International Journal of Reliability and Applications
Vol. 3, No. 2, pp. 99-112, 2002

Performance and Fairness Analyses of a STA/LTA based Internet Congestion Algorithm

Hwa-Sun Song
Department of Electrical Engineering
Kangwon National University, Chunchon 200-701, Korea

Sang-Yeol Joo
Department of Statistics
Kangwon National University, Chunchon 200-701, Korea

Young-Jun Chung*
Department of Computer Science
Kangwon National University, Chunchon 200-701, Korea

Abstract. Traffic congestion is one of critical factors in Internet applications to guarantee their QoS and provide reliable services. This paper discusses many existing congestion control algorithms and proposes a new ISDA. The algorithm is analyzed in respect of queue length, throughput and fairness. The proposed algorithm is working well with TCP and UDP traffics to offer QoS guarantee and fairness.

Key Words : *congestion control, QoS, TCP, UDP.*

1. INTRODUCTION

The early Internet has developed as a computer network, which mainly carries character data. Most applications at those days were best effort services that do not care the quality of service (QoS). However, The today's Internet, as a core infra data network, provides a multimedia service including characters, graphics, voices, and videos and so on. Internet should be an efficient and reliable data communication network satisfying QoS. In other word, Internet should offer a performance specification, as called as a service level agreement, for all applications which includes as transmission delay, jitter, and bandwidth, as well as reliability.

One of critical factors in Internet performance is a traffic congestion control. In general, congestion means, due to traffic concentration into a node, a network fails or cannot provide its own capabilities. Accordingly, a congestion control mechanism is required to provide multimedia services with QoS.

Internet services are classified into two types of traffics: TCP traffics that offer

* *E-mail address : ychung@kangwon.ac.kr*

reliable communications and UDP traffics that offer efficient communications. TCP typically includes a flow control and error control mechanisms to maintain reliability at the end-to-end communication. However, UDP has no such mechanisms. Thus, since reliable services satisfy their QoS, such heterogeneous and complex traffic natures make an Internet traffic control mechanism more complicated. In addition, reliable services should be provided with fairness.

This paper analyzes many proposed congestion control algorithms in terms of QoS and fairness and evaluates their performance. Section 2 discusses traffic flow control mechanisms and section 3 presents existing congestion controls and a new congestion control algorithm with using packet drop schemes. Section 4 presents a simulation model and evaluates the performance of the proposed control algorithms. Finally conclusion is discussed in Section 5.

2. INTERNET TRAFFIC FLOW CONTROLS

The TCP flow control in Internet uses a sliding window mechanism, which provides smooth traffic flows and reliable services between the end-to-end nodes. To maximize link utilization and network throughput, the window length generally sets as a maximum transport window value (called as a threshold value). A sender continuously transmits packets until the receiver window is full. After sending a packet, the sender decreases the transmitter window by the packet size in bytes. So, when the window is zero, wait until the window increases. The receiver reads packets form a data buffer and return acknowledges signals to the sender.

In Internet, packets generally arrive to their destination through many intermediate nodes. Some nodes may have serious traffic congestions. The congestions cause critical problems such as packet loss, node buffer overflow, and queue delay. Many flow control mechanisms have been proposed to improve such traffic control problems, one of which is a slow start flow control.

The slow-start flow control dynamically sets a maximum transmitter window length of a node buffer as a threshold to detect congestion and then marks the congestion notification bit for all flow-through traffic if the window exceeds the threshold. At the receiver side, all received packets are checked if the bit is set or not. If the bit set, congestion occurs at some nodes and the congestion reports to the sender. Then the sender performs some actions such as data rate deceasing [1,2,3].

The important factor of traffic control is early to detect congestion in the beginning stage, to notify to the sender and quickly to eliminate the congestion. A typical method for congestion notifications uses an indication bit. If a node is under congestion, the traffic passing through the node is marked in a congestion indication bit. The receiver checks the indication bit and recognizes the status of congestion.

Another method for congestion notification uses ICMP (Internet Control Message Protocol) messages [4]. In this scheme a congested node notifies the fact to senders by using an ICMP message. This scheme is simple to implement but generates an extra control packet for sender notification. Another different, called as ECN (Explicit Congestion Notification), has been proposed [5]. ECN uses an indication bit of IP packet for congestion notification. However, ECN gets worse in reliability view and

response time than ICMP.

3. CONGESTION CONTROLS FOR TCP AND UDP TRAFFICS

When heavy traffics flow through a node, the node memory buffer may be overflowed and some packets may be lost. This is called as traffic congestion. Such congestion may raise critical network problems. Thus a control mechanism is required to void from such critical congestion problems. In this section we will introduce some congestion algorithms.

3.1 Random Drop Algorithm

Random Drop Algorithm (RDA) is a simple congestion control algorithm [6]. RDA monitors incoming traffics at a node and checks a queue status of the node. If the queue exceeds a threshold, the node identifies itself under congestion and starts dropping some incoming traffics. This algorithm is very simple to implement but provides poor throughput, because, assuming that congestion occurs and several TCP connections exist, the node transmit window and packet rate will decrease at the same time for all TCP connections.

3.2 RDA with Fixed Probability (RDA/FP)

As mentioned in the above section, RDA drops all incoming traffics when the buffer exceeds a threshold. This results in critical performance degradation in case of burst traffic applications. To improve such problems, a RDA/FP has been proposed, in which the incoming traffic is selectively eliminated with a probability [7,8,9]. In other words, if the node buffer exceeds a threshold of queue, the incoming packet is determined if it is dropped, or not, with a proper probability as shown in Figure 1. This algorithm is also simple but has difficulties in optimizing a packet drop probability to maximizing network throughput. However, as a network gets lager and more complex, the optimization of throughput is not easy. Also in case of UDP and TCP traffic being together, TCP performance may be seriously degraded in respect of throughput and fairness due to uncontrollable and unreliable characteristics of UDP.

```
for each packet arrival
    calculate average queue as QueAvg
    if (MinQue < QueAvg < MaxQue)
      mark the arriving packet with probability Pa
    else if (MaxQue   <   QueAvg)
      mark the arriving packet
If (congestion)
Drop the marked packet
```

Figure 1. RDA with Drop Probability

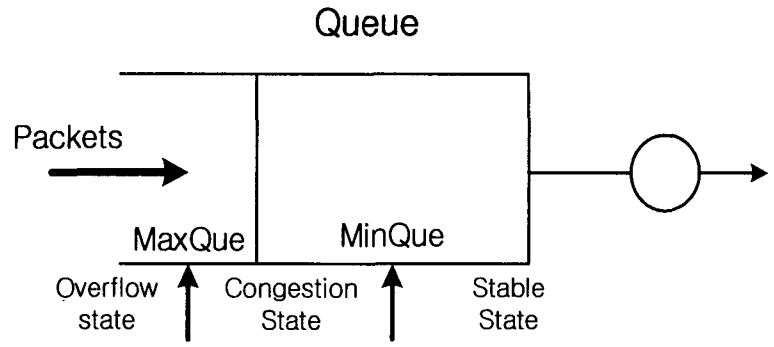

Figure 2. A Queue Structure of RDA/FP

3.3 RDA with Variable Probability (RDA/VP)

RDA/FP has a difficulty in maximizing link utilization alone with dynamic adaptation to network parameters. RDA with Variable Probability (RDA/VP) can improve such drawbacks. RDA/VP computes a packet drop probability in term of the network queue length. This algorithm defines two threshold values as Maximum threshold and Minimum threshold. As queue approaches to the minimum threshold, the drop probability decreases and, as it goes to the maximum threshold, the probability increases.

The packet drop probability (Pa) can be expressed as follows.

$$Pa = P0 \frac{QueAvg - MinQue}{MaxQue - MinQue} \quad (1)$$

Where P0 is the initial value of the drop probability, and MaxQue and MinQue are the Maximum value and the Minimum value of queue length, respectively. P0 can be computed from a TCP model [10] as follows.

$$P0 = \frac{1.225}{R * BW0.5} \quad (2)$$

Where R is the target round trip time between the end-to-end TCP connections and BW is the average transmission rate of a node. Since this algorithm can absorb the burst characteristics, the network throughput can clearly be improved. When QueAvg is close to MinQue (i.e. Pa approaches to 0), this algorithm accelerates to a severe congestion state. Also, when QueAvg is close to MaxQue (i.e. Pa approaches to 1), this algorithm drops most incoming traffics and so the throughput is quickly degraded. The results cannot guarantee QoS. Thus, a scheme for dynamically changing MaxQue and MinQue can be an important issue to optimize Pa. In addition, since a packet drop scheme is

applied for various types of traffic services, the fairness for all services is another topic to study.

3.4 Weighted Random Drop Algorithm (WRDA)

RDA/VP uses a single packet drop scheme, which does not consider all types of services. In other words, the scheme has some difficulties in controlling all congestions of Internet services. In a congestion situation, the high bandwidth message service is critically not interfered by some packet losses, but the low bandwidth message service results in a serious problem. Typically Internet has TCP and UDP services coexist. Traffic congestion makes the TCP transmitter window decreased and TCP throughput reduced. However, UDP traffics maintain its own bandwidth without congestion occurrence because UDP has no traffic flow control mechanisms. Thus congestion makes TCP services critically degraded and loses fairness for bandwidth.

3.5 Improved Selective Drop Algorithm (ISDA)

In general, TCP services cannot be guaranteed from congestion. Under congestion situation, the high bandwidth UDP services can survive but most low bandwidth UDP services is completely destroyed. An algorithm arises to assign the packet drop priority for each service connection and then to throw away only the high bandwidth UDP traffics [10, 11].

A traffic measurement for all service connections is a painful and time-consuming process to compute the bandwidth for all services and discriminate if they are high bandwidth traffics or not. A noble scheme, called Improved Selective Drop Algorithm (ISDA), is proposed for improving such a problem. This scheme monitors all traffics with a simple mechanism, called as the short-term average (STA) and the long term average (LTA), and then, if traffic congestion is detected, applies RDA for only high bandwidth traffics.

The average traffic rate is measured by counting packet bytes during a window interval. But since the measured value fluctuates at the beginning of traffic and at the end of traffic, the following filter makes the average value smooth.

$$BWi(n) = (1-\alpha) * BWi(n-1) + \alpha * Measured\_Bwi(n) \quad (3)$$

Where BWi(n) and Measured_BWi(n) are the average traffic rate and the measured traffic rate at the nth time interval for service i. α is a factor to moderate the average in range of 0 to 1. As α is in range of 0.5 to 1, BWi(n) is very sensitive to traffic changes, which is called as STAi(n). And as α is form 0 to 0.5, BWi(n) is slow to traffic changes, which is called as LTAi(n). The ratio of STAi(n) and LTAi(n) may have the history of traffic changes, which can minimize the mistakes of detection due to burst traffics. Thus our algorithm accurately discriminates the high bandwidth services with STAi(n) and the ratio of STAi(n)/LTAi(n), shown in Figure 3.

```
Measure BWi(n) for service i.
Compute STAi(n) and LTAi(n) as follows;
    STAi(n) = STAi(n-1) + α1 * measured_BWi(n); /*  0.5 < α1 < 1 */
    LTAi(n) = LTAi(n-1) + α2 * measured_BWi(n); /*  0   < α2 < 0.5 */
Compute the ratio of STAi(n) and LTAi(n)
If (STAi(n) > High bandwidth threshold) then
        If (The ration > Threshold of burst factor) then
Apply RDA with packet drop probability
Return
```

Figure 3. An operation procedure of ISDA

4. PERFORMANCE EVALUATION

4.1 Simulation Model

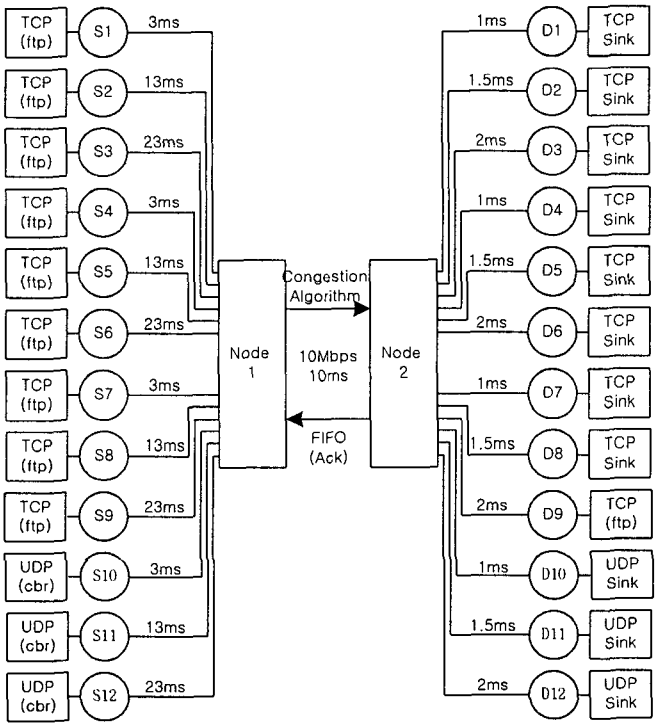

Figure 4. A Simulation Model

In a congestion situation, QoS control and fairness are very important factors in performance between UDP and TCP traffics. With considerations of such factors, a multimedia network that contains a CBR type of UDP traffics and a burst type of TCP traffics is selected as a simulation model. This model is implemented by a real-time network simulation tool, NS-2, which was developed by DARPA [12].

The simulation model consists of 12 sources (Si, i=1, 12), two nodes (node1, node 2), and 12 destinations (Di, i=1, 12). Sources S1 thru S9 are defined as TCP traffics and Sources S10 thru S12 are defined as UDP traffics. The TCP sources have flow control mechanisms and their transmission rate is assumed as the maximum 100 Mbps. The transmission distances between TCP sources and node 1 are recursively defined as 3, 13, and 23 ms. UDP sources are assumed as multimedia VBR traffics such as video and voice. For simplicity of simulation, the UDP sources are defined as CBR traffics of 1, 2, and 3 Mbps, which are 10%, 20%, and 30% of the link rate, respectively. The link rate between node 1 and node 2 is 10Mbps and its distance is defined as 10 ms. The forward data flow at node 1 uses a congestion control and the backward data flow uses a flow control with FIFO queue, which returns acknowledge messages and congestion messages. Finally data flows between sources and destinations are defined as connections from Si to Di for i = 1, 12. Also the max queue length of node 1 and node 2 have are assumed as 50. The thresholds of MaxQue and MinQue for congestion control are 25 and 10, respectively. Other parameters uses default values of NS.

4.2 Simulation Results

1) Node Queue

The first simulation uses TCP, UDP, and TCP+UDP traffics to consider the characteristics of congestion control algorithms mentioned in the previous sections. The simulation results are depicted in Figure 5 thru Figure 10.

Figure 5 presents the queue length of RDA/VP at node 1 for TCP traffics. TCP sources go to 100 Mbps at the beginning and, after 8 seconds, are in the stable state with the maximum queue length of 11. Also the bandwidth for each TCP connection is allocated with fairness. Figure 6 presents the queue length of ISDA at node 1 for TCP traffics. The maximum queue length at the transient state and at the steady state is very similar to that of RDA/VP. However, in the beginning, the queue length is little fluctuating due to detection of the high-bandwidth traffic flows. Figure 7 shows the queue length of RDA/VP at node 1 for UDP traffics. Since sources S10, S11, and S12 are defined as 10%, 20%, and 30% of the link respectively, the total 5 Mbps CBR traffic are used. The system is at the transient state and, after 8 seconds, goes to the queue length of 24 in the steady state. Figure 8 shows the queue length of ISDA at node 1 for UDP traffics. This case is similar to that of RDA/VP at the beginning and goes to the queue length of 24. But after 10 seconds, the detection of the high bandwidth traffics works and works the pre-filtering mechanism as a part of congestion control, which drops packets of the high bandwidth connections.

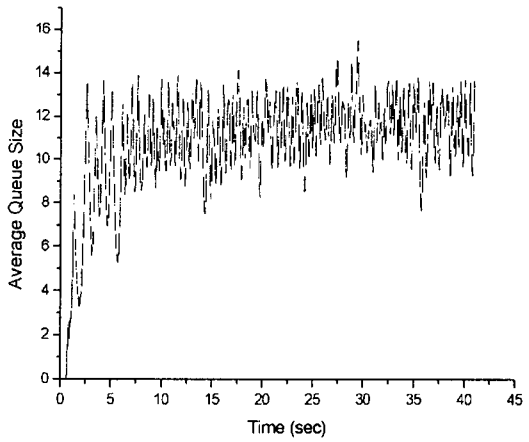

Figure 5. The queue length of RDA/VP at Node 1 for TCP traffics (MaxQue=25, MinQue=10)

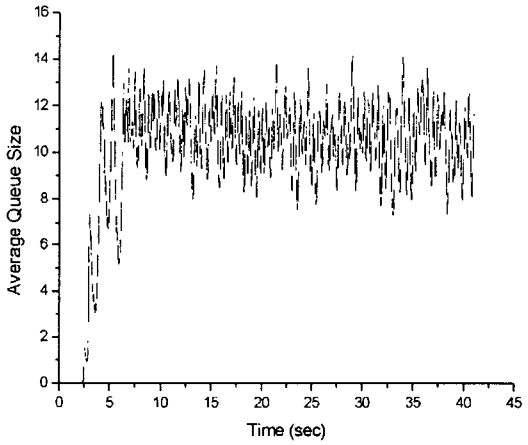

Figure 6. The queue length of ISDA at Node 1 for TCP traffics (MaxQue=25, MinQue=10, $\alpha 1$=0.8, $\alpha 2$=0.3, STA/LTA=3)

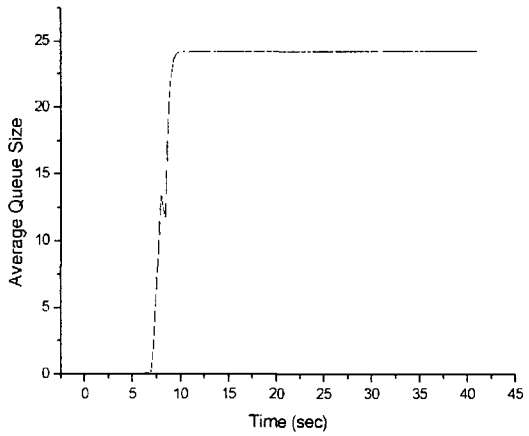

Figure 7. The queue length of RDA/VP at node 1 for UDP traffics (MaxQue=25, MinQue=10)

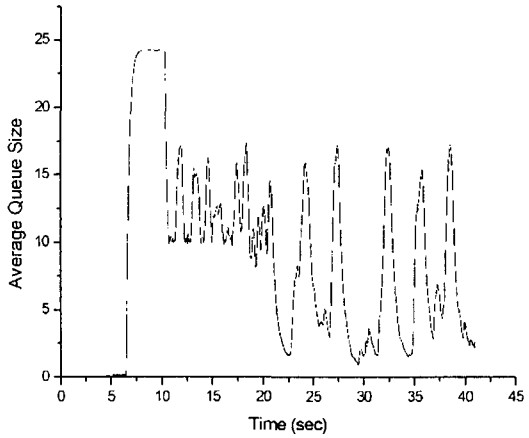

Figure 8. The queue length of ISDA at node 1 for UDP traffics (MaxQue=25, MinQue=10, $\alpha 1$=0.8, $\alpha 2$=0.3, STA/LTA=3)

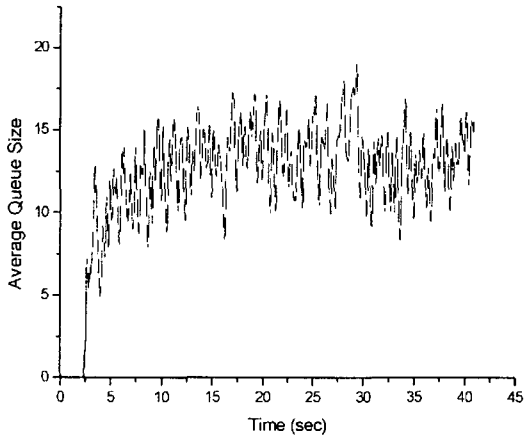

Figure 9. The queue length of RDA/VP at node 1 for TCP + UDP traffics (MaxQue=25, MinQue=10)

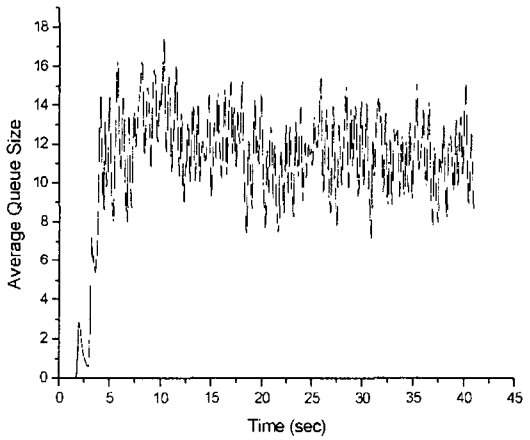

Figure 10. The queue length of ISDA at node 1 for TCP + UDP traffics (MaxQue=25, MinQue=10, $\alpha 1$=0.8, $\alpha 2$=0.3, STA/LTA=3)

Figure 9 presents the queue length of RDA/VP at Node 1 for TCP + UDP traffics. After the transient state of 8 seconds, the system is stable at the threshold of 11. In the steady state, UDP traffics use the initially allocated transmission rates and TCP traffics equally share the rest of the bandwidth. Figure 10 depicts the queue length of ISDA at Node 1 for TCP + UDP traffics. Similarly in RDA/VP, The system is stable at the threshold of 11 after the transient state of 8 seconds. Some fluctuation of traffics is caused by the pre-filtered high bandwidth traffic.

2) Throughput and Fairness

This section presents the throughput of UDP + TCP traffics for some connections under a congestion condition. Figure 11 measures the traffic rate of each connection (Si to Di, i=1, 12) at node 1 during 400 seconds by using RDA/VP. Since UDP has no flow control mechanism, the UDP services approach to 1, 2, and 3Mbps, which are initially allocated. TCP traffic initially starts with maximum rates but their rates soon goes down by flow control mechanisms. In the steady state, TCP traffics use only the remainder of bandwidth resources, which is not used by UDP services. Thus TCP rates oscillate around 0.5Mbps.

RDA/VP has no priority control for dropping packets and so performs the same packet drop policy to all traffics. So UDP maintain the initial rate even in spite of packet dropping. TCP reduces its rate by a flow control and goes to the steady state. Consequently, the rate of TCP traffics goes down and its performance is degraded quickly. Also, since low bandwidth of traffics is dropped with equal probability, some applications with low bandwidth lose all their capabilities.

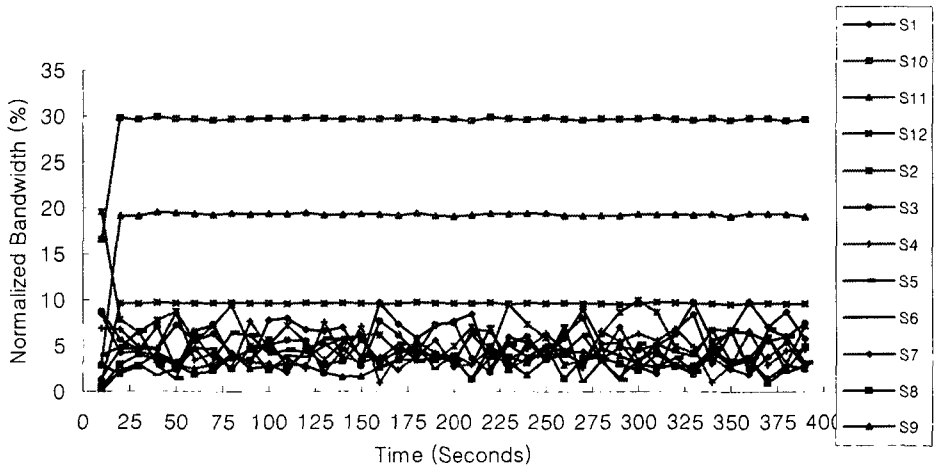

Figure 11. The throughput for each connection when using RDA/VP. (MaxQue=25, MinQue=10)

Figure 12 presents the throughput for each connection when using ISDA. The scheme measures the traffic rate of each connection (Si to Di, i=1, 12) at node 1 during 400 seconds. Collected data determines the high bandwidth traffic with STA and LTA. Here

since STA is sensitive to the traffic changes, it can detect the high bandwidth connections. But when some burst traffics are incoming, fault-detection can be made. LTA can reduce the possibility of such fault-recognitions. Since LTA is the long-term average, it has a history data of the high bandwidth. In other words, the ration of LTA/STA is a characteristic of traffic burst. As LAT/STA goes to "1", traffic burst will be weak and as LAT/STA goes to "0", traffic burst will be serious. Thus ISDA filtering the high bandwidth UDP traffic and mark them for packet dropping. The remaining bandwidth after the high bandwidth packet drops is equally shared by TCP services and low bandwidth services.

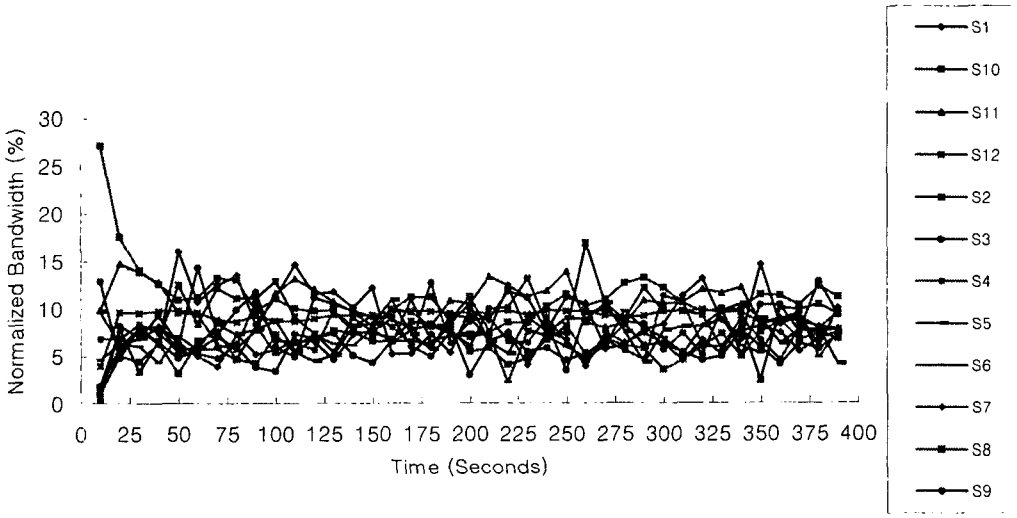

Figure 12. The throughput for each connection when using ISDA. (MaxQue=25, MinQue=10, α1=0.8, α2=0.3, STA/LTA=3)

In Figure 12, UDP traffics set as initial bandwidths. When congestion starts, high bandwidth UDP packet drops increase. The throughput of high bandwidth traffics decreases but the throughput of the low bandwidth UDP traffics and TCP traffics maintain consistent values. In our simulation, UDP traffic are initially set as 1, 2, 3Mbps, they oscillatory converge to 0.9Mbps in the steady state.

3) QoS (Quality of Service)

In Figure 13, the average throughput for each source connection is depicted with using RDA, RDA/VP, SDA and ISDA. RDA and RAD/VP schemes cannot guarantee the bandwidth of TCP traffics (I.e. QoS). In special, congestion occurrence makes TCP flows severely reduced and TCP traffics cannot maintain QoS. However, SDA and ISDA can equally allocate the bandwidth for each service and guarantee their QoS. However In SDA, many packet drops of UDP traffics cause some difficulties in guarantee the high bandwidth traffics. As one of solutions for such problems, ISDA uses a scheme that guarantees TCP and low bandwidth UDP traffics.

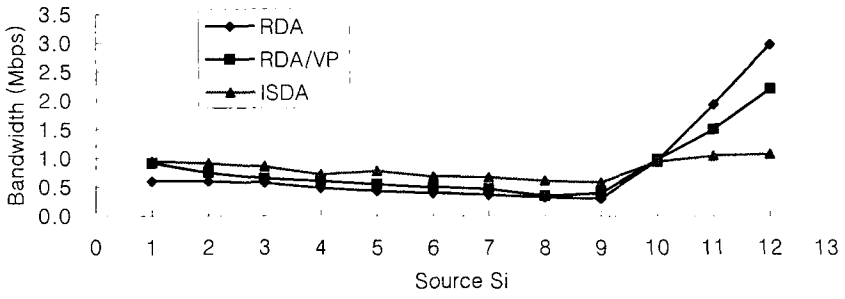

Figure 13. Average throughput for congestion control algorithms (MaxQue=25, MinQue=10, α1=0.8, α2=0.3, STA/LTA=3)

5. CONCLUSIONS

Traffic congestion is one of critical factors in Internet applications to guarantee their QoS and provide reliable services. This paper discusses many existing congestion control algorithms and proposes a new ISDA. The algorithm is analyzed in respect of queue length, throughput and fairness. Also TCP services and UDP services are characterized under congestion. In special, the change of TCP traffics is analyzed in detail with respect of offensive UDP traffics. In typical, if UDP traffics maintain their bandwidth, TCP traffics cannot guarantee their QoS and fairness. If fairness is focused, some UDP traffics have severe damages of their services. Such problems are greatly improved by ISDA. Consequently our ISDA offers fairness for all traffics with properly guaranteeing QoS.

REFERENCES

[1] Jain, R., Ramakrishnan, K.K (1988). Congestion Avoidance in Computer Networks with a onnectionless Network Layer: Concepts, Goals, and Methodology, *proc. IEEE Comp. Networking Symp.*, Washington, D.C., April, 134-143.

[2] http://www.dcifr.hinet.net/html/terms/FECN.html.

[3] http://www.dcifr.hinet.net/html/terms/BECN.html.

[4] Prue, W., and Postel, J. (1987). Something a Host Could Do with Source Quench, *RFC 1016*, July.

[5] Ramakrishnan, K. K., Floyd, S., and Black, D. (2001). The Addition of Explicit Congestion Notification (ECN) to IP, *RFC 3168*, Proposed Std., Sept.

[6] Sally Floyd, Kevin Fall, and Kinh Tieu (1998). Estimating Arrival Rates from the RED Packet Drop History, April, http://www.aciri.org/floyd/end2end-paper.html.

[7] Ratual Mahajan (2001). RED-PD: Controlling High Bandwidth Flows at the Congested Router, April, http://www.cs.washington.edu/homes/ratul.

[8] Sally Floyd and V Jacobson (1993). Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking*, **1**(4), 397-413, August.

[9] Sally Floyd (1997). RED: Discussions of Setting Parameters, http://www.aciri.org/floyd/REDparameters.txt, November.

[10] Sally Floyd (1999). Kevin Fall Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, 7(4), 458-473, August.

[11] Keshav, S. (1991). A control-Theoretic Approach to Flow Control, *Proceedings of SIGCOMM '91*, September, 3-16.

[12] NS Web Page: http://www.isi.edu/nsnam.