マルチメディア通信と分散処理 97−34
コンピュータセキュリティ 8−34
(2000. 3. 22)

2階層からなる適応型トランスポートアーキテクチャ†

戸辺 義人[1] 田村 陽介[2] 徳田 英幸[3]

[1] 慶應義塾大学 SFC 研究所 [2] 慶應義塾大学大学院政策・メディア研究科 [3] 慶應義塾大学環境情報学部

あらまし

インターネットにおいて、帯域保証をしようという試みがあるものの、実際には最善サービスで利用されることが多い。こうした状況で、連続メディア通信をインターネット上で行うには、ユーザの性能を最大にしたいという要求と、ネットワーク内部での公平性維持という要求の相反する要求を満足しなければならない。本稿では、その解決策として、マクロ適応とマイクロ適応の2階層からなる適応型トランスポートアーキテクチャを提案する。マイクロ適応はTCP公平性を維持する制御、マクロ適応はアプリケーションレベルでの性能を最大にする制御を行う。無線LANおよび日米インターネット回線で実験を行った結果、本アーキテクチャの有効性が示された。

キーワード プロトコル, QoS, TCP/IP, 連続メディア

Two-Layer Adaptive Transport Architecture

Yoshito Tobe[1] Yosuke Tamura[2] Hideyuki Tokuda[3]

[1]tobe@sfc.keio.ac.jp Keio University, 5322, Endo, Fujisawa-shi, Kanagawa, 252-8520 Japan,
[2]tamura@mag.keio.ac.jp Keio University, 5322, Endo, Fujisawa-shi, Kanagawa, 252-8520 Japan,
[3] hxt@sfc.keio.ac.jp, Keio University, 5322, Endo, Fujisawa-shi, Kanagawa, 252-8520 Japan.

Abstract.
Although there have been efforts to provide a bandwidth-guaranteed service over the Internet, a best-effort service is most commonly used in reality. In such a situation, continous media communications over the Internet must satisfy two contradictory requirements: fairness of rates from the network and maximization of a rate from the user. We propose two-layer adaptive transport architecture. It consists of micro and macro adaptation. Micro adaptation ensures TCP-friendliness, while macro adaptation selects a behavior to maximize the application-level throughput. The proposed architecture was shown to be effective over both a wireless LAN and the Internet.

Keywords: protocol, QoS, TCP/IP, continuous media

† この研究は，情報処理振興事業協会 (IPA) 実施による次世代アプリケーション開発事業「次世代アプリケーション共通基盤 RT-HDI (Real-Time Human Device Interaction) プラットフォームの研究開発」プロジェクトのもとに行われました．

1 Introduction

Recent advances in computer technology, digital signal processing applications, and packet network capacity have accelerated the proliferation of real-time, packet-based continuous media (CM) communications. Furthermore, the growth of the Internet has fueled demand for these technologies and highlighted the difficulties of handling CM in a wide area network. Although the Internet community has worked and is working to establish a mechanism of bandwidth-guaranteed communications [1], best-effort communication is more common at present and also, most likely to continue to be used in the future. CM best-effort flows mostly use RTP/UDP. Although RTP was designed to accommodate multicast communications, a common way of use at present is unicast CM communications. Therefore we limit ourselves to problems pertaining to unicast RTP-type CM communications.

In such RTP-type communications, retransmission has seldom been considered. However as suggested in [4], in some cases, retransmission can be useful. When one Application Data Unit (ADU) consists of many packets, retransmission may help reconstruct the ADU of which only a few packets were lost, for instance, due to a wireless link error. Although a new retransmission mechanism, potentially using RTCP which is a control protocol for RTP, may work to some extent, generated data including retransmit packets at a CM source are required to satisfy TCP-friendliness [2]. This leads to an idea of simply using RTP/TCP instead of RTP/UDP.

Using TCP in all cases is, however, inefficient. When the network condition is stable and a CM can correctly regulate its rate so that packets will not be lost, UDP is sufficient and outperforms TCP in reducing delay and jitter. Based on the consideration, we introduce a **Dual-Data Single-Control (DDSC) channel** for adaptive unicast CM communications. The DDSC channel consists of an RTP/UDP and a RTP/TCP data sub-channels and an RTCP/UDP control sub-channel.

Given a DDSC channel, we must define when and how the two data sub-channels are used. We divide the operation of the CM sender into two: ARQ and non-ARQ mode. The basic strategy of selecting which sub-channel is used is as follows: TCP is chosen if the ADU throughput is expected to be raised with retransmission in ARQ mode. Otherwise UDP is chosen in non-ARQ mode. Even in non-ARQ mode, TCP is intermittently used to probe for a TCP-equivalent rate [7]. Likewise, UDP is intermittently used to probe for error status of the path between the sender and its corresponding receiver. The adaptation by changing the mode is referred to as **macro adaptation**, while adjustment of rate to a TCP-equivalent rate in non-ARQ mode is called **micro adaptation**. We propose a transport architecture consisting of macro and micro adaptation, **Adaptive Transport Architecture (ATA)**.

We also propose receiver's assistance to congestion control. A rate of RTP/UDP sender is designed to be controlled based on the packet loss ratio reported by RTCP Receiver Reports (RRs). However such a scheme has two deficiencies. First, the rate is unfavorably reduced when packet loss occur due to wireless link errors. Second, when the RTP flow competes with multiple TCP flows, packets of the RTP flow may not be lost because of the faster response of the TCP flows, which hide congestion from the RTP flow. To overcome the problem, the receiver shall classify the path status depending on the change in delay (which is smoothed in our scheme) as well as packet loss. The sender receives the path status instead of the packet loss ratio alone, thereby determining the control of its rate.

We designed and implemented ATA in FreeBSD. We compared ATA with Loss-Delay based Adjustment (LDA) [6] in experiments and showed that our scheme outperforms the two in a wireless LAN and the Internet. Especially the rate of ADU is raised where an LDA receiver cannot reconstruct any piece of ADUs in a heavy packet loss state.

2 Related Work

There has been an abundance of research about TCP-friendly schemes. LDA [6] algorithm utilizes RTP for feedback information. Using RTCP, the control protocol for RTP, results in execution of rate control in a coarser period than the magnitude of round-trip time (RTT). LDA has additional features to estimate the TCP-equivalent rate at the sender; RTT is measured by the reaction of an RTCP RR to an RTCP Sender Report (SR) and the bottleneck bandwidth is measured by producing Packet Pair at the sender. Additive Increase/Multiplicative Decrease is performed depending on packet loss periodically. TFRCP [3] is based on a more precise formula of estimating TCP throughput developed by the authors. In the formula, the fast retransmit and the retransmission timeouts are taken into account. Rate Adaptation Protocol (RAP) [5] adopts a fine-grain Additive Increase/Multiplicative Decrease control that uses ACKs (in a manner similar to TCP) to estimate RTT and detect packet loss. The sender of RAP adjusts it rate every round trip time. The authors also propose to use the ratio of long-term and short-term average of RTT to tune the rate more precisely. We also propose a scheme called Rate Probing Based Adaptation (RPBA) [7].

Although these approaches toward a TCP-friendly control are important, there is no consideration for the interaction between applications and the transport protocol. In this paper, we assume more general continuous media data over RPBA.

3 Adaptive Transport Architecture

The adaptation of ATA consists of two sub-layers of micro and macro adaptation as illustrated in Figure 1.

- Micro adaptation
  The objective of micro adaptation is to maintain a TCP-friendly rate. Therefore the control is conducted in a short-term on a magnitude of the order of 100 ms to 1 s. The micro adaptation is always hidden from the user and transparent to its upper layers.
- Macro adaptation
  The objective of macro adaptation is to maximize the application-level performance under the constraint of rate regulated by the micro adaptation. Macro adaptation is triggered less frequently. There is a need for macro adaptation because the characteristics of packet loss may vary under the same rate. Since the rate itself means the throughput at the transport level, it does not necessarily

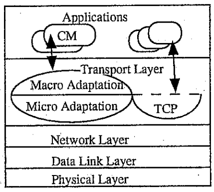

Figure 1: Macro and micro adaptation

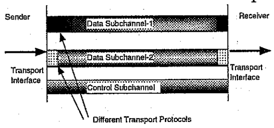

Figure 2: DDSC channel

represent the throughput at the application level. Sometimes, retransmission of lost packets give higher throughput for the application. Hence, the macro adaptation deals with the selection of the behavior of the transport protocol. This includes the determination of whether or not the retransmission for data reliability is conducted. Ideally, macro adaptation can be extended to accommodate calibrating parameters such as the window sizes of an existing or a newly developed transport protocol. However, for simplicity, let us limit the functions only to selecting between only existing transport protocols.

3.1 DDSC Channel

To enable switching between transport protocols, we introduced a Dual-Data Single-Control (DDSC) channel. Figure 2 illustrates the DDSC channel. It has one rate-controlled UDP, one TCP sub-channels, and one RTCP sub-channel. Both the micro and macro adaptation use the DDSC channel. In the micro adaptation, the channel intermittently switches the UDP sub-channel to the TCP sub-channel to probe for the available rate. In contrast, the macro adaptation decides the use of the sub-channel that provides better application-level performance.

4 Micro Adaptation

We use TCP-rate Probing Based Adaptation (TPBA) [1] as micro adaptation. Since the basic behavior of TPBA is described in [7], we herein describe an outline of TPBA and a modification to a scheme in [7].

In TPBA, like other CM applications, a CM flow uses RTP/UDP. However, it repeatedly switches its transport protocol to TCP in order to measure the **exact TCP-equivalent** rate. Thus, the rate probing period and the running period appear alternately. The ATA sender starts transmission with the rate probing period; data is transmitted over TCP. During the rate probing period, only normal data for the flow is transmitted over the TCP sub-channel and no data for probing is used; the action of probing does not generate an extra bandwidth that is irrelevant to data transmission. The transmission rate is examined based on the transition of the TCP sender's unacknowledged sequence number *snd_una*. When the transmission rate becomes stable, the ATA sender switches from TCP to UDP sub-channel. Once the ATA sender moves on to the running period, it transmits data over the UDP sub-channel at the measured rate during the rate probing period.

[1] In [7], we called this Rate Probing Based Adaptation (RPBA).

Originally in [7], we defined the behavior of TPBA sender in the running period as follows.

- The transmission rate is not increased during the running period. However, like LDA, if packet loss is reported from a TPBA receiver, the TPBA sender reduces its rate depending on the loss ratio.
- The running period lasts until predefined time expires.

These policies are undesirable for three main reasons. First, detecting congestion relying merely on information about packet loss is insufficient; when an RTP flow competes with a TCP flow on a link and traffic increases on that link, the RTP flow may not undergo a packet loss because in many cases the TCP flow is much quicker in reducing its rate. Thus the RTP flow may continue to use a bandwidth in an unfair manner after the TCP flow has reduced its rate. Second, in an error-prone wireless link, packets may be lost irrelevantly to congestion. Therefore, congestion control depending on packet loss ratio alone results in an erroneously low rate. Finally, it is unknown whether a rate is reduced in a TCP-friendly manner or not even when packet is lost due to congestion.

To overcome the problems above , we introduce path status analysis. The ATA receiver determines the path status depending on a change in delay as well as packet loss. The analyzed path status is transfered to an ATA sender periodically over RTCP RR messages. The ATA sender can switch to the rate probing period in reaction to notification of congestion included in the RTCP RR message.

Prior to discussing how to determine the path status, we define how to deal with delay. Since a clock at a sender and one at a receiver has a skew, we eliminate the skew with Estimation of Skew with Reduced Sample (ESRS) [9]. The variation in deley after the removal of skew is used to determine the path status.

4.1 Path Status

Now we classify the path status into five types. When delay remains stable in the context of the above approximation and there is no packet loss, the path status is stable. Let PATH_STAT_STABLE denote this status. When delay increases, it is suggested that the CM flow has entered to congestion irrespective of packet loss. Let PATH_STAT_INIT_CONGESTED denote this status. In contrast, when delay decreases, another flow has been terminated or disconnected. This status is referred to as PATH_STAT_INIT_LOOSE. When packet loss takes place in spite of stable delay, the CM flow either traverses a link of narrower bandwidth than it can be accommodated (PATH_STAT_SHRINKED) or one with random packet loss (PATH_STAT_RANDOM). We can distinguish between PATH_STAT_SHRINKED and PATH_STAT_RANDOM. When the flow travels through a narrow bottleneck link, packet loss does not occur until

```
switch( delay ) {
  case "decreased":path status =
        PATH_STAT_INIT_LOOSE;
                  break;
  case "increased": path status =
            PATH_STAT_INIT_CONGESTED;
                  break;
  case "stable": {
   if (packet loss == 0) path status = PATH_STAT_STABLE;
    else path status is either PATH_STAT_SHRINKED or
    PATH_STAT_RANDOM;
                  }
                  break;
      }
```

Figure 3: Classifying the path status

the buffer at the bottleneck link is full. Until then, delay remains constant. But after the buffer is filled, packet is constantly dropped at the buffer. In contrast, when the flow travels a lossy link, there is no such sudden point of change. Thus we can distinguish between the two, but the former hardly occurs when TPBA is used. A rate is already ensured to be below the bottleneck bandwidth after rate probing. Figure 3 shows a pseudo code for classifying the path status.

It is possible to raise a rate by conducting rate probing after receiving PATH_STAT_INIT_LOOSE. However, we decided not to do so because a rate can be increased in the Rate Probing period.

5 Macro Adaptation

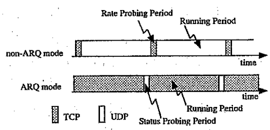

Figure 4: non-ARQ and ARQ modes

We defined the objective of macro adaptation as maximizing the application-level performance and suggested using protocol switching. Assuming that the rate available at the transport level, which is referred to as "raw rate," is regulated by the micro adaptation, we limit the function of the macro adaptation to selecting between UDP and TCP to raise the rate at the application level, which is referred to as "ADU rate" hereafter. Unless specified otherwise, a rate means a raw rate and ADU rate will be described as "ADU rate."

Let us consider what is needed to determine the selection between UDP and TCP. The difference between UDP and TCP is whether or not error control is performed by retransmission. Necessity of retransmission is closely related to the type of ADU and loss ratio. Even when the rate of UDP is the same as that of TCP, the ADU rate of UDP becomes worse if θ and loss ratio, η_p, are high. For instance, suppose θ and η_p are 20 and 0.05. In an extreme case when the distribution of packet loss has periodicity with the same period, every ADU has one corrupted packet and, as a result, the ADU rate becomes zero. At the other extreme, when packet loss occurs in a burst and is concentrated in a single ADU out of twenty ADUs, the ADU rate of UDP is smaller to that of TCP only by η_p. In reality, the ADU rate falls into these two extremes. Therefore, it is more important to identify the pattern or distribution of packet loss than to calculate η_p.

As described in the previous section, the path status is analyzed at the receiver of a CM flow. The receiver can notify the sender of more accurate information about the path status by reporting not only the loss ratio but the status such as PATH_STAT_INIT_CONGESTED and PATH_STAT_RANDOM. Then the sender can decide whether or not to alter the transmission rate. This is a large improvement from a simple RTCP RR message. In a conventional scheme with RTCP RR, if the receiver detects packet loss of some sort, it reports the sender loss ratio. We call this **"receiver's assistance to congestion control,"** which implies that the receiver enhances congestion control.

Together with this notification of the path status, the receiver can explicitly request for change of transport protocols. For instance, the following is a criterion of changing protocols.

- **When UDP is used:**
  When the path status is PATH_STAT_RANDOM, $\eta_p > \eta_{p0}$, and $\eta_a > \eta_{a0}$, switch to TCP.
- **When TCP is used:**
  When the path status has changed from PATH_STAT_RANDOM to PATH_STAT_STABLE, $\eta_p < \eta_{p1}$, or $\eta_a < \eta_{a1}$, switch to UDP.

Here, η_{p0}, η_{a0}, η_{p1}, and η_{a1} are threshold values. η_{p1} and η_{a1} are set higher than η_{p0} and η_{a0}, respectively, in order to avoid oscillation. From the viewpoint of macro adaptation, we define the following two modes:

- **Non-ARQ mode:**
  UDP is used as a transport protocol. However, intermittently, TCP is used to probe for a TCP-friendly rate. During the period with UDP, the receiver notifies the sender of the path status.
- **ARQ mode:**
  TCP is used as a transport protocol. However, intermittently, UDP is used to probe for the path status.

Due to the limitation in space, the complete behaviors of the ATA sender and receiver are described in [8].

6 Experiments

6.1 Wireless LAN

In Topology 1a shown in Figure 5 with Cond3, traffic was generated as follows.

Traffic Condition A

- Flows 1-3 begins transmission at time 0 and are terminated at time 80 s.
- Flows 1 and 2 are TCP flows generated at Host 2.
- Flow 3 carries CM ADUs with either ATA, LDA, or TCP.

Snapshots that show transition of the above metrics for ADUs which consist of 1440-byte packets are shown in Figures 7 - 9. In the figures raw rate and ADU rate

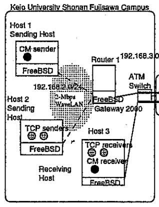

Figure 5: Path including a wireless link

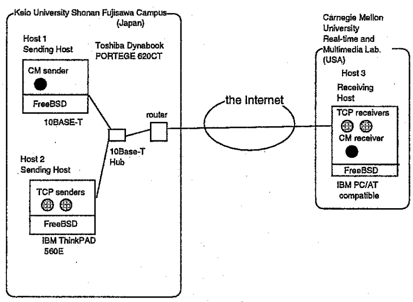

Figure 6: Experimental path over the Internet

represent a rate at the transport level and the application level, respectively. Time 0 in all these graphs corresponds to the time when the first packet of one of Flows 1-3 arrived. This case corresponds to a deteriorated wireless environment and packet losses occur frequently. Therefore all metrics of ATA, LDA, and TCP remain poor. As seen in these figures, LDA cannot raise a transmission rate due to frequent packet losses. It can be said that an algorithm in which a transmission rate is reduced merely depending on information about packet losses is not suitable for wireless networks. Let θ denote the number of packets per ADU. We changed θ from 1 to 10 and 20. As exhibited in the figures, when θ becomes high, ATA selects ARQ-mode, thereby attaining the same performance as that of TCP.

6.2 Internet

We tested our scheme over a path in the Internet 2 shown in Figure 6. Since we cannot control traffic in the Internet, it is difficult to compare between results with different schemes if they run in a different time. Therefore we compared ATA with TCP as follows.

Traffic Condition B

- Flows 1 and 2 begin transmission at time 0 and are terminated at time 80 s.
- Flows 1 carries motion JPEG data with TCP from Host 2.
- Flow 2 carries the Motion JPEG data with ATA from Host 1.

The measured rates and delay are shown in Figures 13 - 18. To evaluate the distribution of delay, we use a cumulative distribution function (CDF). As seen in the graphs, the TCP flow sometimes ceases probably due to a packet loss. The number of the deterioration in delay for the TCP flow varies on the trials; in Trial 3, there is only one jump in delay in 80-s period. In contrast, the ATA flow can maintain its stability in delay. At the same time it retains a TCP-equivalent rate inserting several times of rate probing.

7 Conclusion

In this paper, we proposed Adaptive Transport Architecture for continuous media communication using a DDSC channel. We designed and implemented ATA in FreeBSD. We compared ATA with Loss-Delay based Adjustment in experiments and showed that our scheme outperforms LDA in a wireless LAN and the Internet.

ATA has shown to be robust in a lossy wireless environment and is still applicable in the current Internet. Future work includes fine tuning of parameters for accurate classification of the path status.

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview," *RFC 1633,* June 1994.

[2] J. Mahdavi and S. Floyd, "TCP-friendly unicast rate-based flow control," http://www.psc.edu/networking/papers/tcp_friendly.html.

[3] J. Padhye, J. Kurose, D. Towsley, and R. Koodli, "A model based TCP-friendly rate control protocol," *Proc. of the 9th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video,* pp. 137-151, June 1999.

[4] C. Papadopoulos and G. M. Parulkar, "Retransmission-based error control for continuous media applications," *Proc. of the 6th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video,* pp. 5-12, 1996.

[5] R. Rejaie, M. Handley, and D. Estrin, "An end-to-end rate-based congestion control mechanism for realtime streams in the Internet," *Proc. of INFOCOM'99,* Mar. 1999.

[6] D. Sisalem and H. Schulzrinne, "The loss-delay based adjustment algorithm: a TCP-friendly adaptation scheme," *Proc. of the 8th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video,* pp. 215-226, July 1998.

[7] Y. Tobe, Y. Tamura, H. Nishino, and H. Tokuda, "Rate probing based adaptation at end hosts," *Proc. of IEEE Workshop on QoS Support for Real-Time Internet Applications,* pp. 58 - 65, June 1999.

[8] Y. Tobe, "A host architecture of QoS control for continuous media stream communications," *Ph.D. Thesis,* Graduate School of Media and Governance, Keio University, Dec. 1999 (http://www.mkg.sfc.keio.ac.jp/~tobe/thesis.ps.gz).

[9] Y. Tobe, H. Aida, Y. Tamura, and H. Tokuda, "Dectection of Change in Delay for Analyzing the Path Status," *Passive and Active Measurement Workshop,* Apr. 2000 (http://www.mkg.sfc.keio.ac.jp/~tobe/pam2000.ps).

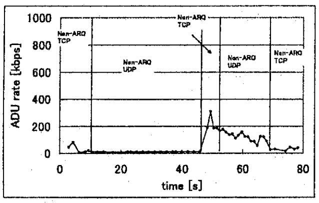

Figure 7: ADU rate of Flow 3 (ATA, Wireless, $\theta = 1$)

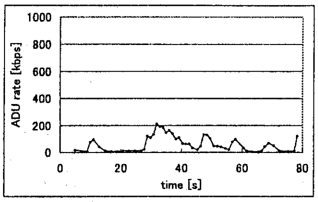

Figure 8: ADU rate of Flow 3 (LDA, Wireless, $\theta = 1$)

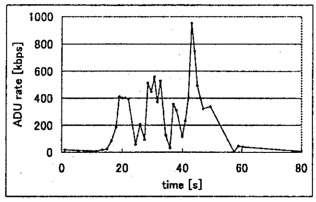

Figure 9: ADU rate of Flow 3 (TCP, Wireless, $\theta = 1$)

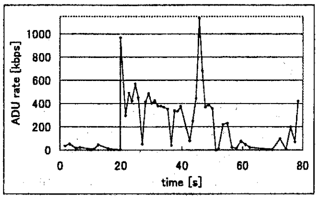

Figure 10: ADU rate of Flow 3 (ATA, Wireless, $\theta = 10$)

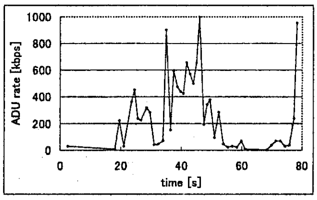

Figure 11: ADU rate of Flow 3 (ATA, Wireless, $\theta = 20$)

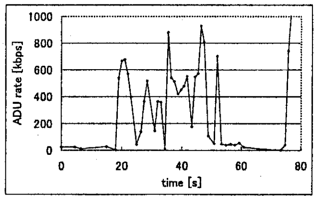

Figure 12: ADU rate of Flow 3 (TCP, Wireless, $\theta = 20$)

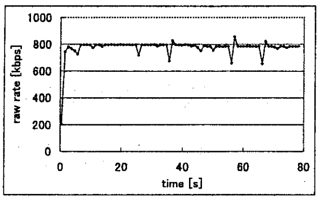

Figure 13: Raw rate of ATA Flow (Internet)

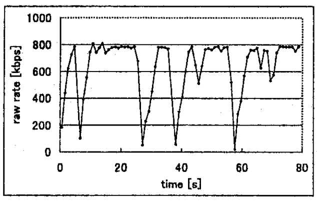

Figure 14: Raw rate of TCP Flow (Internet)

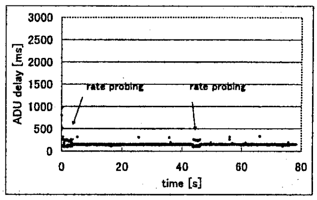

Figure 15: ADU delay of ATA Flow (Internet)

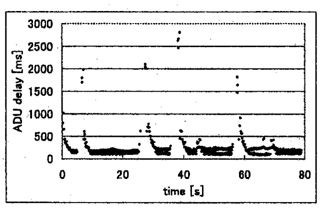

Figure 16: ADU delay of TCP Flow (Internet)

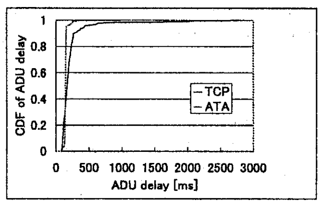

Figure 17: CDF of ADU delay (Internet)

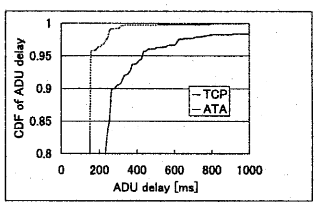

Figure 18: CDF of ADU delay, magnified (Internet)