International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-2 Issue-5, November 2012

Analysis of Sliding Window Protocol for Connected Node

Kapil Kumar Bansal, Rakesh Kumar Yadav

Abstract— Abstract- In conventional networks lack of reliable and efficient transmission of data over unreliable channels that can lose, reorder and duplicate messages due to router or receiver limited buffer space, are retransmitted by the source. These scenarios usually data blocks contain error detection codes to detect whether an error has occurred during data transmission or not. If an error is detected, the receiver can try to fix the error if the received data has enough redundant bits or request a retransmission of data, for fulfill this requirement sliding window protocol is use. This paper shows calculated link utilization is better than observed link utilization for sliding window protocol using Deft Netz2.0 network simulation.

Index Terms-- Network, Congestion, Buffer, Simulation, Bandwidth, Reliable, Efficient, Netz2.0.

I. INTRODUCTION

In networks reliable transmission of data over unreliable channels is big problem in computer networks. Without a satisfactory solution, computer networks would be useless, because they transmit data over channels that often lose, duplicate, or reorder messages. As a result, the received signal is often badly distorted that the received message cannot be reconstructed unless some kind of error control is used. Basically, there are two approaches to error control in digital communications: forward error correction (FEC) and automatic repeat request (ARQ).

In telecommunication and information theory, forward error correction (FEC) (also called channel coding [1]) is a system of error control for data transmission, whereby the sender adds redundant data to its messages, also known as an error-correcting code. This allows the receiver to detect and correct errors without the need to ask the sender for additional data. The advantages of forward error correction are that a back-channel is not required and retransmission of data can often be avoided. FEC is therefore applied in situations where retransmissions are relatively costly or impossible. In particular, FEC information is usually added to most mass storage devices to protect against damage to the stored data. The FEC systems are designed for use in simplex channels. Another scheme, in an automatic repeat request (ARQ) is use high-rate error-detecting code together with some retransmission protocol. If the receiver detects errors in the received word, it generates a retransmission request, or a negative acknowledgement (NACK). If no errors are detected in the received word, the receiver sends positive acknowledgements, called an ACK, to the transmitter. Cyclic redundancy check (CRC) codes are the most widely used error-detecting codes because of the ease of their implementation. Unlike the FEC systems, the ARQ schemes require the presence of a feedback channel. One of the most efficient protocols for reliable transmission is the sliding window protocol [2]. This paper, show the performance of sliding window protocol will be simulated and analyzed under various conditions.

The remainder of this paper is organized as follows. The next section gives an overview of the sliding window protocol. In section III, the simulation results shows. Finally conclusions are summarized in the last section.

Manuscript Received on October, 2012.

Dr. Kapil Kumar Bansal, Head Research and Publication, SRM University, NCR Campus Modinagr, Ghaziabad (U.P.), India

Rakesh Kumar Yadav, Department of Computer Science & Engineering, SRM University, NCR Campus Modinagr, Ghaziabad (U.P.), India

II. SLIDING WINDOW PROTOCOL DESCRIPTION

The sliding window protocol is an established protocol in the ISO-OSI protocol stack. Sliding window protocol assumes two-way communication. It uses two types of frames first data second acknowledgment. The basic idea of sliding window protocol is that both sender and receiver keep a "window" of acknowledgement. The sender keeps the value of expected acknowledgement, while the receiver keeps the value of expected receiving frame. When it receives an acknowledgement from the receiver, the sender advances the window. When it receives the expected frame, the receiver advances the window. The sliding window protocol has been widely used in many popular communication protocols such as TCP, HDLC and SPX. The protocol can ensure a correct data transfer over poor quality communication channels where the packets may be duplicated, lost, or re-ordered [3][4][5]. In next section we present the basics of a sliding window protocol.

A. Sender and Receiver

In a sliding protocol, there are two main components: the sender and the receiver. The sender obtains an infinite sequence of data from the sending host. We call indivisible blocks of data in this sequence "frames", and the sequence itself the "input sequence". The input sequence must be transmitted to the receiver via an unreliable network. After receiving a frame via the channel, the receiver may decide to accept the frame and eventually deliver it to the receiving host. The correctness condition for a sliding window protocol says that the receiver should deliver the frames to the receiving host in the same order in which they appear in the input sequence.

B. Messages and Channels

In order to transmit a frame, the sender puts it In order to transmit a frame; the sender puts it into a frame message together with some additional information, and sends it to the frame channel.

After the receiver eventually receives the frame message from this channel, it sends an acknowledgment message for the corresponding frame back to the sender. This acknowledgment message is transmitted via the acknowledgment channel. After receiving an acknowledgment message, the sender knows that the corresponding frame has been received by the receiver. Thus the communication between the sender and the receiver is bi-directional; the sender transmits frames to the receiver via the frame channel, and the receiver transmits acknowledgments for these frames to the sender via the acknowledgment channel.

C. Sequence Numbers

The sender sends the frames in the same order in which they appear in its input sequence. However, the frame channel is unreliable, so the receiver may receive these frames in a very different order. Therefore it is clear that each frame message must contain some information about the order of the corresponding frame in the input sequence. Such additional information is called sequence number. In the sliding window protocol, instead of the exact position of the frame in the input sequence, the sender sends the remainder of this position with respect to some fixed modulus.

D. Sending and Receiving Windows

At any instant of time, the sender maintains a sending window of consecutive sequence numbers corresponding to frames it is permitted to send. These sequence numbers within the sending window represent frames sent but not yet acknowledged. Similarly, the receiver maintains a receiving window, corresponding to the number of out of order frames it is permitted to accept. Any frames falling outside the receiving window are discarded without comment.

E. Communication Procedure

In the beginning of communication between the sender and receiver, the sending window and the receiving window are empty. The sender is starting to send data frames to the receiver via the frame channel. On the other side, the receiver is waiting for receiving data frames from the network. Whenever a new packet arrives at the sender from the network layer, the next highest sequence number according to the sending window is given, and the upper edge of the sending window is advanced by one. The sender puts the packet and the sequence number with some other control information into a frame, and then sends the frame to the frame channel.

After transmitting a frame and starting a timer, the sender will transmit the next data frame to the frame channel until the sending window is filled up. In the same time, the sender is waiting for the acknowledgement arriving from the network. There are three possibilities: an acknowledgement frame arrives undamaged, a damaged acknowledgement frame staggers in, and the timer goes off. If a valid acknowledgement comes in, the sender fetches the next packet from the network layer and put it in the buffer, overwriting the previous packet. It also advanced the input sequence number. If a damaged frame arrives or a timer goes off, a duplicate should be sent. In [6] many different policies for sending and resending of data frames exist. In this paper, we only focus on the sliding window protocol. The sender will send all unacknowledged frames in order, starting with the damaged or lost one. At the receiver, when a valid frame arrives; its sequence number is checked to see if it is the next one. If it is the next one, it is accepted, passed to the network layer and an acknowledgement frame generated. Otherwise, it will be discard and is not passed to the network layer.

III. SIMULATION METHODOLOGY

This section describes the simulation methodology and the parameters used for performance analysis of sliding window protocol. In simulation methodology DEFT NETZ-2.0 simulator used for the simulation of sliding window protocol for network model

A. Simulation Parameters

Following simulation parameters are used in the simulation process.

Table 1. Simulation Parameters

Parameters	Values
No of Nodes	04
Simulation Tolerance	.8 sec
Final propagation Time	100 micro-sec
Frame Transmission Time	1 sec
Increment Step	1
Initial Propagation Time	1 micro-sec
Window Size	7

IV. CONFIGURING THE PROTOCOL TESTER

A number of parameters affect the performance of the sliding window protocol. Here we conclude some key-parameters are used in the simulation process.

A. Simulation Tolerance

Simulation tolerance defines when the simulation would end. If you want precise result, the simulation tolerance should be high, but it would take extremely long to simulate.

B. Frame Transmission Time

Frame transmission time is the time for a frame to be transmitted from source node to destination node.

C. Simulation Tolerance

The increment step tells how the simulation will proceed. This is similar to the steps in which we alter the parameters in any normal experiments to finally obtain plots.

D. Window Size

Window Size is the governing parameter of the sliding window protocol.

E. Bandwidth

It is the maximum amount of data passing the channel at a given time. Usually, it depends on the medium capacity of the communication channels itself.

V. SIMULATION RESULTS

This section presents detailed simulation results of sliding window protocol.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A. Observed Utilization

Figure 4.1 shows the observed link utilization respect to normalized propagation time. The observed link utilization U, with a simulation tolerance of 0.8, window size is 7 and propagation time is 1 to 100 micro-seconds with error free.

Figure 4.1. Observed Link Utilization

B. Calculated Utilization

Figure 4.2 shows the calculated standard results for sliding window protocol, window size is 7, simulation tolerance is 0.8 and propagation time is 1 to 100 micro-seconds with error free.

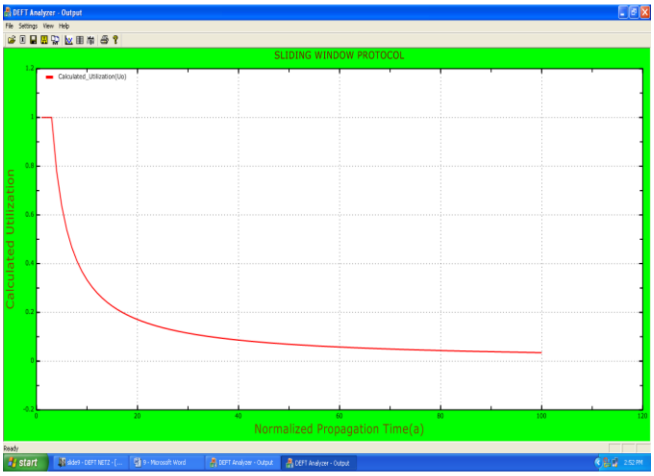

Figure 4.2. Calculated Link Utilization

C. Combined Analysis

Figure 4.3 shows the comparative analysis of observed link utilization and calculated link utilization for sliding window protocol with window size is 7, simulation tolerance is 0.8 and propagation time is 1 to 100 micro-seconds with error free, graph shows calculated link utilization gives better performance than observed link utilization.

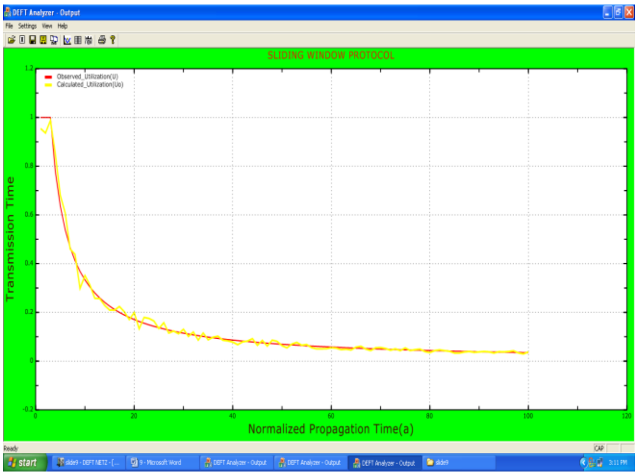

Figure 4.3. Observed and Calculated Utilization

VI. CONCLUSION

This paper presented the performance analysis of sliding window protocol for connected nodes using DEFT NETZ2.0 simulations. In this paper preliminary evaluation of performance, results shown that calculated link utilization is better than observed link utilization for network.

REFERENCES

1. Wang et al., Forward Error-Correction Coding Crosslink - The Aerospace Corporation magazine of advances in aerospace technology. The Aerospace Corporation. Vol-3,Winter 2002.
2. N.V. Stenning. A data transfer protocol. Computer Networks,1(2):99–110, 1976. Eric Madelaine and Didier Vergamini. Specification and Verification of a Sliding Window Protocol in LOTOS. FORTE '91, Sydney, Australia, , pages 495-510 November 1991,.
3. Mark A. Smith and Nils Klarlund. Verification of a Sliding Window Protocol Using IOA and MONA. FORTE/PSTV 2000, Pisa, Italy, pages 19-34, October 2000.
4. Dmitri Chkliaev, Jozef Hooman and Erik de Vink. Formal Verification of an Improved Sliding Window Protocol. In proceeding of the PROGRESS 2002, Utrecht, The Nethelands, pages 18-27, October 2002.
5. A.S. Tanenbaum. Computer Networks. Prentice-Hall International, Inc., 1996.

AUTHORS PROFILE

Rakesh Kumar Yadav, Pursuing Ph.D. Degree and currently, working as a, Assistant Professor in Computer Science and Engineering Department since 2009. He has published 15 research papers in international journal and conferences. The area of research is wireless communication and wireless sensor network.