in an aggregated manner. In this paper we work on the resource management implementation issues required to support a wide variety of Quality of Service (QoS) traffic streams having different parameters. A well known problem with diffserv [2, 3] is that, being based on aggregate streams, it currently does not support end-to-end QoS. We believe our approach to diffserv can help to achieve dynamically allocated end-to-end QoS using a Bandwidth Broker (BB) architecture. We consider our resource management scheme to be simple and well suited to implementation in a diffserv internet of multiple domains. Bandwidth Brokers (BB) in each domain are the point of control for various activities performed within and between the domains.

Key words: diffserv, QoS, Bandwidth Broker, BB, resource reservation, allocation

1. Introduction

In recent years there has been considerable research focused on extending the Internet architecture to allow different QoS to different traffic classes. After extensive study, the Internet Engineering Task Force (IETF) has proposed two different models in order to guarantee proper QoS. Integrated Services (intserv) can provide end-to-end QoS using the Resource Reservation Protocol (RSVP) [4] to individual flows, but lacks scalability because of the problem of maintaining individual flow states in the core routers in the Internet and because its signalling complexity grows with the number of flows [5]. Differentiated Services (diffserv) on the other hand relies on packet marking and policing at the access or edge routers and by

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35673-0_28
C. McDonald (ed.), *Converged Networking*

considering aggregated flows in the core routers reduces complexity and improves scalability, but of course, since the flows are aggregated there is no end-to-end QoS. In this paper we propose a hierarchical resource management scheme, compatible with existing Internet protocols, to allow end-to-end QoS based on the diffserv architecture. The focus of this paper is to present a concept of how to extend diffserv to enable it to dynamically support end-to-end QoS by providing an extra management plane to an existing diffserv structure.

The rest of this paper starts in section 2 by considering how we might prioritize traffic into different classes and the implications of the choices made; then in section 3 we give an overview of how a Bandwidth Broker (BB) might manage resources for the different classes in a single BB domain; finally, in section 4 we give an overview of how the interaction between BB domains might be conducted.

2. Traffic classification

As an example of traffic classification, we propose to provide 5 classes of service running over the same network infrastructure. For simplicity we will differentiate these classes in terms of their loading, which means the highest priority class is the least loaded class, i.e. even under heavy traffic conditions it should provide the same QoS as the traditional Best Effort (BE) class over a lightly loaded network. In effect we are prioritizing the traffic in terms of relative bandwidth (load = user data rate/channel capacity) and proposing capacity differentiation in a relative differentiated services model [6]. Ultimately, the aim is that the scheme can be extended to deal with other QoS parameters such as packet loss, delay, cost and jitter. Our 5 classes in decreasing order of priority are:

1.Expedited Forward (EF)
2.Assured Forward gold (AFg)
3.Assured Forward silver (AFs)
4.Assured Forward bronze (AFb)
5.Best Effort (BE)

The different QoS streams are achieved by differential loading, e.g. more AFg traffic is allowed per unit of resource than EF traffic (i.e. the relative bandwidth is higher) etc.

It is also an aim that the scheme can reserve particular network resources for particular priority levels, e.g. EF and AFg might be reserved for low latency but expensive links, i.e. we aim initially to use policy based routing. Of course, no QoS protocol is necessary if a different physical network is used for each traffic class, but we wish to be able to mix and match different physical and logical network resources to provide the required QoS.

In fact, one way of considering the problem of providing QoS is that it requires supporting several logical networks over the same physical network, which highlights the fundamental problem with providing QoS in the Internet: that the underlying network only provides a connectionless service with connection oriented services having to be provided at the end hosts, either by the Transport or Application layers.

Our approach aims to be consistent with the ethos of the Internet as we understand it, i.e. a dynamic, fault tolerant, self healing system which is based on a connectionless service. In our view, this precludes any guarantee that high priority traffic can always be strictly managed. This leads to the paradoxical conclusion that it is how the lower priority traffic is treated which is more important than what is done with the higher priority traffic. Hence, a key requirement of our proposal is that at least the BE stream is allowed to continue operating, though of course with reduced resources if the network is loaded with higher priority streams. In fact we propose that each class of traffic has a minimum resource allocation. As an example, consider a 100 Mbps link with 20 Mbps reserved for BE and 10 Mbps for the other classes, i.e. for the streams discussed above under maximum demand for all classes of traffic the stream allocation will be as in table 1. The minimum resource allocation for each class of service is important so that connections across the Internet can be maintained, albeit at a lower grade of service. Then, when resources become available again, traffic flow can increase as determined by the standard TCP congestion control algorithm.

Table 1. fully loaded resource reservation

EF 50 Mbps
AFg 10 Mbps
AFs 10 Mbps
AFb 10 Mbps
BE 20 Mbps

On the other hand, if there is no demand for any traffic but BE then the entire 100 Mbps is allocated to BE. Any higher priority traffic will be able to 'bump' some BE traffic to gain access to the resource. Assume that it is a 10 Mbps stream of AFg traffic that needs to be accommodated. The resource reservation will then appear as in table 2, with extra BE traffic being able to use the resource allocated to the unused EF, AFs and AFb streams.

Table 2. partially loaded resource reservation (10 Mbps AFg, 90 Mbps BE)

EF 50 Mbps - not used
AFg 10 Mbps
AFs 10 Mbps - not used
AFb 10 Mbps - not used
BE 20 Mbps

Total BE traffic 90 Mbps

Only BE traffic is allowed to take advantage of unused resources. This traffic acts as a buffer which can be 'bumped' on demand, so that useful traffic is being carried when resources are free, but resources can be quickly made available to higher priority classes. Up to 10 Mbps each of AFs and AFb can be accommodated by bumping BE traffic, but if any more were allowed there would be a problem if there were then to be a demand for, say, 50 Mbps of EF traffic. In order to carry this potentially lucrative traffic, some non BE traffic would have to be dropped, or the minimum BE traffic allocation reduced. Neither of these are acceptable solutions: if we have accepted the AFs and AFb traffic, we are presumably under an obligation to deliver it; and the BE minimum allocation is especially important so that traffic which is not QoS aware can still find its way across the Internet. We stress that we are simply proposing a concept here, the actual implementation could well be more complicated, e.g. some resources could be allocated to either AFs OR AFb, etc., but the principle is that each stream has a minimum allocation and only BE traffic (traffic we have not guaranteed to deliver and which we can drop when required) can use temporary spare capacity.

A potential issue [6] is that under some conditions lower priority traffic could enjoy a higher grade of service than all but the EF traffic. For example, consider the case of a fully loaded network (as in table 1), and then virtually all of the BE traffic dies away. By definition, the EF traffic is configured to enjoy the same grade of service as traffic on a lightly loaded network, so this will have the same grade of service as the BE traffic, but the AF streams will have a worse grade of service. Statistically such large fluctuations are less likely to occur as traffic is concentrated towards the core; so a reasonable course of action is to do nothing, on the basis that networks should not be optimized to cope with pathological conditions.

Another reasonable option is to negotiate to accept other traffic as BE traffic (charged as BE traffic), on the basis that currently the BE stream will support a higher grade of service, but we cannot guarantee that this will continue. We differ in this from [6] who when discussing capacity differentiation in a relative differentiated services model state that an important feature of such a model is "predictability, in the sense that the class differentiation should be consistent (i.e. higher classes are better, or at least no worse) even in short timescales, independent of the variations of the class loads". Fundamentally, because of the connectionless nature of the Internet, there are no absolute guarantees of a particular grade of service, only statistical 'guarantees' (an oxymoron). What we are providing with our different classes is our promise that we will do our best to ensure that over time these traffic streams will be able to support appropriate applications and that the applications can continue to completion. Although the BE class may at some times be lightly loaded and able to deliver a high grade of service, we do not promise that the BE stream will be able to continue at that level of performance, or even that the application can continue to completion at a degraded grade of service.

In the above scenario we have considered that the network is homogenous and all resources can be utilized by BE traffic if not otherwise used. But, because there is a clear and consistent mapping of resources to different traffic classes, this scheme could easily be adapted to account for other cases, e.g. where EF and AFg traffic is carried over special links which must not be used by other traffic.

3. Dynamic resource allocation and traffic policing

It would seem natural that in order to dynamically allocate or reserve resources, they first need to be discovered. This is the task of a QoS-based routing protocol such as QOSPF (QoS routing extensions to OSPF) [7], which is triggered by a resource reservation request. Such protocols could indeed be used as an extension to our proposal, but we do not consider resource discovery per reservation request to be essential. Indeed this will place a burden on the core routers, something we aim to avoid. Initially, we propose that network resources are statically entered into the BB, and the BB maintains an overview of allocated and free network resources. However, the network topology can change dynamically, e.g. because of link failure, and for the future such topology changes should map into the resource allocation. The system needs to be dynamically adjusted to take account of this, and also of slowly changing (~ hours) user demand (e.g. more demand for AFg, less for AFb). What is needed is a per class resource discovery protocol, not a per flow resource discovery protocol, see e.g. [8]. Hence initially we propose to use policy based routing, but then migrate to using

constraint based routing where the system takes account both of policy and the current state of the network.

We now consider the tasks required to dynamically set up and tear down traffic flows in a diffserv environment, and where these tasks can best be located. In the first instance, the H_{SA} (host source address) requests a particular QoS for a stream to H_{DA} (host destination address) from its BB. If necessary a negotiation phase occurs in which H_{SA} and H_{DA} decide on the appropriate level of service using the Best Effort (BE) stream, and then one of them effectively becomes H_{SA} and applies to its BB for the desired resources. The default is a symmetric channel, but an asymmetric channel request can also be made.

Each domain is considered to have one or more Ingress routers, Egress routers and Core routers, see figure 1 for a single domain. Note the important point in figure 1 that core routers are not involved in QoS signalling (- - -). A suitable signalling protocol would be COPS [9, 10], with an extension COPS-SLS [11] specifically proposed for dynamic Service Level Specification negotiation or alternatively the Dynamic Service Negotiation Protocol (DSNP) [12].

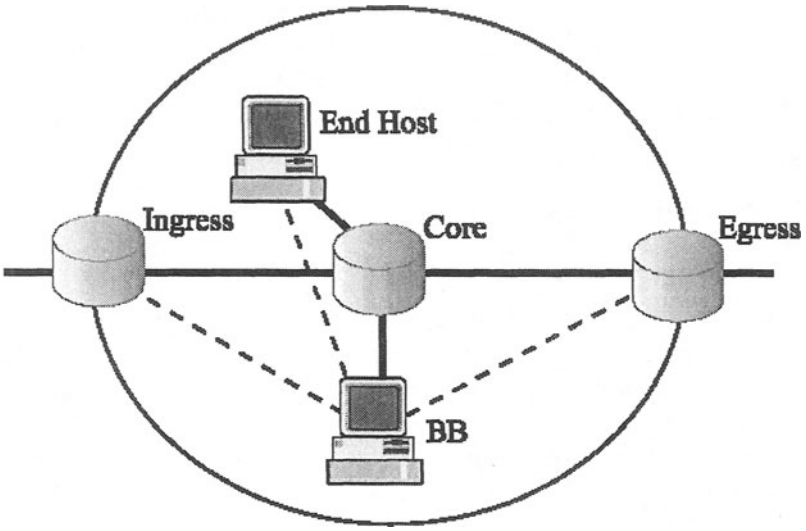

Figure 1. A single Bandwidth Broker domain

For an isolated domain a BB is seemingly pointless, so figure 2 shows how multiple domains might be connected. Normally the path is duplex, so each Ingress router would also act as an Egress router too. The process by which resources are requested and reserved by BBs between domains is

discussed in section 4. For now we assume the request is authorized and that the Ingress and Egress routers are updated by the BB in their domain with the new aggregated rates to take an account of the end-to-end channel for the successful connection.

Where a single BB domain might be used is where an IP network is used as an access network to some other WAN which supports QoS (e.g. ATM). The BB would then interface with the QoS management system in the other network.

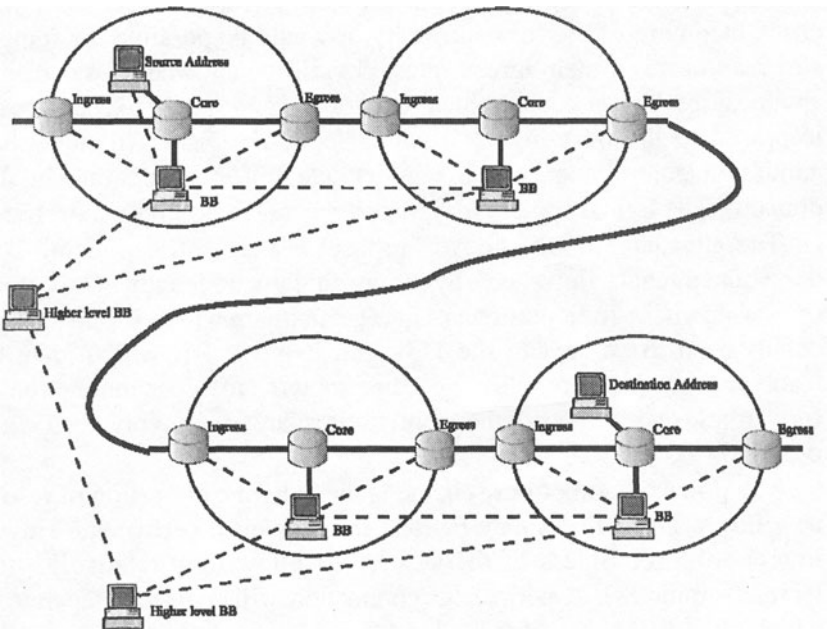

Figure 2. Multiple Bandwidth Broker domains

The Ingress routers must perform admission control on the incoming traffic to check it conforms to the aggregate profile. If the aggregate flow is within the limits specified by the BB there is no necessity to check individual flows and it is proposed that to save on overhead this is not done. However, if an aggregate stream is out of limits then summary details of individual flows in that stream need to be sent to the BB for policing. The policing function can be separated into two sub tasks: traffic authentication (is this traffic allowed to use the resource) and admission control (is the resource sufficient to support the traffic). The BB finds the problem and instructs the Ingress router to take appropriate action on a particular stream (e.g. dropping traffic or putting it in a different class). Summary details may also be sent back to the BB periodically even if the traffic levels are within

limits, so that the BB can monitor and update traffic stream states, and detect and prepare to remove unauthorized traffic before it causes congestion. However, we propose a relaxed policing policy, so that traffic is not removed unless it is causing a problem or is near to causing a problem. Monitoring traffic states also allows the BB to 'look ahead' and reserve resources if it predicts a peak in demand is coming. If the BB cannot be contacted for some reason, the Ingress routers default to some pre-arranged policy to drop certain traffic under congestion conditions.

At first site it would appear that the Egress routers do not need to perform admission control. However, because the Ingress routers do not always check incoming traffic for conformity, it would be possible for some ingress streams to exceed their agreed rates, but still be allowed, and converge on a single Egress router causing local congestion and traffic loss before the data is passed to the next Ingress router. Hence the Egress routers should do admission control too (which will include traffic originating in the local domain). The Egress routers may also shape traffic to improve traffic flow.

The advantage of the above proposal is that traffic policing is mainly done on aggregate flows, e.g. by monitoring queue lengths. Only for call set up/tear-down, or for a particular aggregate stream which is outside limits, do details need to be sent to the BB. And it is the BB which identifies and resolves the problem, leaving the routers to concentrate on packet forwarding tasks. Indeed the core routers are not involved at all in our proposal.

The principal aim of our scheme is the dynamic allocation of resources to new flows. These flows may be long term, or even permanent. However, to improve the robustness of the system we allow only relatively short term leases (~ minutes). A permanent connection will require a separate process which periodically asks for the lease to be renewed. Hence if the BB tables become corrupted, invalid entries will expire over time and new valid entries replace them. Meanwhile, if the system does not experience congestion, no action will occur. Only if a router experiences congestion in one of its aggregate streams will details be sent back to the BB. Left to itself, the out of synch BB will probably decide most of the traffic is unauthorized and take action. However, since the principle of operation is to allow traffic if it is not causing a problem, only some traffic streams will be lost. It will be unfortunate if this is authorized traffic, but such is life. As the tables are rebuilt and lost connections re-established, authorized traffic will flush out unauthorized traffic.

4. Inter-domain resource management

BBs in adjacent domains signal using either the BE channel, or a negotiated higher priority channel, to allocate resources in aggregate streams

between their Egress and the adjacent domain's Ingress routers, see figure 2. It is assumed that within a domain there is no need to reserve resources, i.e. a BB domain is considered internally to be adequately resourced. These resources are either statically entered or discovered by a per-class resource discovery protocol. If this is not the case, the domain needs to be subdivided into sub-domains each with their own Ingress, Core, and Egress routers, but a single BB could serve a whole Autonomous System or multiple BBs could be used for the sub-domains. This makes the mapping of network resources simpler, because it would normally be coarse grained. An alternative approach is described in [13] which requires the core routers to be involved in protecting traffic reservations, but we feel that for best scalability the core routers should not be burdened more than absolutely necessary.

Adjacent BBs along the path between the source address host (H_{SA}) and the destination address host (H_{DA}) continue the process, negotiating between themselves as to whether they can allow the connection. In order to implement this, the Ingress routers would accept messages for their BB and forward them on, see figure 3 (per-domain signalling). It is recognized that such a domain-by-domain negotiation scheme does not scale well over multiple domains and it is proposed that a hierarchical scheme be implemented, analogous to the DNS hierarchy, with each BB being required to register with its superior (done in the initial configuration). Also, end-to-end QoS can only be guaranteed if all domains between source and destination allow the connection and all support the BB architecture, hence the need for some intelligent route selection done by some central node, rather than by domain-by-domain negotiation. As a backup however, we propose the domain-by-domain method to account for the case when a BB has lost its superior.

The delay in setting up a call over multiple domains, using domain-by-domain negotiation is likely to be excessive. However, it should be stressed that this is fall back scenario to allow some dynamic QoS traffic over diffserv routes which would otherwise not be able to accept any QoS streams, even though there were free resources. The normal method for a new destination would be to use the BB hierarchy, see figure 3 (hierarchical signalling). But for existing major traffic routes we envisage that certain QoS channels would be reserved to enable calls to be quickly accepted. As traffic using these routes varies over the long term, a class-based resource discovery protocol would enable the BBs in the domains along the route to intelligently reserve or release resources.

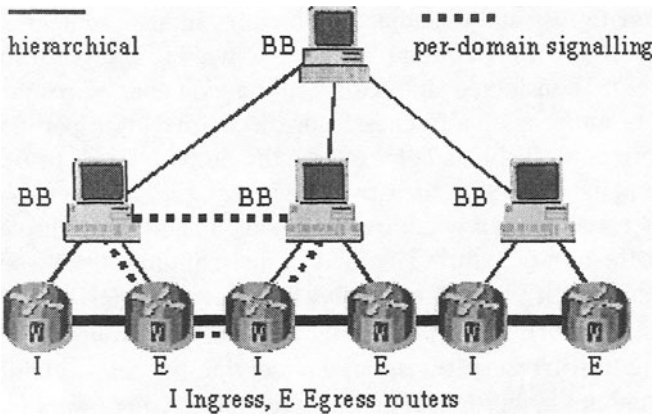

Figure 3. BB signalling hierarchy

5. Conclusion

We have presented a scheme which is robust and will enable end-to-end dynamically configured QoS to be delivered over an existing diffserv internet. We believe our solution will scale well in the Internet because:

1.We are dealing with aggregate streams, which means core routers do not have to do traffic control tasks and we strived not to load the core routers with any further tasks.

2.Because our BB architecture is hierarchical;

3.Because the Egress and Ingress routers are principally concerned with aggregate flows, only sending summary details to their BB.

We have explained the concept on the basis of a five tier priority scheme based on a base load of BE traffic. Because we preserve resources for traditional BE traffic our scheme is fully compatible with existing Internet protocols, and we have taken care to ensure the ethos of operation of our proposal is compatible with that of the Internet.

For existing major traffic routes over the Internet a new request is quickly accepted or declined, otherwise, if the destination is new the reservation is subject to negotiation by a hierarchical BB system. If all else fails, the reservation is subject to negotiation on a per-domain BB system.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for differentiated services", RFC 2475, December 1998 and J Heinanen et al, "Assured Forwarding PHB Group", RFC 2597, June 1999

[2] Manuel Gunter, Torsten Braun, "Evaluation of Bandwidth Broker Signaling", 7th International conference on Network Protocols (ICNP), October, 1999

[3] Chen-Nee Chuah, Lakshminarayan Subramaniam, Randy H. Katz and Anthony D. Joseph, "Resource Provisioning Using A Clearing House Architecture", ACM SIGCOMM 2000-Poster Session, August 2000.

[4] RFC 2205 "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification" and RFC 2210 "The Use of RSVP with IETF Integrated Services"

[5] Raju Rajan, Dinesh Verma, Sanjay Kamat, Eyal Felstaine, Shai Herzog, "A policy Framework for Integrated and Differentiated Services in the Internet", IEEE Network, September/October 1999

[6] C Dovrolis, P Ramanathan, "A Case for Relative Differentiated Services and the Proportional Differentiation Model", IEEE Network, Sept/Oct 1999, pp. 26 - 34

[7] G. Apostolopoulos, et al, "QoS Routing Mechanisms and OSPF Extensions", RFC 2676, August 1999

[8] Doan B. Hoang, Qing Yu, Ming Li, David Dagan Feng, "Fair Intelligent Congestion Control Resource Discovery Protocol on TCPBased Network", IFIP Interworking 2002, 6th International Symposium, 13 – 16 Oct 2002, Perth, WA, Australia

[9] D Durham, et al, "The COPS (Common Open Policy Service) Protocol", RFC 2748, Jan 2000

[10] S Giordano, M Mancino, A Martucci, S Niccolini, "VOIP Dynamic Resource Allocation in IP DiffServ Domain: H.323 vs. COPS interworking"

[11] Thi Mai Trang Nguyen, Nadia Boukhatem, Yacine Doudane, Guy Pujolle, "COPS-SLS: A Service Level Negotiation Protocol for the Internet", IEEE Communications Magazine, May 2002, pp 158 – 165

[12] J –C Chen, A McAuley, V Sarangan, S Baba, Y Ohba, "Dynamic service negotiation protocol (DSNP) and wireless Diffserv", Proc IEEE Int Conf on Communications (ICC'02), New York, April 2002

[13] Ibrahim Khalil, Torsten Braun, "Implementation of a Bandwidth Broker for Dynamic End-to-End Resource Reservation in Outsourced Virtual Private Networks", Proc 25th Annual IEEE Conf on Local Computer Networks (LCN'00), 8 - 10 Nov, 2000, Tampa, Florida.