

Forecasting Error Tolerable Resource Allocation for All-IP Networks

Yao-Nan Lien and Yi-Min Chen
Computer Science Department
National Chengchi University
Taipei, Taiwan, R.O.C.
lien@cs.nccu.edu.tw

Abstract This paper will discuss some resource allocation methods that can tolerate forecast errors under the Budget-Based management infrastructure, *BBQ*, which is designed to offer end-to-end QoS assurance for All-IP networks. *BBQ* takes pre-planning approach to forecast incoming traffic based on historic statistics and allocates link resources accordingly. Traffic forecast may not be perfectly accurate due to traffic fluctuation and imperfect forecast. Forecasting errors may lead to poor resource allocation. We have designed some mechanisms that can compensate forecasting errors and thus may reduce performance degradation accordingly.

Index Terms— All-IP Network, QoS.

I. INTRODUCTION[‡]

1.1 All-IP Networks

An *All-IP Network* uses a single IP based packet-switched network to carry all types of network traffics [1,8]. This revolutionary All-IP network not only reduces network deployment and management costs, but also offers a great opportunity opening for various new services that are not possible on the conventional separated networks. However, running time-sensitive services such as VoIP on packet-switched networks may suffer from poor quality problem due to long delay time, large jitter, and high packet loss rate. To make All-IP networks possible, QoS is a critical problem yet to be overcome [3,8].

Without loss of generality, we assume the following simplified All-IP network architecture. A worldwide All-IP network consists of several core networks interconnected together through some interconnection links (e.g. cross Pacific undersea cables/fibers) and some number of stub networks (also named *access networks*) connected to core networks. A *core network* consists of some *Interior Routers (IR)* and some *edge routers*. A *stub network* is connected (attached) via an *Access Gateway (AG)* to an edge router, called *Border Gateway (BG)*, of a core network. The network architecture is depicted in Fig. 1.

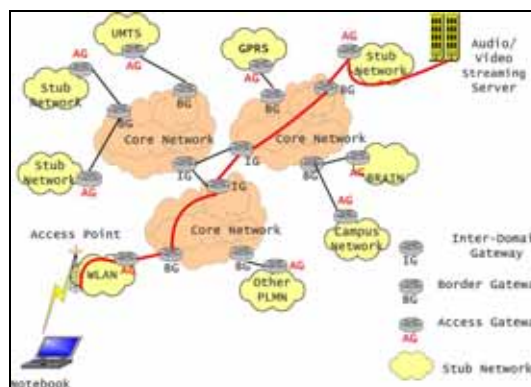


Fig. 1. Simplified All-IP Network Architecture

1.2. Related Work

The two most popular QoS technologies are Differentiated Services (DiffServ) and Integrated Services (IntServ) [2,4,5]. The heart of IntServ is RSVP (Resource Reservation Protocol) [4,5]. Before admitting a service request, IntServ first reserves demanded resources along the path selected for the request. It can provide end-to-end QoS with higher confidence, but it suffers from scalability due to tremendous management overhead. On the other hand, DiffServ can avoid scalability problem. DiffServ is a mechanism for specifying and controlling network traffics by their classes so that certain types of traffic, such as voice, get precedence [2]. The major advantage of DiffServ is its simplicity and easy to implement. However, it can only control class-based per-hop behavior. Extra mechanisms are necessary to guarantee per-flow end-to-end QoS [9].

The two most famous large-scale efforts that are trying to provide end-to-end QoS for All-IP Networks are TEQUILA and AQUILA [1,8]. TEQUILA (Traffic Engineering for Quality of Service in the Internet, at Large Scale) is a project partially funding by the European Commission [8]. The objective of the project is to study, specify, implement and validate a set of service definitions and traffic engineering tools to obtain quantitative end-to-end QoS guarantees. AQUILA (Adaptive Resource Control for QoS Using an IP-based Layered Architecture) is another European project aiming to similar objectives [1].

Paper Organization

The BBQ infrastructure is described in Section 2. Section 3

[‡] This project is sponsored by NSC grant 93-2213-E-004-010

shows various resource allocation methods that can tolerate forecast errors. Section 4 shows the performance evaluation for the designed methods.

II. BUDGET-BASED QoS MANAGEMENT ARCHITECTURE

2.1. Design Philosophy

Under Budget-Based End-to-end QoS management infrastructure, *BBQ*, the quality bound of each component network is controlled based on a calculated budget plan. End-to-end QoS will be guaranteed by a global QoS management agent. The objective of this infrastructure is to facilitate network operators to tune their networks with a great flexibility and scalability to achieve their own operational objectives.

In order to maximize network performance and to minimize service response time, many of management mechanisms in *BBQ*, such as resource allocation and reservation, take pre-planning approach, instead of real-time on-demand approach. Using pre-planning approach, a network planner uses historical traffic statistics to forecast incoming traffics in the future. In this approach, inevitable forecasting errors caused by traffic fluctuation and imperfect forecast may hurt management objectives. This paper will discuss several methods that can compensate the performance degradation caused by forecast errors.

The other two design principles are Class Based Service Policies and Path Centric Per-Flow End-to-End QoS Assurance.

2.2. End-to-End QoS Management Bearer Service Hierarchy

An end-to-end service is carried by many smaller bearer services. Each core network provides a short-path bearer service, and a long-path is the combination of all short-paths that a packet travels. Adding together the bearer services provided by the Entrance and Exit stub networks, an end-to-end bearer service is established. In this manner, end-to-end QoS is assured by piecewise QoS assurances.

Management Functional Hierarchy

Based on the simplicity principle, *BBQ* organizes the software agents in different network components into layered management hierarchy, namely, End2End Resource Coordination, End2End QoS Control, Sub-network Resource Management, and Sub-network QoS Control. The end-to-end QoS assurance responsibility is then decomposed into smaller pieces and distributed to many agents in different layers. With autonomous authority within the designated responsibility, each agent may make some decisions by themselves without any negotiation with other entities.

Distributed Long-Path Planning

The most important task in the End-to-End QoS

Management functionality is to plan a set of long-paths to meet the quality requirements of the anticipated service requests. Core network operators then provision their own core networks based on the forecasted demand. Since an All-IP network is a federation of many sub-networks, we assume there does not exist any global network planner to plan the entire network from the global viewpoint. Network planning has to be performed by all core network operators in a cooperative (distributed) manner, instead. The design detailed can be found in [6].

2.3. QoS Management for Core Networks

A core network is owned and operated by an independent operator. Under *BBQ* infrastructure, each core network is responsible for providing many QoS assured short-paths. The traffic flow that is sent into a short-path will travel along that short-path so that its quality will be guaranteed. The Global ACA of each stub network will perform admission control so that only designated amount of traffic is allowed to enter the network. Under this circumstance, each core network will be able to guarantee the QoS level for all admitted traffic flows.

A key design philosophy in designing QoS management scheme for core networks is to reduce the real time per-flow reservation overhead by batch order. Under this scheme, each edge router is pre-allocated with some short-paths and equipped with an admission control agent (ACA) to perform admission control autonomously. *BBQ* proposes several resource allocation mechanisms to improve resource utilization in order to achieve the maximum performance.

Software Architecture for Core Network

We assume each core network has DiffServ-like capability. There is a central management module handling centralized policies such as resource allocation, etc. There are three components in the central management module *Core Network Coordinator* (CNC). Within CNC, there is a *Bandwidth Broker* (BB), which controls and allocates link resources to Ingress routers. In each Ingress, there is a *Bandwidth Order Agent* (BOA), which acquires link resources from BB and turns it to other agents to construct short-paths.

Short-Path Planning Procedure

Path planning can be done in a centralized manner for simplicity. However, a centralized planning approach may induce fairness problem such that some Ingress may be allocated excessive resources while some may be under allocated. On the other hand, a distributed planning approach may have less fairness problem and higher capability to deal with local traffic fluctuation, but is more complicated. *BBQ* proposes to use a hybrid approach. An agent in CNC, *SPPA*, initiates the short-path planning and allocation process first. If the result is acceptable according to some fairness criteria, the process terminates. Otherwise, a distributed process is initiated.

Link Resource Allocation by Batch Order

In the Distributed Phase of short-path planning, each BOA must calculate and acquire the right amount of bandwidth to

fulfill the demands of anticipated TSCO traffic requests for each link. Since real-time on-demand resource allocation is usually more expensive than pre-allocation, an insufficient resource allocation may induce extra cost for real-time on-demand resource acquisition. On the other hand, an excessive resource allocation will waste unused resources. We have designed an optimization model aiming to maximize the expected profit as well as their solutions for BOA. The details can be found in [9].

When the BB receives the orders submitted from BOAs, it needs to determine the right amount of bandwidth to fulfill these bandwidth orders. The basic allocation policy is as straightforward. If the remaining bandwidth of a link is more than the sum of orders, it just allocates the demanded bandwidths to these orders. Otherwise, it divides the remaining bandwidth proportional to the demanded bandwidth and allocates them accordingly.

III. COMPENSATION FOR FORECASTING ERRORS

Traffic forecast may not be perfectly accurate due to traffic fluctuation and imperfect forecast. Forecasting errors may lead to poor resource allocation such that a network may reject some service requests while it actually has some available resources held unused. We have designed several mechanisms that can compensate forecasting errors and thus may reduce performance degradation accordingly. The objective is to maximize resource utilization in terms of operational objectives [6,7].

Under BBQ, link bandwidths are allocated to edge routers and then to short-paths. Two straightforward approaches are *central pool* and *distributed reallocation*. The central pool approach is to keep a small portion of link bandwidth at the BB for real-time on-demand allocation and to have edge routers make on-demand reservation when the preallocated bandwidth of any link is exhausted (or below a certain threshold). This approach has some drawbacks. First, on-demand resource allocation is expensive. Secondly, it is not easy to determine the right amount of bandwidth to be reserved in the central pool for real-time on-demand allocation.

On the other hand, in the distributed reallocation approach, edge routers perform bandwidth reallocation by themselves. Whenever the bandwidth of a link allocated to an edge router is exhausted, that router will negotiate with other edge routers for bandwidth reallocation. This greedy approach has less computation overhead. However, its reallocation efficiency is not predictable at all while it can't avoid the expensive real-time on-demand management overhead.

Under BBQ infrastructure, we designed an innovative mechanism, *overbooking*, in which a BB can commit more resources to the requesters than what it actually possesses. Assuming that some edge routers may have some unused bandwidth due to forecasting error, we may be able to control the level of over-commitment such that the total actual bandwidth consumption for a link will not likely exceed the link capacity. The challenge is to determine the right level of

over-commitment such that bandwidth demands can be better fulfilled and the probability of traffic overflow is minimized. We have designed an optimization model with good solutions provided. The central pool and overbooking approaches will be elaborated in the rest of this paper.

3.1. Central Pool

As mentioned above, the challenge to the central pool approach is to determine the right amount of bandwidth to reserve in the central pool. We have developed a very basic model for operators to use. More delicate models will be needed for practical use. Some notations are defined as follows:

- θ total requested bandwidth
- τ total committed bandwidth
- C_1 per unit cost of pre-allocated bandwidth
- C_2 per unit cost of on-demand allocated bandwidth
- C_1' per unit cost of on-demand allocated bandwidth that is owed by BB.
- P expected profit
- N maximum possible incoming traffic
- p_i probability of actually demanding i units of bandwidth in total

Note that the bandwidth owed by BB is the bandwidth that is requested by an edge router in the preplanning phase but not allocated by BB. The value of C_1' must be smaller than C_1 . Otherwise, BB may just hold the entire link bandwidth for on demand request in order to maximize the bandwidth utilization. Assuming the probability distribution of the total bandwidth demand is known and applicable, it is very straightforward to obtain the formula to compute the expected profit as follows.

$$P = C_1 \times \tau + \sum_{i=\tau}^{\theta} p_i \times C_1' \times (i - \tau) + \sum_{i=\theta}^N p_i \times [C_2 \times (i - \theta) + C_1 \times (\theta - \tau)] \quad (1)$$

The optimum amount of bandwidth to keep in the central pool can be easily calculated using Eq. (1). The derivation of Eq. (1) can be found in [7].

3.2 Overbooking

3.2.1 Basic Ideas

As mentioned in the beginning of this section, by overbooking, a BB can commit more resources to the requesters than what it actually possesses. Assuming that some edge routers may have some unused bandwidth due to forecasting error, we may be able to control the level of over-commitment such that the total actual bandwidth consumption for a link will not likely exceed the link capacity.

3.2.2 Optimum Level of Overbooking

The challenge is to determine the right level of over-

commitment such that bandwidth demands can be best fulfilled and the probability of traffic overflow is minimized. We have designed an optimization model with good solutions provided. Extra notations are defined as follows:

- m_1 per unit profit earned by overbooked bandwidth
- m_2 per unit packet loss penalty due to overflow
- R net profit
- $p(b)$ probability of actual traffic demand of b units
- $h(b)$ loss rate on the committed bandwidth of b units
- C link capacity

Assuming the probability distribution of total bandwidth demand is known and applicable, we can derive the formula to maximize the expected profit as a function of the committed total bandwidth:

$$R = m_1 \times (\tau - C) - m_2 \times \left[\sum_{b=C}^{\tau} p(b) \times h(b) + \sum_{b=\tau}^N p(b) \times h(\tau) \right] \quad (2)$$

The derivation of Eq. (2) can be found in [7].

IV. PERFORMANCE EVALUATION

The proposed central pool and overbooking methods were evaluated using a simple simulation model on the UCB NS2 network simulation environment. Although the evaluation model is simple, it is good enough to demonstrate the merit of overbooking method. The evaluation metrics are the total profit and the admission ratio, which is the ratio of the admitted traffic to the total incoming traffic

4.1. Experiment Environment

Fig. 2 shows the basic topology used in the simulation. To simplify the simulation, we use CBR traffic and the capacity of the link between Ingress and Egress is 50 units. Although we use CBR traffic only, the results should be applicable to other types of traffic. Each of the traffic source nodes has an access link to the Ingress. Each access link has a designated flow capacity. The traffic is randomly generated and is fed into these access links.

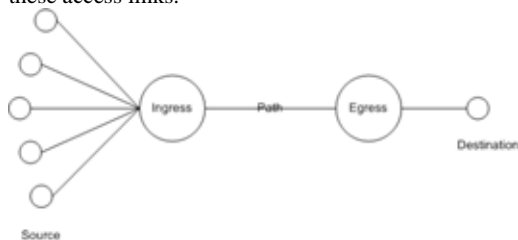


Fig.2. Network Topology in Simulation

To simulate central pool method, another "central pool" node is added to the network. The capacity of each access link is reduced by a designated ratio and the overflowed traffic at

each source node is directed to the "central pool" node to simulate on-demand bandwidth allocation.

The forecasted traffic demand of each of 5 traffic sources is 10 flow units. The traffic is generated using shifted normal distribution with a mean of 10 flow units and in the range between 0 and 20. The deviations are controlled parameters in the experiments.

4.2. Experiment Results

4.2.1. Central Pool

The traffic is generated with a deviation varying from 1 to 5, increased by 1 each step. The ratio of the bandwidth reserved in the central pool varies from 10% to 50%, increased by 10% each step. The ratio of $C_2:C_1:C_1'$ is set to 6:2:1. The evaluation results are shown in Fig. 3 and 4.

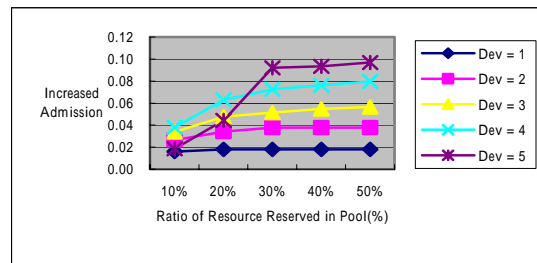


Fig. 3. Increased Admission Rate (with vs. without Central Pool)

As we can see from the results shown above, as the ratio of the bandwidth reserved in the central pool increases, admission ratio is increased too. However, due to higher on-demand cost, it is not appropriate to reserve too much resource at the central pool as we can see from Fig. 4.

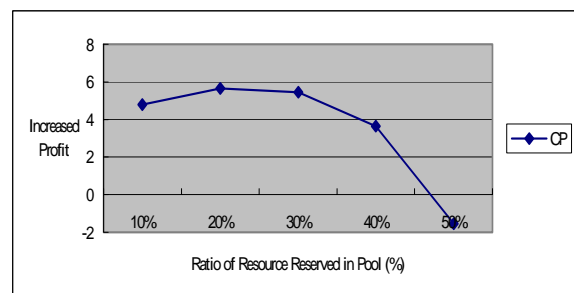


Fig. 4. Increased Profit Earned by Central Pool

4.2.2. Overbooking

The traffic is generated with a deviation varying from 1 to 5, increased by 1 each step. The ratio of overbooked bandwidth varies from 10% to 50%, increased by 10% each step. The ratio of m_1 to m_2 is set to 1:6. The evaluation results are shown in Fig. 5 and 6.

As we can see, overbooking performs extremely well when forecasting errors (traffic variation) are high.

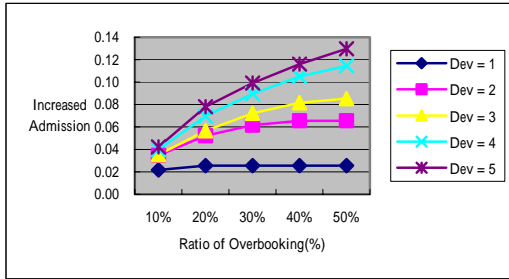


Fig. 5. Increased Admission Rate (with vs. without Overbooking)

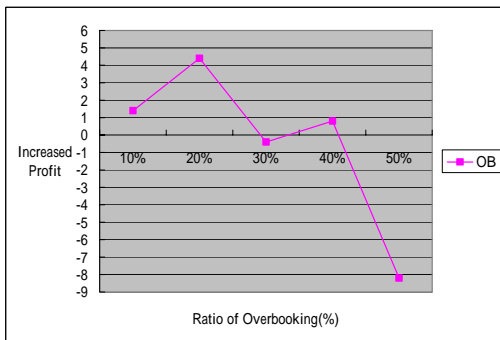


Fig. 6. Increased Profit by Overbooking Ratio

4.2.3. Comparison of Central Pool and Overbooking

The comparisons between Central Pool and Overbooking are shown in Fig. 7 and 8. Note that the ratio of bandwidth reserved in Central Pool and the ratio of overbooking are different parameters. They may not be meaningful by comparing them on the "ratio" scale. Reader must read the results with caution.

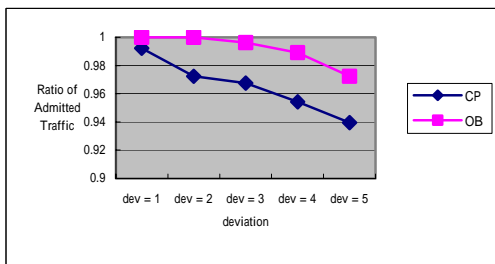


Fig. 7. Comparison of Central Pool and Overbooking

As we can see from the Fig. 8, overbooking approach is better than central pool approach when the traffic has higher degree of fluctuation.

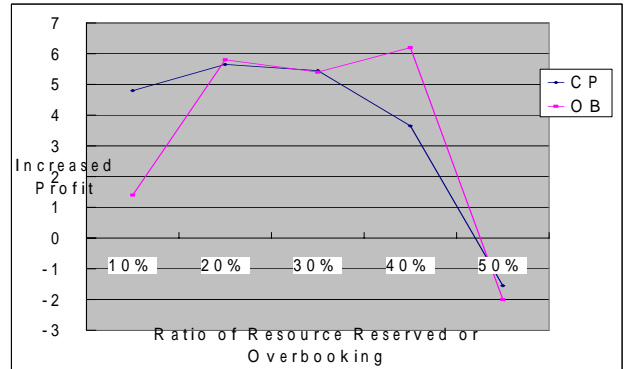


Fig. 8. Comparison of Central Pool and Overbooking

V. CONCLUDING REMARKS

This paper discusses some resource allocation methods that can tolerate forecasting errors under the BBQ infrastructure. BBQ takes pre-planning approach to forecast incoming traffic based on historic statistics and allocates link resources accordingly. However, forecasting errors may lead to poor resource allocation. We have designed some mechanisms that can compensate forecasting errors, namely, central pool, distributed reallocation, and overbooking. We also derived some optimization models to determine the right amount of resources to keep in the central pool for on-demand allocation and right level for overbooking that can maximize resource utilization without causing any real-time on-demand allocation overhead. The simulation results based on the UCB NS2 shows that both methods can improve resource utilization.

REFERENCE

1. AQUILA, <http://www.salzburgresearch.at>.
2. D. Black, M. Carlson, E. Davies, Z. Wang, "An Architecture for Differentiated Services", RFC 2475, Dec. 1998.
3. Janusz Gozdecki, Andrzej Jajszczyk, and Rafal Stankiewicz, "Quality of Service Terminology in IP Networks", *IEEE Communications*, Mar. 2003.
4. IETF RFC 1633, Integrated Service Framework (IntServ).
5. IETF RFC 2205, Resource reSerVation Protocol (RSVP).
6. Yao-Nan Lien, Hung-Chin Jang, Tse-Chieh Tsai and Hsing Luh, "BBQ: A QoS Management Infrastructure for All-IP Networks", *Communications of IICM*, vol. 7, no. 1, Mar. 2004, pp. 89-115.
7. Yao-Nan Lien, Yi-Min Chen, "Forecasting Error Tolerable Resource Allocation in Budget-Based QoS Management for All-IP Core Networks", NCCU-CS Tech. Report, Sep. 2003.
8. TEQUILA, <http://www.ee.ucl.ac.uk/~pants/projects/tequila/>.
9. Michael Welzl, Max Muhlhauser, "Scalability and Quality of Service: A Trade-off?", *IEEE Communications*, June 2003.