

行政院國家科學委員會補助專題研究計畫成果報告

提供一個 TCP 友善的 ABR 流量控制

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC89-2218-E-009-108-

執行期間： 89 年 10 月 1 日至 90 年 7 月 31 日

計畫主持人：廖維國

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學電信工程系

中 華 民 國 90 年 10 月 23 日

行政院國家科學委員會專題研究計畫成果報告

國科會專題研究計畫成果報告撰寫格式說明

Preparation of NSC Project Reports

計畫編號：NSC 89-2218-E-009-108

執行期限：89年10月1日至90年7月31日

主持人：廖維國 國立交通大學電信工程系

計畫參與人員：施邦欣 國立交通大學電信工程系

林尚億 國立交通大學電信工程系

王聰瑞 國立交通大學電信工程系

1. 中文摘要

這篇報告顯示了本年度計劃我們研究所得到的結果。根據我們的結果顯示，要提供一個 TCP 友善的 ABR 流量控制，對於 ABR 來源端 (Source) 緩衝區的管理主導著整個連線效能 (產量以及存取的延遲) 的高低。我們從兩個方面來著手，第一種對來源端緩衝區管理方法稱為 naïve，需要有關於往返延遲、TCP 快速重傳及復原機制的詳細資訊。第二種方法只需要使用代理伺服器的快取作為來源端的緩衝區。有了代理伺服器的快取，一個 TCP 的連線可以被分割成數個獨立的 TCP 連線，如此流量控制得以簡化。因此對於代理伺服器快取的管理成為設計的瓶頸 (重心)。利用嚴格的機率分析，我們得到了管理代理伺服器快取的一個近似最佳化的策略，稱作預置代理伺服器 (proxy prefetch) 以及字首 (前綴 prefix) 快取。

關鍵詞：TCP 友善，ABR 流量控制，全球資訊網，代理伺服器快取，預置

Abstract

This report demonstrates the results we have obtained during this project year. By our results, we conclude that to provide the TCP-friendly ABR flow control, the capability of buffer management in the ABR source dominates the performance (throughput and access latency) of the connection. Two approaches are therefore forked. The first approach for source buffer management is naïve and needs the detailed information about the round trip delay and TCP fast retransmission and recovery scheme. The second approach is simply using the proxy cache as the source buffer. With the proxy cache, a TCP connection is segmented to several independent TCP connections and thus the flow control scheme is simplified. Therefore, the management of proxy cache turns out to be the bottleneck of the design path. By using a rigorous probability analysis, we derive a near-optimal strategy called proxy prefetch and prefix caching for the purpose of proxy cache management.

Keywords: TCP-friendliness, ARB flow control, World Wide Web, Proxy Caching, Prefetching.

2. Motivations and Objectives

Nowadays most of the loss-free transmissions over Internet are realized by the transmission control protocol (TCP). The jobs of TCP are 1) to improve the network effective throughput subject to the fairness constraints and 2) to retransmit the packets upon request. To satisfy these goals, the strategies to increase and decrease the amount of TCP segments injected into the network should be carefully designed. In general, TCP takes actions only based on whether loss events detected and some history information kept in the sender.

A TCP connection may go through a set of different types of networks. These networks are interconnected and geographically dispersed. For example, a TCP connection may start from a host that resides in the Ethernet-based LAN, then via a gateway interfaced with ATM network, and then a gateway connecting the destination host. We will consider the case when a TCP connection goes through a ATM network, or so called *TCP over ATM transmission*.

The ATM networks enable the integration of traffic with different characteristics. The variable bit rate (VBR) and constant bit rate (CBR) are defined in ATM to support the multimedia traffic, such as those generated

from the compressed video sources. The available bit rate (ABR) and unspecified bit rate (UBR) services are defined to convey the *elastic traffic*. The ABR service aims at utilizing the leftover bandwidth left by the CBR and VBR traffic. The UBR services then attempt to utilize the bandwidth spared by all other services. Literatures on TCP over ATM transmission focus on the TCP over ABR or UBR transmission since the elastic traffic is the traditional type supported by TCP/IP Internet. However, most of them are along the line of the stand-alone approach. They are interested in the performance study of various management schemes for TCP over ABR or UBR transmission excluding the existence of the VBR traffic and other types of ABR or UBR traffic in the ATM networks. Only few papers [1,2] study the TCP over ABR transmission in the presence of the VBR traffic. We will step ahead toward the study on TCP over ABR transmission while the VBR and other types of ABR traffic co-exist in the ATM network.

The difficulty in developing TCP over ATM transmission while other types of ABR traffic co-exist in the network is caused by the fairness and bandwidth under-utilization problems. TCP will decrease its sending capability whenever a loss is detected. During this interval, either the non-TCP sources will take all of the leftover bandwidth or simply the leftover bandwidth cannot be utilized by any connection. A flow control able to solve such a problem is called *TCP-friendly flow control*. While VBR traffic co-exists in the network, the difficulty arises from the complex correlation structure of VBR traffic [5], which will lead to long-interval bursty traffic and may result in multiple TCP segments lost. It is emerging to develop a feasible solution to attack these difficulties.

3. Results and Discussions

We separate our results into two parts. The first part is called the naïve source buffer management and the second part is based on the technique of web proxy caching and is

called proxy prefetch and prefix caching.

3.1. Naïve Source Buffer Management

In the first part, we use a loss ratio approximation formula to comprehend the interaction between these techniques and VBR traffic. Based on these techniques and the formula, we propose a method called source packet discard (SPD) for ABR source buffer [3]. As simulation shows, SPD turns the TCP flows into a robust flows and thus the convergence of max-min fairness can be assured by using any ER-based ABR flow control which can guarantee such a convergence. Therefore, we show that in presence of VBR traffic, the criteria of TCP-friendliness can be achieved by the existing techniques. Our experience also shows that the size of source buffer should be proportional to that of switch buffer to support TCP-friendliness. In addition, it is almost impossible to fully utilize the bandwidth left by the high-priority traffic while the TCP-friendliness is also guaranteed.

3.2. Using Web Proxy Cache as Source Buffer

The major fraction of all network traffic today comes from applications in which clients retrieve objects from servers (e.g., the World Wide Web). Therefore, the other alternate to improve the performance of TCP is to use the Web proxy cache as the ABR source buffer. A *proxy cache* is an application gateway lying in the middle of the path from a client to its WWW server. The proxy cache intercepts client requests and, if it has cached the requested web objects in its *cache store*, replies the requested objects to the client directly. Otherwise, it forwards the requests to the upstream servers. These upstream servers could be another proxy cache or the original server itself. With the proxy cache in place, one could anticipate that most of the requests are served by the proxy cache, without having to go all the way to the original server. The response time of the requests can thus be

cut, and the traffic from the server to the proxy can be reduced, so can the server's load.

We study the design of proxy caches that are deployed near the clients to segment the original TCP connection to two independent connections, one is from the server to the proxy cache, and the other is from the proxy cache to the client, and focus on the latency reduction aspect of such design. Note that the TCP connection from the proxy cache to the client is usually with short round trip delay and could be lying on top of an ATM ABR connection if ADSL system is adopted in the local loop system.

Many techniques have been proposed to further enhance the latency reduction capability of proxy caches. One set of techniques that have attracted much attention is *proxy prefetch caching*, i.e., those techniques that prefetches and caches web objects in the proxy cache in anticipation of their requests. Thus, when the objects are actually requested, they can be served from the proxy cache directly, whereby masking the latency in accessing them from the upstream servers. *Proxy prefetch caching* prefetches web objects into proxy caches in anticipation that they might be requested. Upon arrival of actual requests, the object can be served directly from the proxy cache, whereby reducing the latency in serving web requests. A key design issue in prefetch caching is to maximize prefetch accuracy and, at the same time, minimize the stress on system resources. In [4], we examine the plausibility of combining proxy prefetch caching with *proxy prefix caching* to reduce resource stress, especially on network bandwidth and cache space. The basic idea of proxy prefix caching is to cache only the first portions of objects, called the *prefixes*. Whenever the object is requested, the proxy will send the cached portion to the client first and transfer the remainder from the upstream server. We then argue that effective *proxy prefetch and prefix caching* requires tight integration with system resource management, which in turn demands all

objects, cached or to be prefetched, be evaluated under a unified metric and their consumption on resources be accounted for. We show how such a metric, called *object profit*, can be derived and incorporated into prefetch prediction and cache replacement. Based on the metric, a near-optimal greedy strategy for prefetch/prefix caching and cache replacement is introduced.

4. Self-Evaluation

From doing the project, we have identified that the source buffer management is pivotal in the TCP-friendly ABR flow control. We have obtained several significant results, such as the proxy prefetch and prefix caching, and we believe they are worthwhile for the further publication on the international journals. Efforts for doing so are now underway.

5. References

- [1] S. Kalyanaraman and et al., "Performance of TCP over ABR on ATM backbone and with various VBR traffic patterns" *ICC'97*, pp. 1035-1040, 1997.
- [2] S. Kalyanaraman and et al., "Performance of TCP over ABR with Long-Range Dependent VBR Background Traffic over Terrestrial and Satellite ATM networks", *LCN'98*, pp. 11-14, 1998
- [3] W.-K. Liao, "Towards A TCP-Friendly ATM ABR Flow Control", Technical Report, 2000.
- [4] W.-K. Liao and C.-T. King, "Proxy Prefetch and Prefix Caching," *ICPP01'*, 2001.
- [5] W. Willinger and V. Paxson and M. S. Taqqu., "Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic", *A Practical Guide to Heavy Tails*, pp. 27-53, Birkhauser, 1998.