A Dynamic Admission Control Scheme in a DiffServ Domain

Jin-Cheol Kim, Young-Hee Lee
Information and Communications Univ. Daejeon, Korea
jckim@icu.ac.kr yhlee@icu.ac.kr

Abstract

Gaining lessons from the IntServ/RSVP experiences, more simplified Differentiated Services(DiffServ) has emerged as the core network architecture with enhanced scalability for implementation and deployment. In order to provision differentiated services on a DiffServ domain, it is necessary to provision resources appropriately in the control or management plane of the DiffServ domain, in addition to the data plane service protocols implementation. But, efforts in the control or management plane are not matured in the DiffServ domain. While the static provisioning may be sufficient for the quantitative traffic, the qualitative provisioning is more difficult to support, so that the provisioning parameters should be estimated based on the accurate traffic characteristics, which can be obtained from possibly real time measurements.

The paper aims to resolve dynamic resource provisioning, especially for the qualitative traffic by measurement-based admission control scheme having some implicit signaling component using control plane DSCPs. The proposed scheme has been simulated on the ns-2 simulator, using traffic with burst characteristics. The results shows that the proposed scheme has the capability to support a fine-grained, dynamic admission control with flexibility in DiffServ domain.

1. Introduction

The IP network is recently desired to be prepared with appropriate resources and QoS to support the real-time services arising from the merging datacom and telecom services industry. Such real-time services would be streaming media, Internet telephony or some video conferencing applications, etc. The way to provide bandwidth, delay or loss guarantees in the Internet would heavily depends on the actual network and traffic conditions[4].

The DiffServ architecture handles the IP level QoS in addition to the traditional best effort service, and it leaves more complexity to edge nodes and the interior nodes do not care per-flow processing. The DiffServ model defines three major traffic classes, and different PHBs (Per-Hop Behavior) are applied for each traffic class. The PHB can be defined as the required general behaviors at the network nodes with some level QoS awareness, combining the classification, dropping, queuing, scheduling, and forwarding actions. The Expedited Forwarding(EF) service class is defined to support the quantitative traffic which is characterized by a traffic profile that can be verified by parameter measurement. The Assured Forwarding(AF) service class is defined for the qualitative traffic with relatively higher priority than the best effort service. The AF class is further subdivided into several subclasses. These PHBs can be mapped to the DSCP(DiffServ Code Point) fields of the data packets, and network nodes process these packets in a differentiated manner according to the DSCP fields.

Resource provisioning refers to the determination and allocation of the resources needed at various points in the network. It can consist of the admission control, and resource allocations with configurations along the edge-to-edge traffic path. There are contradictions in the requirements of the service providers and the customers. The service providers require multiplexing gain by utilizing resources optimally, while the users want to use the network as much as possible [2]. Quantitative traffic can be serviced by resource reservation based on the SLA/SLS(Service Level Specification). The SLS is basically static, although the users can dynamically change it. So, for the quantitative traffic whose SLS is static, static provisioning may be sufficient since the quantitative traffic pattern can be well predicted.

However, qualitative traffic does not specify egress points and will not restrict submitted traffic to specific egress points, so the qualitative traffic cannot be assumed to follow specific routes with the same degree of predictability as quantitative traffic. Thus, the qualitative provisioning is difficult to support, and provisioning parameters should be estimated based on the accurate traffic characteristics as far as possible. Since the qualitative traffic offers less predictable patterns, causing traffic at different nodes to change dynamically even when the SLA is static, dynamic resource provisioning is necessary especially for the qualitative traffic for better use of network resources including statistical gains[1].

Dynamic resource provisioning may be signaling based, measurement based or both. Measurement based provisioning would adjust the resources committed in response to the traffic loads actually measured at the nodes. This paper aims to resolve the dynamic resource provisioning problem, and suggests a dynamic admission control scheme for the fine-grained and fairly lightweight dynamic resource provisioning, which is based on the real time measurement along the edge-to-edge paths in a DiffServ intra-domain. This paper focuses on the intra-domain resource provisioning, and inter-domain issues are left for the future study.

2. Related Work

The main approaches to the resource provisioning mechanisms for the DiffServ domain can be classified as either centralized or distributed. In the centralized approach, such as Bandwidth Broker, one central server manages the resource allocation for the entire domain. In the distributed approach, the edge nodes support the resource allocation and the interior nodes can help them.

The centralized mechanisms are easy to manage, but they should upgrade and maintain domain topologies and resource allocation status. Also, they should support a signaling protocol, and in the server area faulty situations, the domain management may be heavily troublesome. Another issue in the centralized mechanism is the potential scalability problem. A large-scale network domain would be difficult to manage dynamically and to predict traffic by the central server, so the improvement of resource utilization and dynamic admission control would be more difficult as the networks scale up. The centralized approach can be applied to the quantitative traffic services that are easy to predict, but would be limited for the qualitative traffic.

Distributed mechanism can improve the efficiency of resource utilization, and possibly could support dynamic admission control. But, it might cause additional control overheads in the interior nodes, and the implementation should be deployed all the intra-domain. The qualitative traffic service may be well supported by a distributed resource provisioning mechanism. As an example of the distributed approach, the 2-bit resource allocation scheme[4] used 2-bit currently unused bits in the DS field, and provides a simple, lightweight resource control mechanism. The edge node in this scheme signals to the interior nodes by marking the unused bits of a data packet, so that each node can inform on the packet whether it can admit or not on the basis of maximum flow size. The resource reservations or refreshments are done by counting the number of marked packets, each packet being a basic unit for the resource amount control. Although the scheme has suggested some useful probing concepts, the scheme has some limitations in the control functionalities due to the limited number of control bits. It cannot denote the required resource quantity when probing. It also might cause potential compatibility problem with the Explicit Congestion Notification(ECN) mechanism. The concept of maximum flow size may be a constraint for the non-homogeneous environment, so fine-grained control in actual networks would be limited.

3. Dynamic Admission Control Scheme

This paper proposes a measurement-based admission control scheme for dynamic resource provisioning in the DiffServ domain. In this scheme, some considerations for the DiffServ domain characteristics have been reflected. So that being in accordance with the DiffServ concepts, the proposed scheme poses essential role onto the edge nodes, while small overhead is put onto the core nodes. The basic elements of the proposed admission process are real-time traffic measurements in each node, edge-to-edge resource probing along the routing path and status gathering via control packets, and admission decisions based on the measured status.

The proposed scheme is intended to support resource control for the service classes which have traffic bursts and so the prediction of its dynamic characteristics is difficult due to the bursts. The edge-to-edge resource probing process is shown in figure 2. When a customer requests a service in EF or AF class with some traffic profiles, the ingress node takes the role of admission decision. The ingress node first sends a control packet to probe the intra-domain resource status, and then the core nodes, along the routing path to the edge node, mark their available resource status into the control packet if its resource is smaller than that of the packet. In order to support these control or management packets, some DSCP codes are defined as the control packet indicator. Several codepoints among DSCP codepoints pool-2 are appropriate for this control plane purpose, and we call it control codepoints.

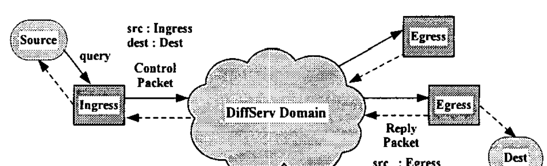

Fig. 1 Edge-to-Edge Route Resource Probing

The core nodes measure, in a background job, AF qualitative traffic load in their queues occupation using RED variant such as RIO mechanism. When a control packet is received, core nodes identify the actions to take according to the control codepoint, and update the required resource status information into the control packet if its own is smaller. The egress node responds to the ingress node with the probed resource status, and then the ingress node replies to the source with the admission decision. In this way, the minimum available resource information along the edge-to-edge route is reported to the ingress node in real-time, and based on that information, of which process is carried out through control packets that use control DSCPs. Here, the edge-to-edge path probing is assumed to route the same path with that of the traffic data that will follow.

In order to support dynamic admission control, real-time measurement of the resource usage at every node is important, and the targets of measurements are the queuing status and bandwidth utilization. The queuing measurement can be carried out using the existing queue management scheme, RIO in case of AF traffic class, where the RIO uses EWMA(Exponentially Weighted Moving Average) to update average queue occupation, which can be available as one of the metrics for the resource usage level. This average is smoothed to get some mean value. To get bandwidth utilization needs some statistical analysis, which can be assumed to be applicable to many flows aggregated. Gaussian approximation can be applied to the heavy-tail traffic since the AF traffic is conditioned at the ingress node by the conditioner. The techniques is used to get target quantile from the mean and variance of the bandwidth utilization. The estimated quantile offset relative to the link or traffic class bandwidth determines the available bandwidth for the new admissions, taking also the average queue occupancy into account. Generally, the heavy-tail traffic tends to generate large variance and thus utilization is low compared to the non-heavy tail traffic.

The proposed scheme can be easily extended to support several functions. Basically, the EF class service needs resource reservation and periodic refreshment of the reservations. For the purpose of AF class background monitoring, periodic refreshment event can be used to gather AF class resource status and drop status as the background task. The drop status information of EF and AF classes can be used as the feedback for the admission decision control. The background monitoring and drop status feedback via the periodic refreshment packet is for the fine-grained control of the AF class, and to reduce admission decision errors. In this regard, the period of the refreshment is chosen to be small(a few seconds), while the overhead of it is kept small.

The reservation events for the DiffServ EF traffic are processed in aggregated flow based, and the periodic reservation refreshment carries the full reserved resource to refresh. A new packet field codes are defined for the control packet to manage the admission control & resource probing, and are used to represent the control events and to carry resource status. Newly defined control DSCP codes for control or management plane can be used as the control message event id for the probe signaling purpose. Since each DiffServ domain needs its own intra-domain resource control mechanism for effective provisioning, and further, the data plane PHB definitions are not expected to explode because it is reported that the service differentiation into many service classes is difficult to be effective in the implementations, we think that this approach may well deserve its own value in the intra-domain.

When determining available resources, the ingress node manager applies different criteria according to the measured load level. A simple prediction mechanism is applied to the measured data to estimate next future available resource level. Based on the probed result, when the next estimation of the available bandwidth is less than the requested traffic profile, the admission decision is returned to the customer with reject. The result of the admissions can be reflected via feedback of the In_profile traffic drop ratios through periodic background refreshment events. In this manner, the degree of admission can be flexibly controlled by the provider policy.

The simple equations for the prediction are as follows.

$$S_t = \alpha Y_t + (1-\alpha) S_{t-1} \qquad 0 < \alpha < 1 \qquad (1)$$

$$S'_t = \alpha S_t + (1-\alpha) S'_{t-1} \qquad 0 < \alpha < 1 \qquad (2)$$

$$S_{t+1} = A_t + B_t \qquad (3)$$

$$\text{where,} \quad A_t = 2S_t - S'_t \qquad (4)$$

$$B_t = (\alpha/(1-\alpha))(S_t - S'_t) \qquad (5)$$

These are well-known double exponential smoothing equations used here to estimate and reflect the trend of the resource usage status, where equation (4) indicates the level, and equation (5) represents the estimated trend. From these data, the next future expected bandwidth can be obtained by the equation (3). The trend is reflected in the calculation of the path available resource in order to compensate for the lagging effect of double exponential smoothing and to predict next available resource level in the dynamic traffic environment.

4. Simulation Experiments

The simulation is performed using ns-2 simulator, and is implemented upon S. Murphy's DiffServ patch module, which can support some of the basic DiffServ functionalities such as traffic class-based scheduling based on the WRR (Weighted Round Robin), service differentiation according to the DSCP codepoints, RED/RIO queue management schemes, and token-bucket based traffic conditioner at front of the Ingress node, which marks In or Out profile for AF class traffic and drops EF class traffic which exceeds permitted rate. Using that patch module as the DiffServ data plane infrastructure, additional control & management plane functionalities for the admission control and resource provisioning are implemented on the simulator. The simulation topology is shown in Fig. 2.

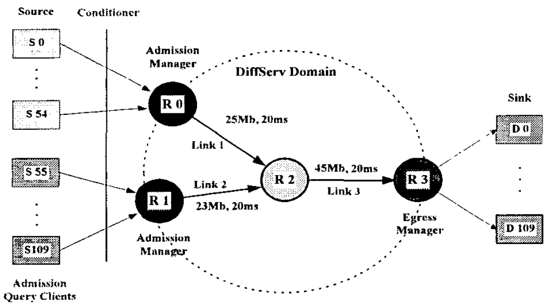

Fig. 2 Topology of the simulation architecture

The one hundred and ten traffic source nodes consists of 3 classes. There are 10 EF class source nodes, 60 AF class nodes and 40 BE class nodes. They are partitioned into 2 groups, where group 1 consists of 5 EF class sources, 30 AF class sources, and 20 BE class sources. Group 2 also consists of 5 EF, 30 AF, and 20 BE class sources. The weights of the three classes in the WRR scheduling mechanism are set to 1:5:4 for the EF:AF:BE classes. The BE class sources are FTP sources with TCP transport agents. While the BE class sources start traffic transmission at simulation start, the EF and AF class traffic sources should query admissions to the ingress nodes before starting data traffic transmission.

There are some choices in the parameters for the setup of the simulation. The three links are configured to 25Mbps with 20ms delay, 23Mbps with 20 ms delay, and 45Mbps with 20ms delay. The source access links are each 10Mbps with 0.1ms delays. Traffic sources, as application layer, are CBR for the EF class, Pareto or Exponential for the AF class, and FTP for the BE class. The transport agents for the applications are UDP for CBR and Pareto sources, and TCP for FTP sources. For the AF class, Pareto on/off source is associated with UDP agent, since TCP agent smoothens the bursts of the on/off source by self-adjusting its windows at packet drops, so that source traffic rate control is difficult. Through UDP, application traffic characteristics are almost completely reflected into the network. The traffic conditioner at ingress node either drop EF packets or remark AF packets according to the conformance test, where token bucket is set to absorb 30% of the short bursts over the AF source's nominated rate. Traffic sources with Pareto distributed on/off times are configured to have mean on/off periods of 3 seconds each, and shape parameter α of 1.5. The Pareto source rate is the on time rate, so it is assumed that the nominated rate used in the admission process is half of the Pareto source rate parameter.

In order to experiment the admission control scheme, we have implemented three kinds of agents; the source query clients at EF/AF sources, the admission manager at the edge nodes, and relay agents between the clients and manager on the ingress node. There are nine control events in the admission control scenario from source query to the admission decision reply. The simulation experiment is performed with all the BE class traffic sources and 2 EF sources and 10 AF sources are started at start time.

The admission scenario in the simulation starts with the admission queries with traffic profiles by the clients at the traffic source nodes, where the inter-query times are exponentially distributed with mean 1 seconds. The admission manager at ingress node stores the query profile and start probing sequence. The ingress manager gathers AF class resource status through periodic Refreshment events as a background task. The core node measures its resource usage as a background task, and when the probing event is arrived, it compares its available resource with that of the packet. If its available resource is smaller than that of the packet, the core node inserts its value into the control packet. The egress manager responds with an ACK packet to the probing event. In this way, the minimum resource is carried in the control packet along the edge-to-edge route to the ingress node. Receiving ACK from egress, the ingress manager decides admission based on the resource probed, and reply to the source that has issued the query.

If the source client receives admission decision with Accept, it starts data traffic transmission; if rejected, it retries with reduced traffic rate after exponentially distributed delay with mean 9 seconds. The retries can be permitted up to 7 times for each source client. When admitted in a retry, the ingress manager rebuilds the traffic conditioner according to the reduced traffic profile. The AF class queue management is performed by the RIO mechanism, and its parameters are the Early Drop thresholds for the In and Out_profiles, where the upper threshold of the Out_profile is lower than the lower threshold of the In_profile. In this simulation, 90% & 45% of the queue length are assigned for the In_profile thresholds, while 40% & 20% of the queue length are assigned to Out_profile thresholds.

In this simulation, a few variational models are setup and experimented. The variations are made by the combinations of the applications and the transports. Exponential or Pareto On/Off sources for the AF traffic are major concerns in this paper. Under the normal load, the EF/AF sources, which should get admissions, send admission request queries or retries in an exponentially distributed interval with mean 1 seconds. The period of the Refreshment packets is set to 2 seconds, and the simulation time is 40 minutes. Fig. 3 is the AF mean service rates as the result of the admission decisions for the exponential traffic sources along the exponentially distributed query sequence. Fig. 4 is the exponentially weighted moving average of the AF class queues occupations of the 3 links, according to the admission process. Fig. 5 shows the drop ratios of the In_profile AF traffic on the three links.

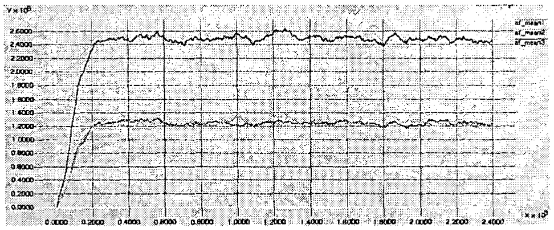

Fig. 3 AF mean service rates on the three links

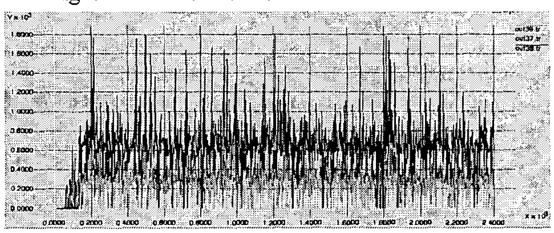

Fig. 4 Moving average for the In profile AF traffic

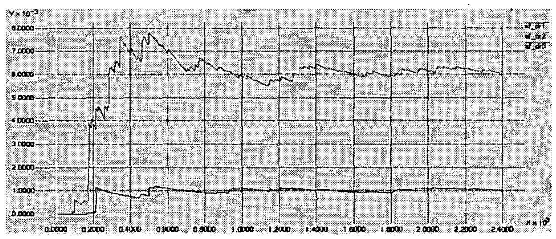

Fig. 5 AF In profile Drop Ratios on the 3 links

The Pareto sources also showed similar result, not shown here, indicating that the Gaussian approximation for the heavy-tail traffic applies reasonably. In these simulation with small group of sources, the results show that the link utilization in the steady state is approximately 88% on the link3 and 82~85% level on the links 1 and 2, with fairly small drop ratios.

These results are the case that In_profile AF traffic only is considered, and the Out_profile traffic is not cared about. If the In & Out_profiles are all counted, the utilization is near 100% with some large drop ratios. In this context, there can be some questions about the applicability of the RIO mechanism to the AF class queue management. How is the Out_profile traffic can be treated if the AF queue is managed by RIO scheme? We think that the is one of the problems on the AF class service differentiation.

The simulation experiment shows that the admission control can be flexibly managed by the provider via real-time measurements and resource probing that fetches out the interior resource status to the edge nodes, and applying simple prediction in estimating the available resource according to the load level. The admission result can be checked by the periodic feedback of drop ratios, so that dynamic and fine-grained resource provisioning can be possible.

5. Conclusion

In this paper, a dynamic admission control scheme has been proposed and simulated on the ns-2 simulator. The simulation experiments show that the control scheme can support fine-grained dynamic admission control for the DiffServ domain resource provisioning with flexible intra-domain resource control capability. The proposed scheme can deal with the dynamics of the AF class qualitative traffic through newly defined control or management plane DSCP codepoints and some protocol entities for the admission control process. Also the drop ratios and some AF class status gathered via periodic Refreshment events can be useful criteria for the provider to provision resources, hence the scheme can be easily extended to have more functionalities by the providers.

After some more experiments in scalable situations, extension to the inter-domain resource provisioning and RSVP interworking would be studied in the future.

Reference

[1] Y. Bernet, et al. "A Framework for Differentiated Services", Internet Draft, draft-ietf-diffserv-framework-02.txt, February, 1999.

[2] Grenville Armitage, "Quality of Service in IP Networks : foundations for a multi-service Internet", MTP, April 2000.

[3] Stefano Salsano, et al. "COPS Usage for Outsourcing Diffserv Resource Allocation", Internet Draft, draft-salsano-issll-cops-odra-00.txt, February 22, 2000.

[4] L. Westberg, "Load Control of Real-Time Traffic - A 2-bit Resource Allocation Scheme", Internet Draft, Oct. 23, 1999.

[5] Xipeng Xiao, Lionel M. Ni, "Internet QoS: the Big Picture",

[6] Kalevi Kilkki, "Differentiated Services for the Internet", MTP, 1999.

[7] D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2475, December 1998.

[8] R. Malghan, "An Architecture for Differentiated Services," RFC 2490, January 1999.

[9] K. Nichols, "Assured Forwarding PHB Group," RFC 2597, June 1999.

[10] V. Jacobson, "An Expedited Forwarding PHB," RFC 2598, June 1999.

[11] ns-2 manual.