Simulation of TCP Traffic over ATM Networks

Dr. Mohammed Najim Abdullah
Computer Engineering Department
University of Technology
mustafamna@yahoo.com

Ahmed Saleam Abbas
Computer department
Karbala University
ahmed_saleam@yahoo.com

Abstract

There are a number of techniques to adapt TCP/IP to ATM network environments because of high speed communication by ATM and the reliability of TCP/IP. Serious performance degradations occur when TCP/IP adapted to ATM, especially when the utilization of network resources is high. The previous techniques that attempt to patch up TCP\IP over ATM are depended on switch-level enhancements so they have limited success in alleviating performance degradations problem. In this work, a proposed algorithm is designed that operate at AAL5 layer, that algorithm is named as Adding Identification number and Matching (AIDM) algorithm. The resulted transport protocol from using AIDM algorithm is named as Improved TCP Reno. AIDM algorithm is presented and a simulation result is compared with other adaptation of TCP\IP over ATM. Visual Basic (VB6) programming language is used for coding TCP traffic over ATM simulator. This simulator illustrate how AIDM algorithm operate, how AIDM used to enhance network performance and how it reduced the period of time required to recover the lost cells in comparison with TCP Reno.

Keywords: Computer Networks, ATM Networks, ATM Traffic Control, TCP\IP protocol.

الخلاصة

بوجود عدد من التقنيات الخاصة بتبني شبكات TCP/IP في بيئة شبكات الـ ATM والسبب في ذلك يعود الى السرعة العالية التي يتميز بها الـ ATM بالاضافة الى الموثوقية التي يتميز بها الـTCP/IP. الدراسات السابقة بينت انَّ الـ TCP/IP عندما يعمل مع ATM يقود الى أنخفاض واضح في الاداء وخاصة في الشبكات التي تمتاز باستخدام عالي لمصادرها, التقنيات المقترحة مثل تحسينات مفتاح المستوي (Switch-Level Enhancements) امتلكت نجاحاً محدوداً في تخفيف المشكلة. انَّ الجوهر الخاص بالـ TCP المقتَرح هو خوارزمية الاضافة لارقام التعريف والمطابقة (AIDM) و هيَ عملية استرجاع كفوئة حيث تجعل الاداء الخاص بالـ TCP/IP اقل حساسية لفقدان الخلايا. في هذا البحث تم تقديم خوارزمية الـ AIDM ونظرة عامه عن الخصائص الاساسية لـ TCP المحسن. وقدمت نتائج المحاكاة التي تبين التفوق الذي يتميز به البروتوكول المقترح عندما يقارن مع التبنيات الاخرى الخاصه بـTCP/IP بإستخدام الـ ATM. تـم استخدام لغة البرمجة(VB6) في كتابة برنامج الـ (TCP traffic over ATM simulator) الذي يوضح كيفية عمل الخوارزمية المُقترحة و كيفية استخدامها في تحسين اداء الشبكة وتقليل الفترة الزمنيه المطلوبة لاسترجاع الخلايا المفقودة بالمقارنة مع البروتوكول السابق وهو (TCP Reno).

1- Introduction

In this paper a new algorithm is designed and named as "Adding Identification number and Matching" (AIDM) algorithm. This algorithm makes a minor modification on AAL5 layer, where AAL5 layer operates as an interface between IP layer and ATM layer. The transport protocol that uses AIDM algorithm named as Improved TCP Reno. That proposed TCP is also discussed in this paper.

There are some related works as in 1998, Charalambous P. Charalambos and et al they studied the effect of various noisy Bandwidth-Delay-Product (BDP) connections on TCP performance and they tested three transport protocol implementations: TCP Reno, TCP New Reno, and TCP SACK. Their results show that in cases where noisy BDP networks and high speed satellite links are established, throughput obtained by TCP Reno, TCP New Reno, and TCP SACK end systems is very low (about only 1% of the optimal throughput), showing that none of the tested TCP protocols are able to recover under these conditions. In such cases, TCP SACK delivers 36.5% performance improvement over TCP Reno, and TCP New Reno delivers 6.4% performance improvement over TCP Reno. As the channel BER drops, the performance improvement, throughput, and channel utilization of TCP SACK and TCP New Reno over TCP Reno increases. In all cases, TCP SACK is more efficient than TCP New Reno and TCP Reno, because of its ability to recover from multiple losses faster than the other two protocols under test; they use test bed to extract result [1].

In 2003, Azer Bestavros and Gitae Kim, presented TCP Boston, a novel, fragmentation-tolerant TCP protocol, especially suited for ATM network environments. TCP Boston integrates a standard TCP/IP protocol (such as Reno or Vegas) with a powerful encoding mechanism based on AIDA. They have presented their implementation of TCP Boston and have shown its performance superiority when compared to TCP techniques that are more vulnerable to fragmentation, namely TCP Reno and TCP Reno with EPD switch-level enhancements. Their performance evaluation was done in two steps. They first provided them with basic simulation results for a congested ATM switch by use the LBNL Network Simulator (ns), whereas the second allowed them to analytically predict the wasted bandwidth for TCP Boston, Reno, and Reno with EPD in a multi-hop network [2].

From the given survey one can notice that no works has been done on improving the efficiency of TCP over ATM through minor modifications on the AAL5 work, so, this paper will work on that side, where a transport protocol is suggested, named as Improved TCP Reno. At the core of that proposed protocol is "Adding Identification Number and Matching" (AIDM) algorithm. AIDM try to make TCP/IP performance less sensitive to cell losses through its ability to use partial delivered packet and ask sender to retransmit lost cells only, thus enhancing the performance of TCP in ATM..

2- AIDM algorithm

This algorithm is designed to reduce effects of cell loses on the TCP performance. This algorithm work at the AAL5 layer and its essential job is to reduce side effect of fragmentation. AIDM is a technique that adds minimal redundancy to each cell. To explain how AIDM works, suppose there is a packet P of a data object has to be transmitted, and P consist of M cells. By using AIDM algorithm packet P is divided into M fixed size cells and ID number is added for each cell. The resulted N cells are stored at sender's buffer to be used for recovering purposes if any cell lost through transmission process

Now resulted cells are ready for transmission through ATM network. At receiver side, the initial/final ID number is used to create matching pattern named as Actual Pattern (AP). And another pattern is made from ID number that extracted from the received cells; this matching pattern is named as Extracted Pattern (EP). If any cells lost the receiver will discover that by matching the AP pattern with EP. After matching the lost cells are discovered then ask sender to retransmit those lost cells only. When the sender receive negative acknowledge with ID numbers of lost cells, it looks for these cells in its buffer to retransmit them to the destination. Figure (1) shows the flowchart of division and reconstruction operations of AIDM algorithm.

3- AIDM Characteristics

The fragmentation problem occurs when a cell is lost en route, So the receiver could not perform the reconstruction process for the packet to which that cell belonged, unless the lost cell belong to that packet is retransmitted [2], In this work AIDM algorithm is designed to overcome this problem, so AIDM has the advantage of being effective in terms of its use of available bandwidth. In comparison with recent protocol such as TCP Reno the receiver do nothing just discards the received cells belong to that packet and wait for the sender to automatically retransmit all cell of that packet, so recent approach is not effective in term of its use of available bandwidth especially in multi-hop networks, but it also has the advantage of being quite simple to implement since it doesn't require additional functionality at the sender and receiver ends.

To explain how AIDM is incorporated in a TCP protocol, let consider the following scenario, where:

M: Number of cells that belong to the packet that wants to be transmitted before the fragmentation of that packet.

N: Number of cells that belong to the packet that wants to be transmitted after fragmentation and adding identification number to each cell of them.

R: Number of cells that received by the receiver and segmented together to recover the original transmitted packet

At the beginning of the assumed scenario, the sender divided the M-cell packet into N cells, and then adds ID number to each cell of them. The resulted N cells are stored at the sender's buffer for recovering purpose. The packet of that N-cell is transmitted to the receiver. Now, assume that the receiver gets R of these cells.

If R = N, then the receiver do nothing just remove ID numbers from received cells and reconstruct original packet, and then acknowledge that it has completely received that packet by informing sender that it needs no more cells from that packet. The sender will respond to that positive acknowledge by clearing its buffer and work on next packet.

If R < N, that mean (R – N) cells were lost, so the receiver acknowledge that it has partially received that packet by informing sender that there is needing to (N – R) more cells from the original packet and those cells have the following ID numbers, then it send ID numbers of the missed cells. To that negative acknowledge, the sender will respond by looking for missed (N – R) cells in its buffer, and then retransmitting these cells. The process continues until the receiver receives enough cells to be able to reconstruct the original packet.

4- Modeling of the Improved TCP Reno

The improved TCP Reno is the transport protocol that combines the reliability of TCP/IP with the high speed and flexibility of ATM.

In addition, that improved protocol apply AIDM algorithm in its AAL5 layer, so it have the ability to overcome fragmentation problem and to produce a high performance gain especially when it is deployed over ATM networks.

The main functions included in the improved TCP Reno are; session management, packet management, and flow control and transmission. These functions are given below:

A. Session Management: The improved TCP Reno protocol as other recent protocols manages this session in three phases: Connection establishment phase, Data transfer phase, and Termination phase. The roles that played by these phases is similar to that of correspond phases that applied in other implementation of TCP over ATM [2], except that information specific to AIDM which are required by the receiver for reconstruction purposes are transmitted to the receiver at the connection establishment phase. That needed information concluded the number of cells in a packet (N) and initial/final ID number that added to these cells.

B. Packet Management: it has four main processes, they are :-

a. Packet dividing: Let there is a data block (packet) of size (P) bytes, the improved protocol divides that data block into (N) cells of size (C). So the number of cells (N) = size of one packet (P) / size of one cell (C), where the size of one cell is 53 bytes as standard size.

b. **ID number addin**g: when division process completed the ID number is added for each cell at the sender. Where ID number is one byte used for identification purposes to recognize lost cell. These resulted cells are stored at the sender's buffer to be used if any cell lost through transmission process.

c. **ID number extracting and matching:** when all cells which belong to transmitted packet are reaching to the destination and those received cells (R) are less than the actual number of cells in the transmitted packet (N), i.e. if R < N then ID numbers are extracted from the received cells to make pattern named as Extracted Pattern (EP). Also another pattern is made by using initial/final ID number named as Actual Pattern (AP). Where initial/final ID number is a part of an agreement that made at the session management phase. Now matching is made between (EP) and (AP) to determine the lost cells. Where that is the goal from this process.

d. **Packet reconstruction:** This process is performed when R = N, or when R < N but the lost (N – R) cells are retransmitted successfully. The main objective of this process is grouping received cells together to retrieve original packet and passing that packet to the higher IP layer.

- **Flow Control and Transmission:** Flow control determines the dynamics of packet flow in the network, which in turn affects the end-to-end performance of the system.

C. When an ACK arrives, the sender will check that ACK to see if it means that the reconstruction process of the packet is completed successfully or not, if it does the sender will clear its buffer to be ready for next packet transmission. If it doesn't, the sender extracts from the ACK the ID number of lost cells, then looking for these cells in its buffer and retransmission them at the next retransmission time.

The main objective of this work is to reduce side effect of cell loosing, and turns fragmentation into an advantage for TCP/IP, thus enhancing the performance of TCP in general and its performance in ATM environments in particular. In figure (2) and figure (3) the flowcharts of sender and receiver are illustrated, respectively, where they apply improved TCP Reno.

5- TCP Reno

In this section TCP Reno is discussed briefly to make the base for the comparison that will be made in next section with improved TCP Reno. TCP Reno is the most widespread variant of TCP currently. It is derived from the original version of TCP (TCP Tahoe). The TCP stack in Solaris, Linux and Windows is a variant of TCP Reno. TCP Reno uses two key techniques for congestion management. One is slow-start and the other is congestion avoidance [22].

- **Slow Start: -** Slow start is the mechanism which is used when a TCP session is started. Instead of starting transmission at a fixed rate, the transmission is started at the lowest rate possible and doubled.

The rate of doubling is tied to the rate at which acknowledgments come back; thus, the rate

- **Congestion avoidance: -** The congestion avoidance algorithm starts where slow start stops. The TCP increases its transmission rate linearly, by adding one additional packet to its window each transmission time. If congestion is detected at any point, the TCP reduces its transmission rate to half the previous value. Thus the TCP seesaws its way to the right transmission rate. Clearly, the scheme is less aggressive than the slow-start phase (note the linear increase against the exponential growth in slow start) [22].

6- Implementation, Results and Simulations

TCP traffic over ATM simulator is designed and implemented by using VB6 programming language. That simulator has ability to apply TCP Reno and Improved TCP Reno to perform conventional file transfer. Also that simulator can be used to compare TCP Reno and Improved TCP Reno in throughput, speed, and their sensitivity for cell losses in the same environment and for the same file with the same number of cells lost, where these cells are distributed between determined numbers of packet. Now, two cases are taken to get result and to discuss them, they are:-

A. Case 1:-

In this case, file of size 59277 bytes is chosen to be transferred from client to server. Eight lost is ranged between the following values (0, 10, 20, 40, 65, 85, 105, and 150) and all that distortions happened in one packet. Chart (1) shows how throughput changed when cells lost increased in one packet. Chart (2) shows how required time for transmission changed when cells lost increased in one packet. Chart (3) shows how enhancement rate changed when cells lost increased in one packet.

B. Case 2:-

In this case, the same file that selected in case 1 is chosen with the same value of cells destroyed, but these destroyed cells are distributed between two packets. Chart (4), chart (5), and chart (6) showed the throughput, is doubled every round trip time.

If congestion is detected, the transmission rate is reduced to half of what it is currently and the congestion avoidance process starts. The rate doubling and reduction is nothing but a binary search for the 'right' transmission bandwidth [22].

- A TCP is designed to drive a network into saturation. A TCP will never reduce its transmission rate unless it sees a packet drop. This will only occur when the network is overloaded. Thus a combination of TCPs will together ensure that a network is driven into overload constantly. This makes it difficult for new connections to enter the system, since they are at a natural disadvantage due to slow start. TCP Reno attempts to equalize the window size for competing connections.

samples are taken, where the number of cell

However, a measure of the bandwidth achieved is the bandwidth delay product, not the window size only. Thus, TCP Reno is unfair to connections with larger round-trip times [22].

At TCP Reno over ATM network, the AAL5 layer do its job as an interface between IP layer and ATM layer through perform the Segmentation and reassembly of IP datagram (packet). Segmentation occurs at the sender, where packets are divided into M cells and pass them to lower ATM layer to transmit them to the destination. Reassembly occurs at the receiver to reconstruct packet by grouping received cells together and if any cell lost this packet will be discarded and ask sender to retransmit that packet [11].

In figure (4) and figure (5) the flowcharts of sender and receiver are illustrated, respectively, where they apply TCP Reno.

7- Result discussion:-

When Improved TCP Reno is applied the throughput decreases as the number of cells lost increases, but throughput sensitivity for cell loses is lower than its sensitivity when TCP Reno is applied as illustrated in chart (1) and chart (4). Also, when no cell lost the throughput of TCP Reno is better than that of Improved TCP Reno, the reason of that belongs to the ID number that added for each cell, where these numbers are not used because of there is no error. This case is regarded as ideal case.

The incorporation of AIDM algorithm into TCP/IP over ATMs protocol requires only the IP and ATM layers, where the ATM adaptation layer of type five (AAL5) represent that interface. So when AIDM algorithm is used, there is no bandwidth is wasted as a result of packet retransmission or partial packet delivery, and there is no modification is necessary to the switch-level protocols.

When Improved TCP Reno is applied the required time to transfer the selected file with determined cell loses is lower than what is required when TCP Reno is applied, and that time is increases as number of lost cell increases, as illustrated in chart (2) and chart (5).

Chart (3) and chart (6) illustrate the enhancement rate of throughput and time, where that enhancement is made by Improved TCP Reno. Also, there is one disadvantage that happened if and only if when no cell lost through the transmission process, where in this situation the processing time that consumed to achieve AIDM algorithm is unusual, so that the lead to enhancement rate is -2 %,

Chart (7) illustrates that the enhancement rate will be increased when the number of packet that have problem increased. So the enhancement rates reach to 38 % when lost cells distributed between four packets. The average of enhancement rate for all these cases equal approximately to 20%. So that is regarded as good percent to improve the recent TCP Reno protocol.

additional functionality at the interface between

8- Conclusion:-

From this work the following is concluded:-

The Improved TCP Reno is better than TCP Reno in throughput and time, where the average of this enhancement is about 20 %, the reason of this enhancement belong to its ability to recover lost cells in short interval of time

- The transmitted cells that belong to the same packet are useful in spite of the lost cells. This is due to the ability of AIDM algorithm to determine lost cells and recover them by asking sender to retransmit these cells only.
- AIDM algorithm makes Improved TCP Reno throughput less sensitive to cells loses than resent TCP Reno protocol.
- The Improved TCP Reno turn's ATM 53-bytes cell oriented switching architecture into an advantage for TCP/IP. In another word, The Improved TCP Reno overcome fragmentation problem.
- The Improved TCP Reno has one disadvantage which can be noted when no cell is lost through transmission process. So its throughput in this case is lower than that of recent TCP Reno by 2 %.

9- References:-

1. Charalambous P. Charalambos, Victor S. Frost, Joseph B. Evans, "Performance of TCP extensions on noisy high BDP networks", August 26, 1998.
2. Azer Bestavros, Gitae Kim, "A fragmentation-tolerant TCP protocol for ATM networks", Boston University, MA 02215.
3. Stalling, W., "ISDN and Broadband ISDN with frame relay and ATM", Prentice Hall, New Jersey 07458, book, Fourth Edition, 1999.
4. Black, U., "ATM, Volume II, Signaling In Broadband Networks", Prentice Hall. N. J. 07458, Fourth Edition, book, 1999.
5. Vojnovic, M. and J. Boudec, "How Unfair are the ABR Binary Schemes, Actually?", Institute for Computer Communications and Applications, 2000.
6. Wei Lu, "Study on ATM enhancement and Simulation software design", Southeast University, Radio Engineering Department, Thesis, June 2000.
7. Magnus Thors, "Demonstration of Echo Cancellation in AXD301", Royal Institute of Technology, Thesis, 2000.
8. Nana, P., "Congestion Control: An Essential Part Efficiency in ATM Networks", Networking Programming (Course project), pp.1-10, April 2001.
9. Halsall, F., "Multimedia Communications", Person Education Limited, 2001.

10. Yokotani, T., "ATM Traffic Control for Guaranteed QoS", IEEE Network, 15(3):48-53, May 2001.
11. Girroux, N. and S. Ganti, "Quality of Service in ATM Networks: state of the Art Traffic Management", Prentice Hell PTR, NJ 07458, 2001.
12. Evangelos Vlachogiannis, "UMTS Core Packet-Switched Network", University of Essex, Department of Electronic System Engineering, Thesis, 2001.
13. Dongfang Zheng, " Dynamic Optimization of Virtual Path System with Stochastic Traffic Flows in Satellite ATM Networks", University of Alberta, thesis, 2002.
14. Kelly, T., S. Floyd, and S. Shenker, "Patterns of Congestion Collapse", International Computer Science Institute, University of Cambridge, 2003.
15. Trevor Blackwell, "Speeding up Protocols for Small Messages", Harvard University, Cambridge MA 02138.
16. Xia, Y., D. Harrison, S. Kalyanaraman, K. Ramachandran, A. Venkatesan, "An Accumulation-based Congestion Control Model", In Proceeding of the SIGCOMM, August 2003.
17. Weidong Lu, "Designing TCP/IP Functions In FPGAs", Computer Engineering, Delft University of Technology, thesis, 2003.
18. Behrouz A. Forouzan, "TCP/IP Protocol Suite", McGraw-Hill, book 2003.
19. Areej H., A., "Modeling and Simulation of ATM Congestion Control Switching Techniques", University of Technology, thesis, 2004.
20. The Open Group, "ATM", Fujitsu Network Communications Inc. (FNC), 2004.
21. Jing Yan, "Fully Integrated and Switched Test Environment and Automated Testing (FIST)", Linköping Institute of Technology, Thesis, March 2006.
22. Shady Grove Road, Suite and Rockville, "TCP Revisited", publication, Hughes Systique Corporation, 2006.
23. DAN LIU, "Improving start-up and staeady-state Behavior of TCP in high bandwidth-delay product networks ", Department of Electrical Engineering and Computer Science, Case Western Reserve University, 2007.

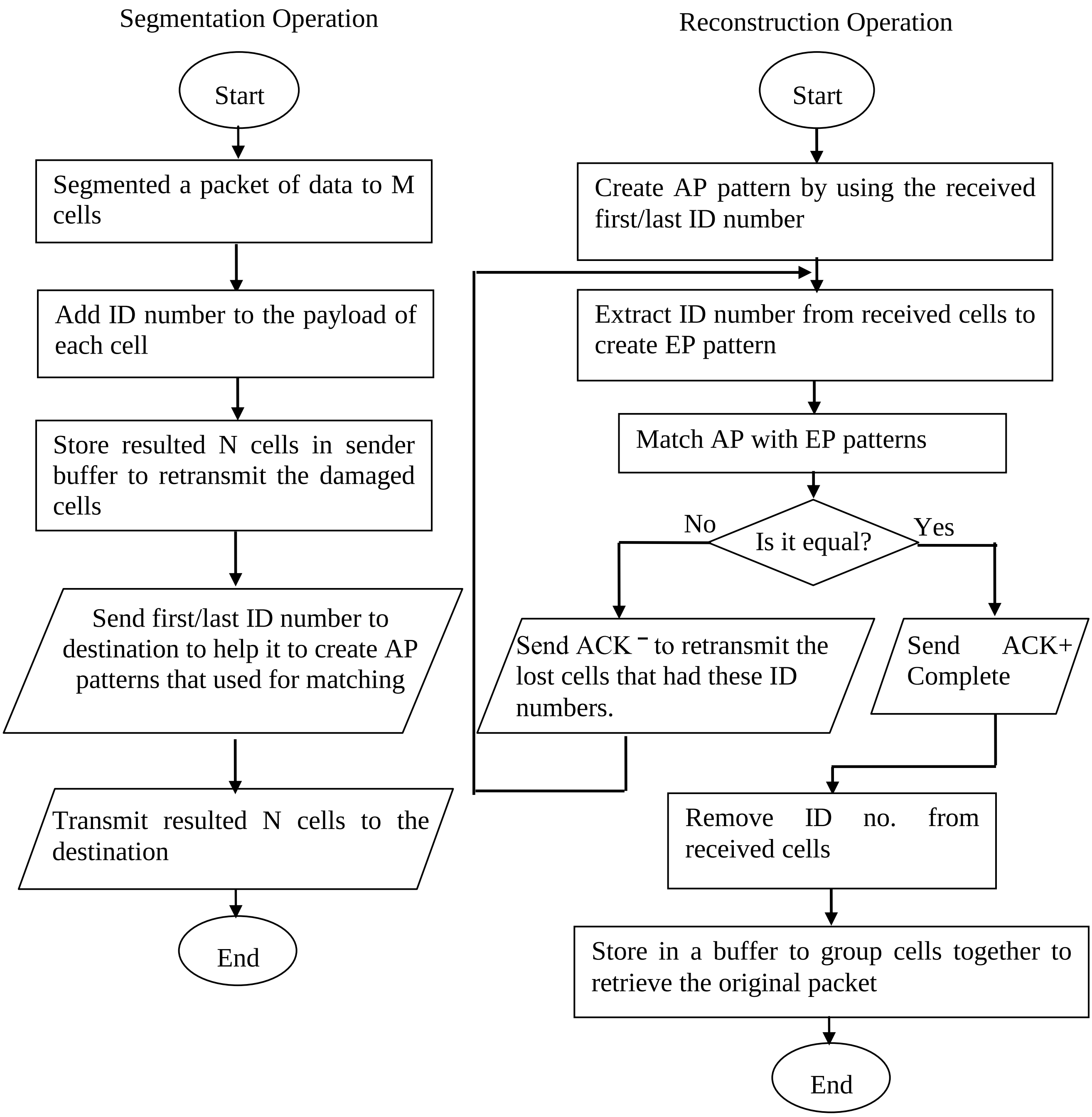

Figure 1: The Segmentation and Reconstruction operations of AIDM algorithm

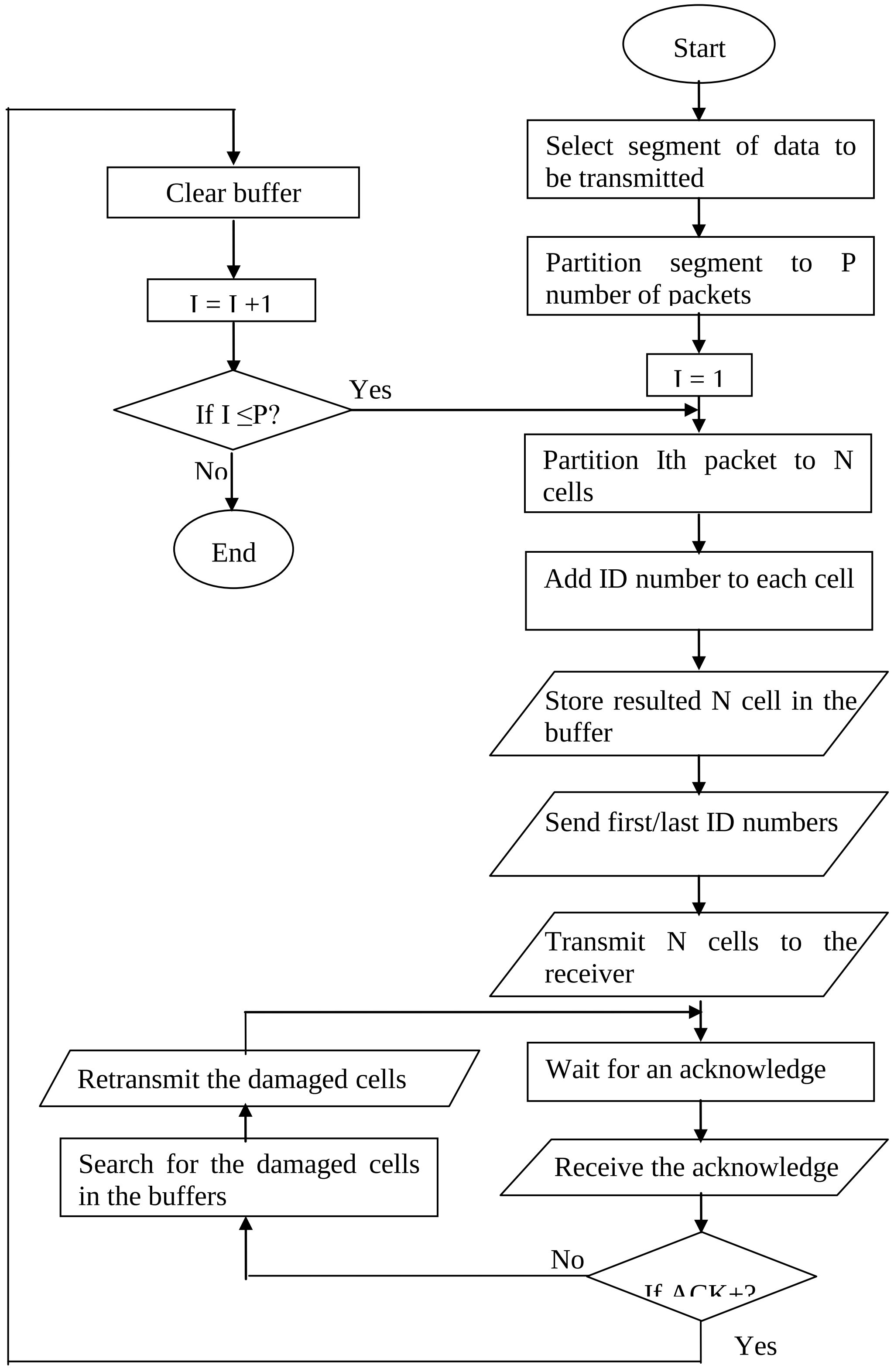

Figure 2: Flowchart of sender, which applies improved TCP Reno.

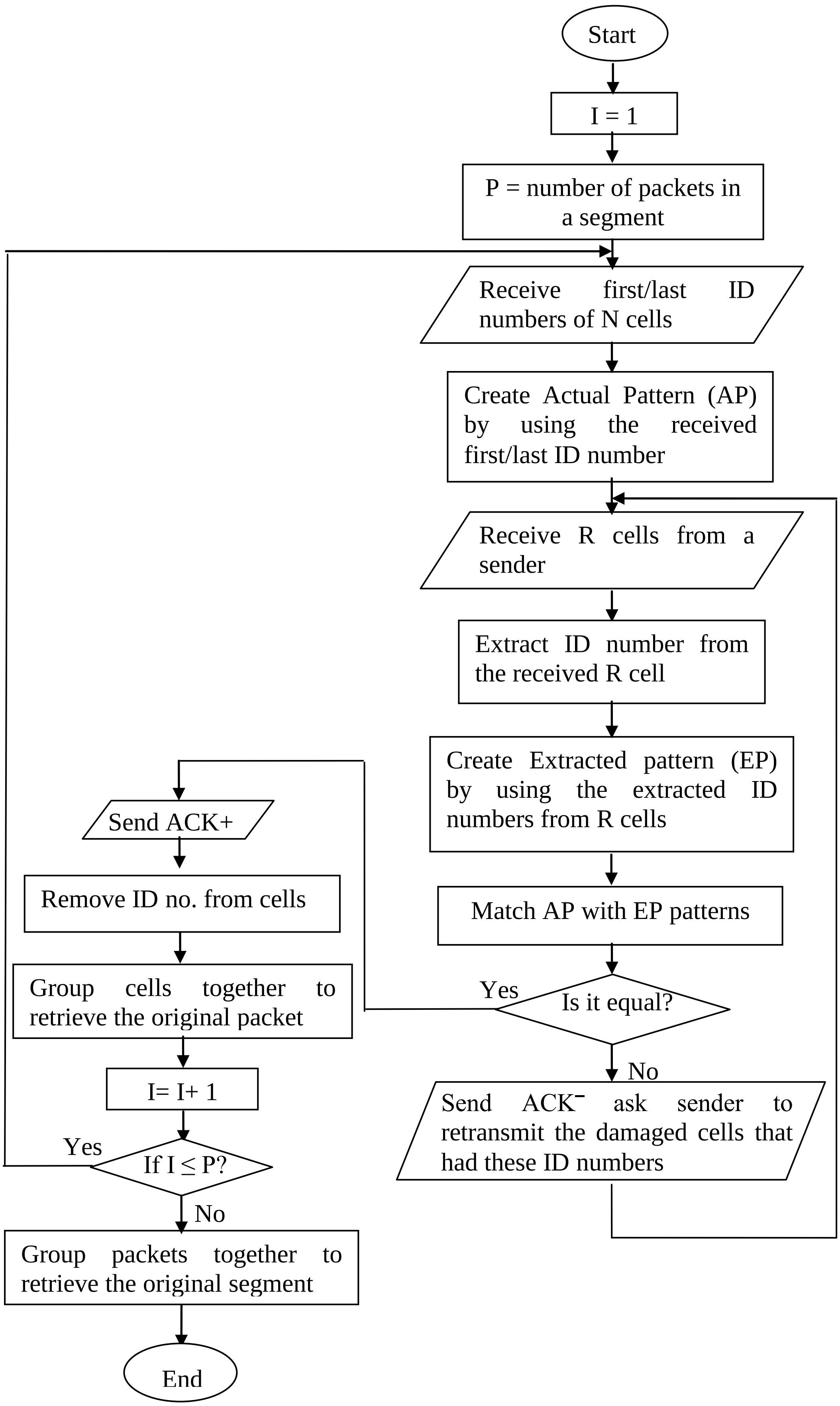

Figure 3: Flowchart of receiver, which applies Improved TCP Reno.

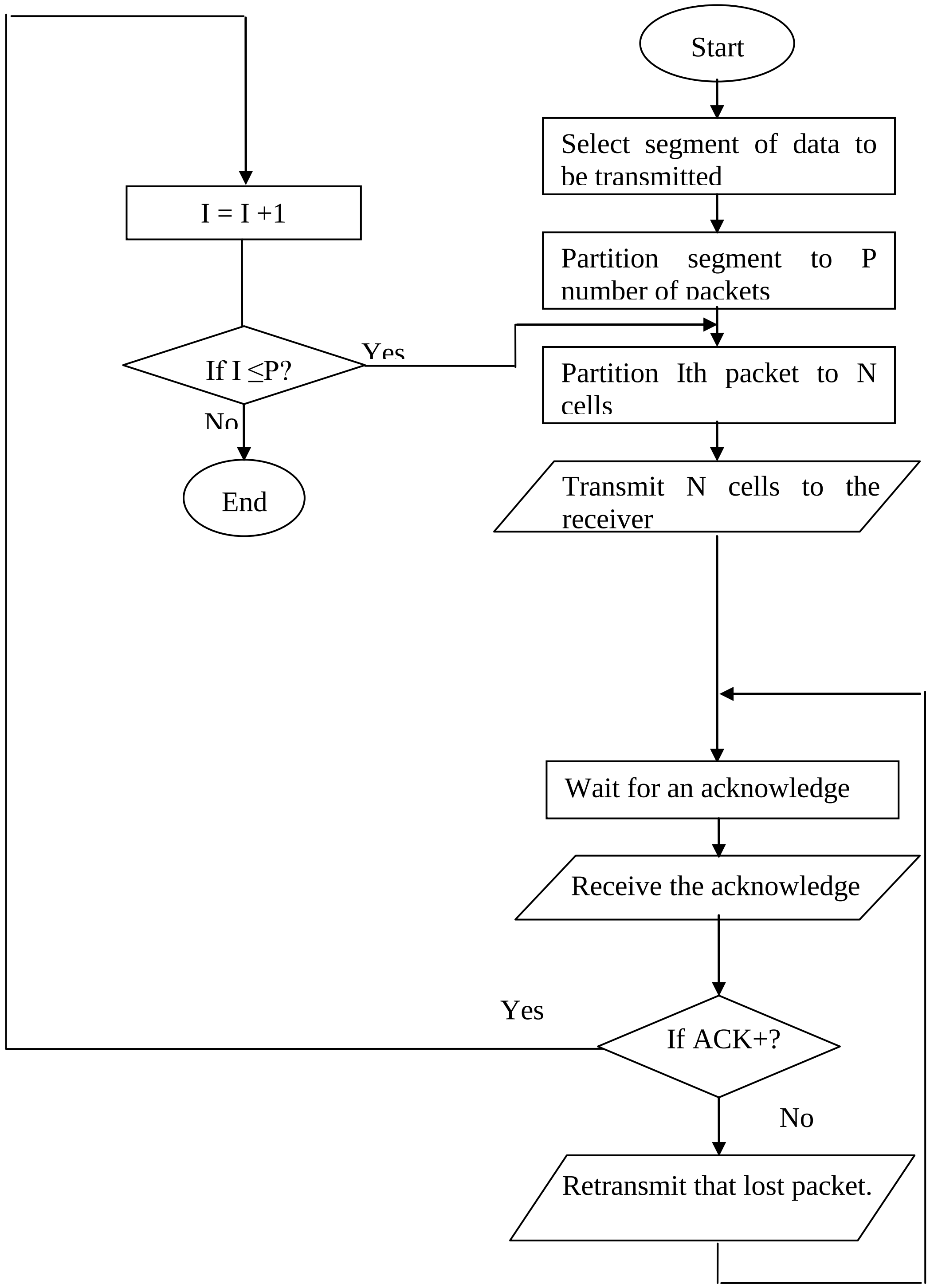

Figure 4: Flowchart of sender, which applies TCP Reno.

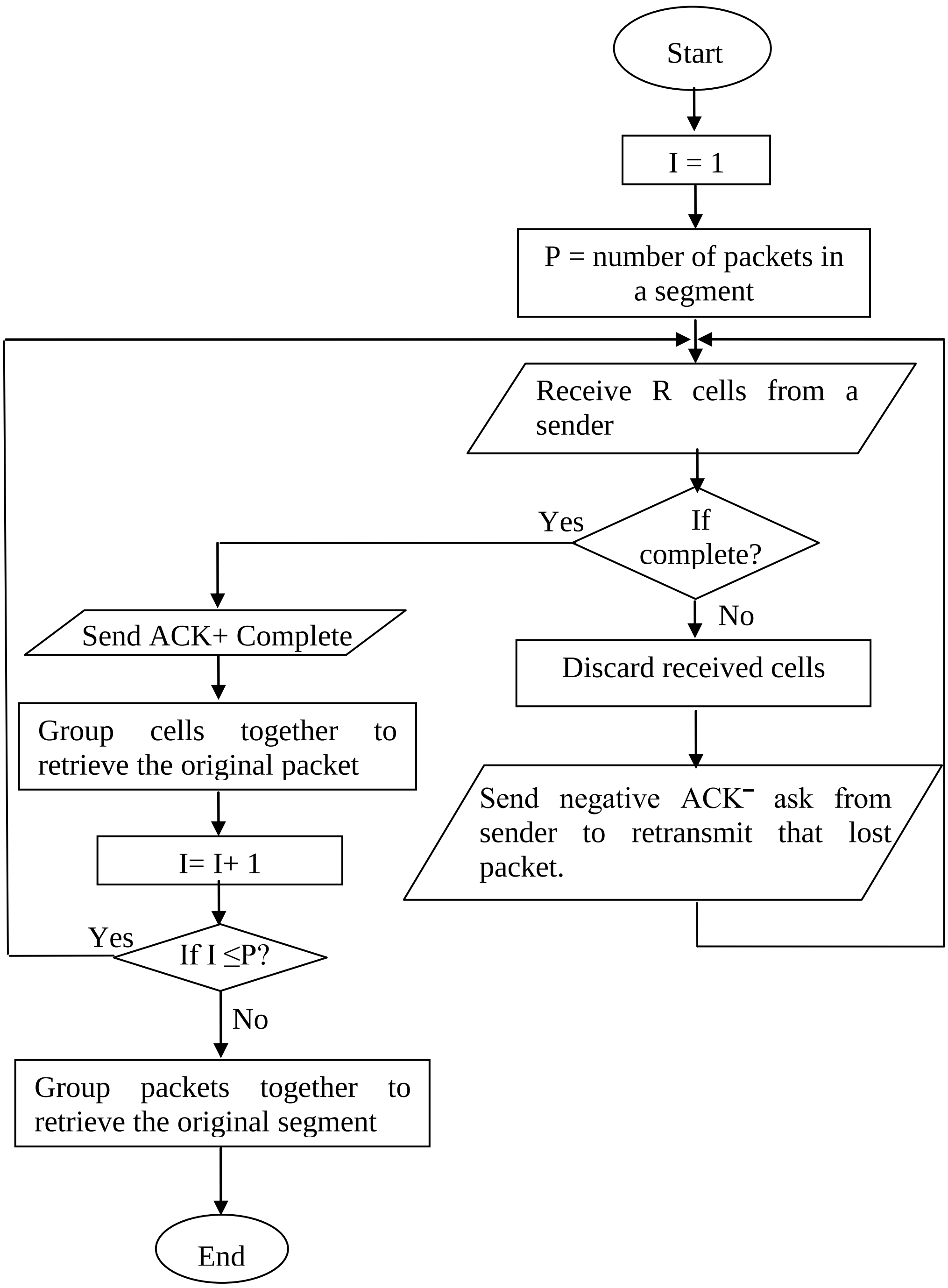

Figure 5: Flowchart of receiver, which applies TCP Reno

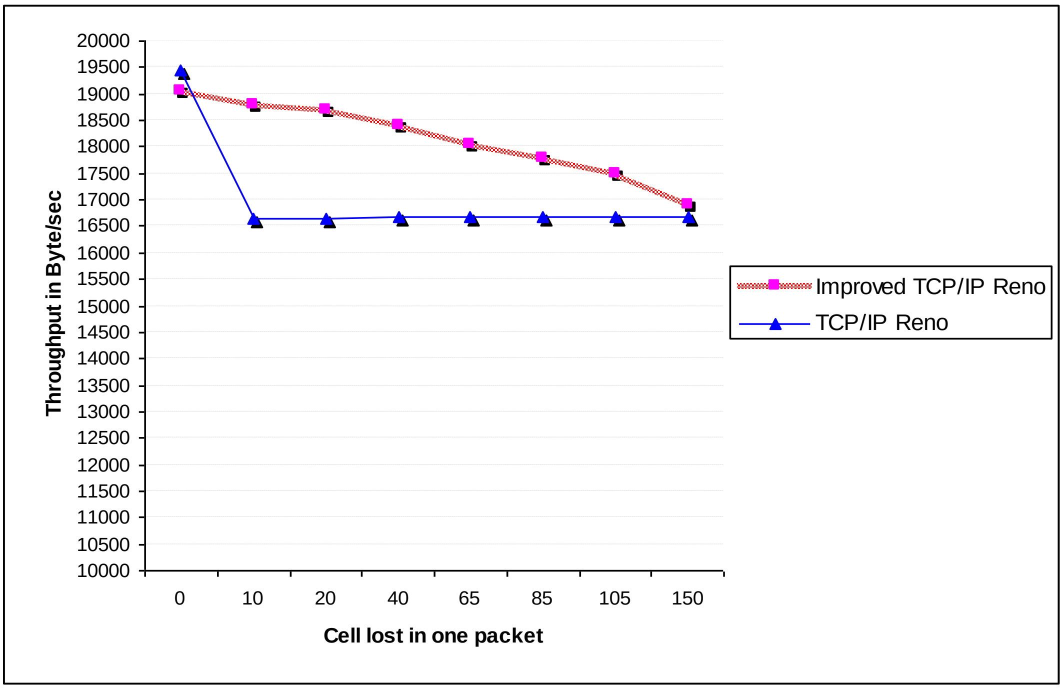

Chart (1) illustrates how throughput changed when cells lost increased in one packet.

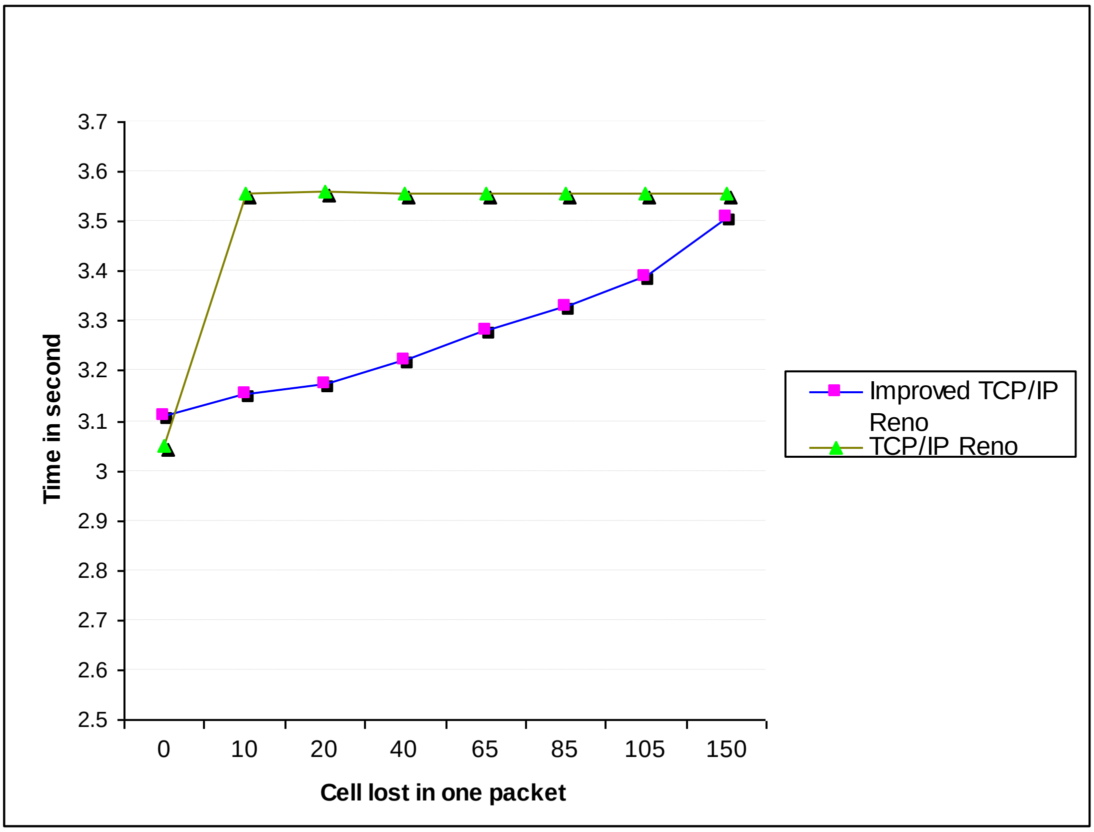

Chart (2) illustrates how required time for transmission changed when cells lost increased in one packet.

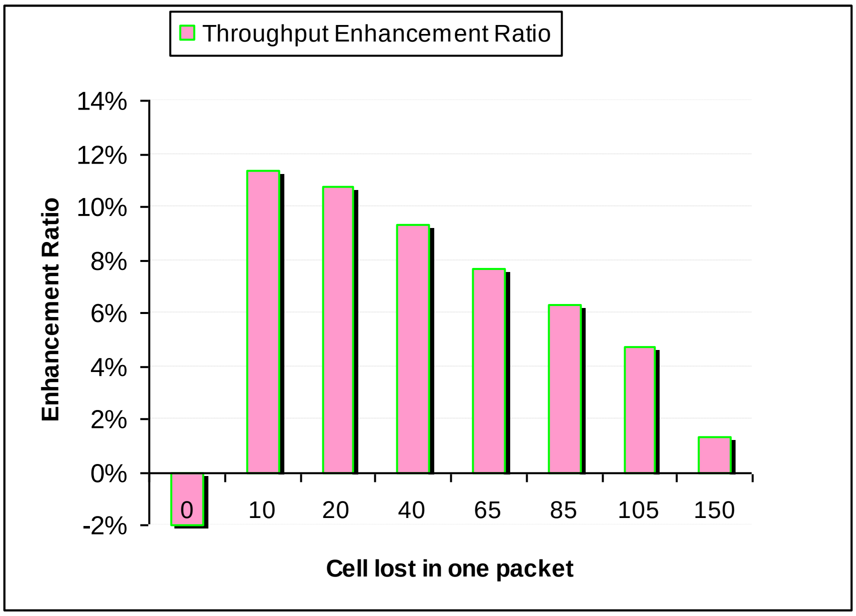

Chart (3) illustrates how enhancement rate changed when cells lost increased in one packet.

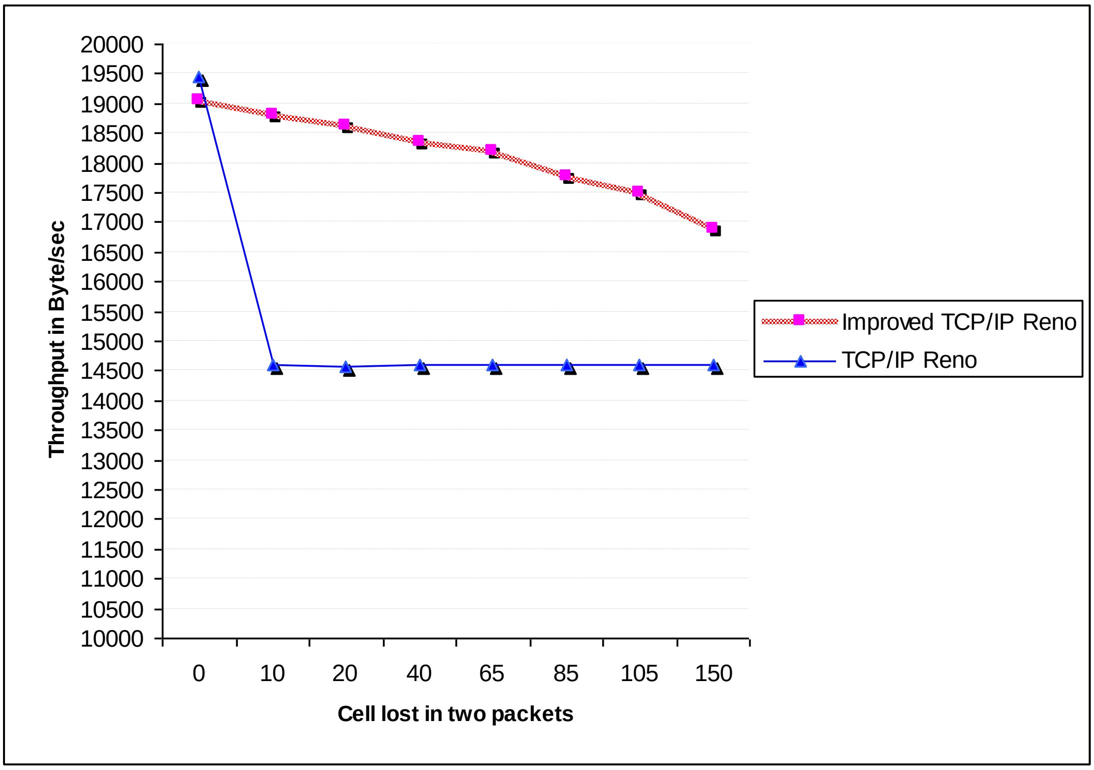

Chart (4) illustrates how throughput changed when cells lost increased in two packets.

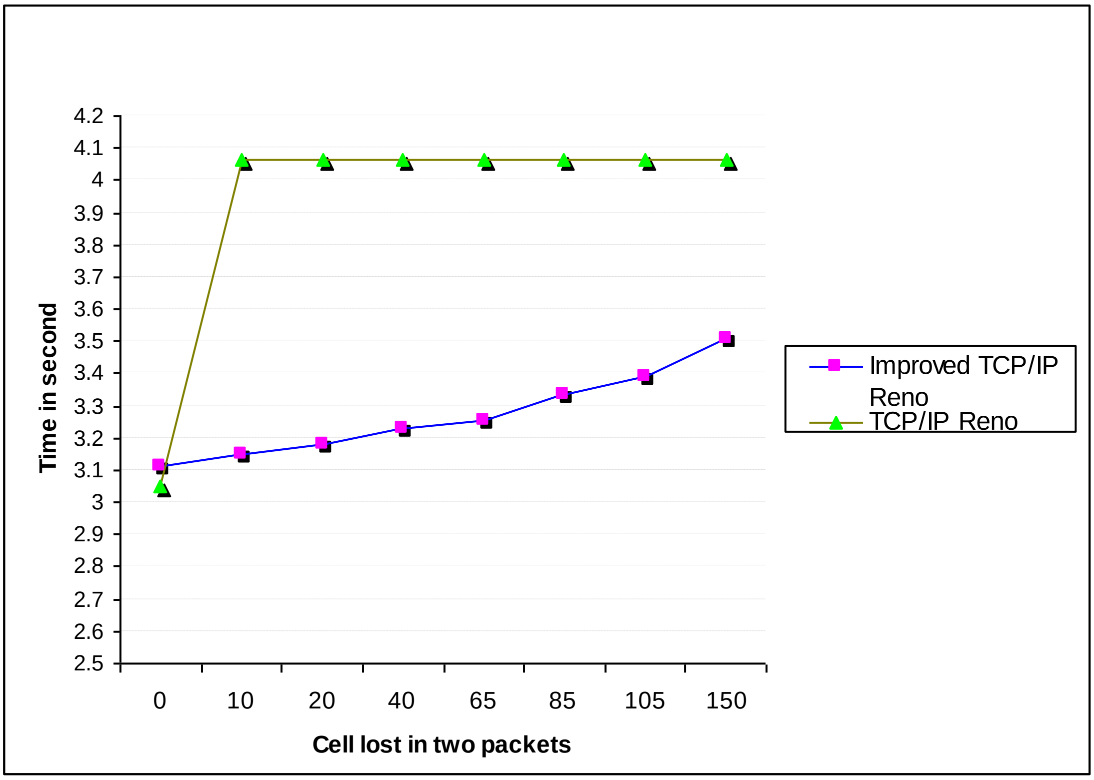

Chart (5) illustrates how required time for transmission changed when cells lost increased in two packets.

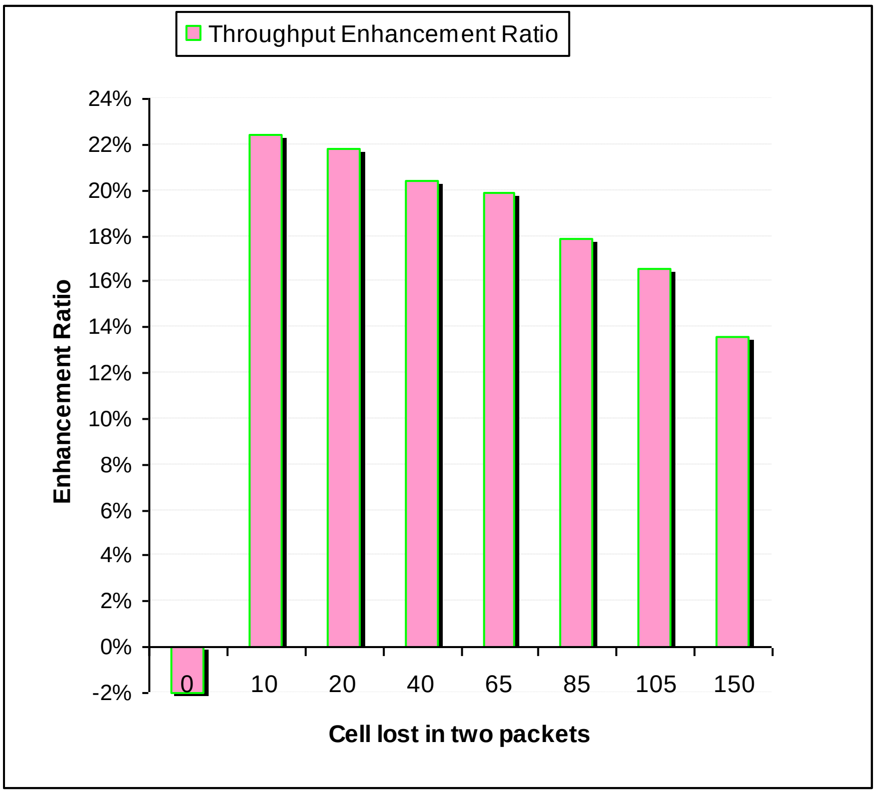

Chart (6) illustrates how enhancement rate changed when cells lost increased in two packets.

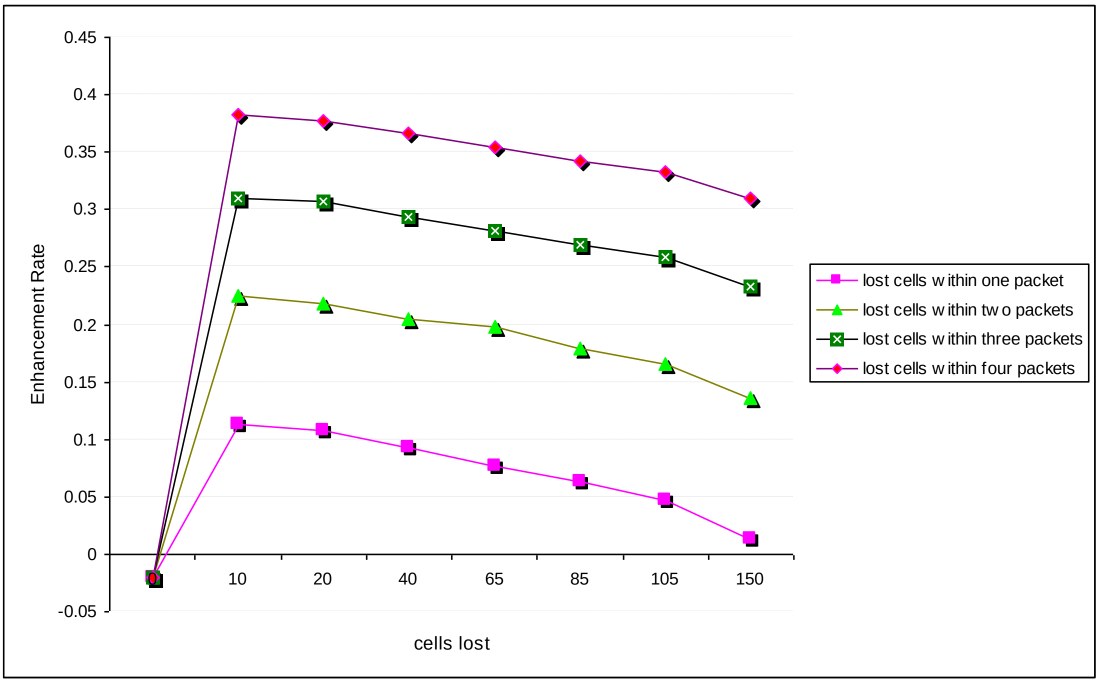

Chart (7) illustrates how enhancement rate changed when cells lost increased in one, two, three, and four packets.