International Journal of Education and Science Research
Review E-ISSN 2348-6457
Volume-3, Issue-2 April- 2016
www.ijesrr.org Email- editor@ijesrr.org

Error and Flow Control Selective Repeat Sliding Window Protocol

Shikhar Kapoor, Oshi, Niti Arora
Department Of Information Technology
SRM University, NCR Campus

ABSTRACT:-
The perspective of this paper is to provide an easy to use visualization system for the sliding window protocol. The SWV or the sliding window visualization system is constructed to understand the error and flow control mechanisms of the protocols of the data link layer. This is an interactive system and gives the authority to the user to modify some parameters of the protocol in accordance with the needs. In this paper a visualization system will be developed where a user will be able to understand the principle of working of the sliding window protocol and will also be able to make comparisons between any two algorithms. Apart from the visualization of this network protocol, data packets are also being sent at the back end which gives an opportunity to the user to understand the real time data transfer. The major purpose is to help visualize newly formulated algorithms where a mathematical model of the algorithm is developed by a researcher which in turn is converted into a visualized form by the developer. This results in easy comparisons of algorithms based on various parameters. This paper is designed in a manner that it is open for future enhancements and various modules can be added to the system without modifications in the code for new algorithms efficiency to be compared with respect to present algorithms. Since the backbone of this visualization system is the tasks and needs of the users, it can be used as a teaching tools and community and college level effectively and also for the purpose of self-study once the student is being introduced with the DLL protocols. Sliding window visualization system's strong point is that it helps articulating the mental images of the mechanisms of protocols and also aids in easy and experimentations using supported protocols.

KEYWORDS-sliding window protocol, visualization

INTRODUCTION:-

The affordability and advancements in technology used for computing along with high performance graphical hardware has resulted in rapid growth and usage of visual tools. Today from most expensive workstations to low cost personal computers inherits the capability of running visualization systems effectively. Visualization system is a system that generates visual images of a data such a graphs, images, histograms etc. Although a lot of visualizations systems have been developed in the past years but the attention is not paid on user needs, comfort and their tasks. Hence, this paper focuses on making the visualization system an easy to use and user friendly one so that it can benefit the masses. Along with the visualization of the protocols, data packets are also sent at the back end which gives an opportunity to the users to understand the real time data transfer amongst and between the physically separated machines or system.

In this paper we are going to implement a bidirectional sliding window protocol which comprises of a forward channel for sending of data packets and a reverse channel for receiving the data packets and also sending of acknowledgement back to the sender once the packet is being received.

SWV is designed is such a way that it will meet the following goals:-

1) Interactive and visually animated
2) Parameters can be modified

3) Events of protocols can be generated at the runtime and the student/user watches the live protocol reactions.
4) Platform independent and easy to use and provide immediate benefits and results in absolutely no time.

LITERATURE SURVEY:-

1) DATA LINK LAYER:

The data link layer or layer 2 is the second layer of the seven-layer OSI model of computer networking. This layer is the protocol layer that transfers data between adjacent network nodes in a wide area network (WAN) or between nodes on the same local area network (LAN) segment.[1] The data link layer provides the functional and procedural means to transfer data between network entities and might provide the means to detect and possibly correct errors that may occur in the physical layer.

The data link layer is concerned with local delivery of frames between devices on the same LAN. Data-link frames, as these protocol data units are called, do not cross the boundaries of a local network. Inter-network routing and global addressing are higher-layer functions, allowing data-link protocols to focus on local delivery, addressing, and media arbitration. This way, the data link layer is analogous to a neighbourhood traffic cop; it endeavours to arbitrate between parties contending for access to a medium, without concern for their ultimate destination. When devices attempt to use a medium simultaneously, frame collisions occur. Data-link protocols specify how devices detect and recover from such collisions, and may provide mechanisms to reduce or prevent them.

2) UNRESTRICTED SIMPLEX PROTOCOL (UTOPIA):

In unrestricted simplex protocol the data transmission in one direction only i.e it is simplex. Also, no errors takes place on the physical channel and the sender/receiver can generate/consume an infinite amount of data and the channel is always ready and in a working condition for the sending/receiving of data packets.

3) SIMPLEX STOP AND WAIT PROTOCOL:

Simplex stop and wait protocol is the protocol that was developed in order to prevent the problem of data packs being lost midway or not received by the receiver. In this protocol, the sender sends a data packet to the receiver and waits for a n acknowledgement from the receiver. The receiver on receiving the data packet send back an acknowledgement frame to the sender. An enhancement to this protocol is to add a 1 bit sequence number to the data frames being sent. This enhancement in the simplex stop and wait protocol is known as Positive Acknowledgement with Retransmission (PAR) which is the protocol supported by the SWV system.

4) SLIDING WINDOW PROTOCOL:

A sliding window protocol is a feature of packet-based data transmission protocols. Sliding window protocols are used where reliable in-order delivery of packets is required, such as in the Data Link Layer (OSI model) as well as in the Transmission Control Protocol (TCP).

Conceptually, each portion of the transmission (packets in most data link layers, but bytes in TCP) is assigned a unique consecutive sequence number, and the receiver uses the numbers to place received packets in the correct order, discarding duplicate packets and identifying missing ones. The problem with this is that there is no limit on the size of the sequence number that can be required.

By placing limits on the number of packets that can be transmitted or received at any given time, a sliding window protocol allows an unlimited number of packets to be communicated using fixed-size sequence numbers. The term "window" on the transmitter side represents the logical boundary of the total number of packets yet to be acknowledged by the receiver. The receiver informs the transmitter in each acknowledgment packet the current maximum receiver buffer size (window boundary). The TCP header uses a 16 bit field to report the receive window size to the sender. Therefore, the largest window that can be used

is 2^{16} = 64 kilobytes. In slow-start mode, the transmitter starts with low packet count and increases the number of packets in each transmission after receiving acknowledgment packets from receiver. For every ack packet received, the window slides by one packet (logically) to transmit one new packet. When the window threshold is reached, the transmitter sends one packet for one ack packet received. If the window limit is 10 packets then in slow start mode the transmitter may start transmitting one packet followed by two packets (before transmitting two packets, one packet ack has to be received), followed by three packets and so on until 10 packets. But after reaching 10 packets, further transmissions are restricted to one packet transmitted for one ack packet received. In a simulation this appears as if the window is moving by one packet distance for every ack packet received. On the receiver side also the window moves one packet for every packet received. The sliding window method ensures that traffic congestion on the network is avoided. The application layer will still be offering data for transmission to TCP without worrying about the network traffic congestion. The window size can be dynamically varied in accordance to network traffic and congestion.

A) ONE-BIT SLIDING WINDOW PROTOCOL:

This is most simple form of the sliding window protocol where the window size of the sender window as well as the receiving window is one.

B) GO-BACK-N SLIDING WINDOW PROTOCOL:

Go-Back-N sliding window is a specific instance of the automatic repeat request (ARQ) protocol, in which the sending process continues to send a number of frames specified by a window size even without receiving an acknowledgement (ACK) packet from the receiver. It is a special case of the general sliding window protocol with the transmit window size of N and receive window size of 1. It can transmit N frames to the peer before requiring an ACK.

The receiver process keeps track of the sequence number of the next frame it expects to receive, and sends that number with every ACK it sends. The receiver will discard any frame that does not have the exact sequence number it expects (either a duplicate frame it already acknowledged or an out-of-order frame it expects to receive later) and will resend an ACK for the last correct in-order frame. Once the sender has sent all of the frames in its window, it will detect that all of the frames since the first lost frame are outstanding, and will go back to the sequence number of the last ACK it received from the receiver process and fill its window starting with that frame and continue the process over again.

Go-Back-N is a more efficient use of a connection than Stop-and-wait, since unlike waiting for an acknowledgement for each packet, the connection is still being utilized as packets are being sent. In other words, during the time that would otherwise be spent waiting, more packets are being sent. However, this method also results in sending frames multiple times – if any frame was lost or damaged, or the ACK acknowledging them was lost or damaged, then that frame and all following frames in the window (even if they were received without error) will be re-sent. To avoid this, Selective Repeat can be used.

C)SELECTIVE REPEAT SLIDING WINDOW PRTOCOL:

Selective Repeat is part of the automatic repeat-request (ARQ). With selective repeat, the sender sends a number of frames specified by a window size even without the need to wait for individual ACK from the receiver as in Go-Back-N ARQ. The receiver may selectively reject a single frame, which may be retransmitted alone. This contrasts with other forms of ARQ, which must

send every frame from that point again. The receiver accepts out-of-order frames and buffers them. The sender individually retransmits frames that have timed out.

IMPLEMENTATION:

Implementation is that stage of a project defined as the process of putting a decision, design or plan into effect or the execution stage of a project. And therefore, it is the most important stage of the entire research process to achieve a successfully designed, implemented and working system.

1) SUPPORTED PROTOCOLS:

Out of the protocols mentioned previously the protocols that will be supported by the visualization system being designed in this paper are as follows:-

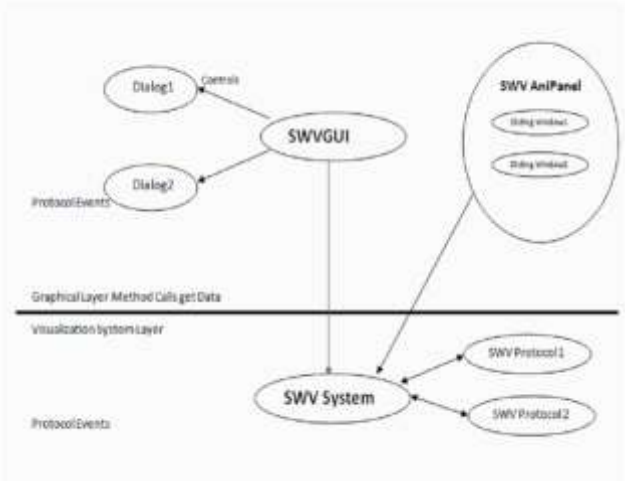
a) Go-Back-N visualization

b) Selective Repeat visualization

c) Simulation with data transfer for selective repeat ARQ

We found it more interesting to concentrate on the protocols which are difficult to understand, so that the visualization system makes them easy to implement and understand.

2) DESIGN OF SLIDING WINDOW VISUALIZATION SYSTEM:



Shown above is a well descriptive and detailed diagram of the design of the sliding window visualization system.

At the bottom or the core a framework is present that allows the controlling of animations and aids in feeding of animated and visualized output of the protocols to users using the visualization system.

The protocols used can be modified anytime in accordance to the needs of the users using the system. This entire process is platform independent and also framework independent. The protocol if required can use sliding window optionally a fixed maximum sequence or the window size. A graphical display layer is present which includes a sliding window component in the encapsulated form and can be used by other java application with the minimum requirement of a sliding window GUI control.

FUNCTIONAL REQUIREMENTS:

1) The system should define at least two algorithms to compare the working.

2) The system shall provide appropriate visualization for the user to understand the respective algorithm.

3) The user should have the liberty to change the parameters in order to properly visualize the response of the algorithm in different scenarios.

4) The system should be accurate enough in order to show the differences among all the algorithms to the user.

5) Appropriate message should be displayed after the completion of each task.

APPLICATIONS:

1) The sliding window visualization system or SWV can be used as a teaching technique for a term at community college level.
2) This SWV will also be useful in a laboratory for a self-study situation after being introduced to DLL protocols.
3) Sliding window visualization system's strong point is in helping to create mental images of the protocol mechanisms, and in allowing easy and painless experimentation with the supported protocols.

EXISTING PROBLEMS:

1) There are many underutilized techniques for creating effective visualization which will continue to spark the creativity of the developers in the years to come.
2) But the main problem still remains the same, there is no single visualization, specialized or general purpose that can satisfy all kinds of users,
all kinds of tasks and can be used in all kind of environments.
3) Without the careful analysis of the users need, tasks, scope, information, and resources the effort put in developing a visualization package will probably be wasted.

CONCLUSION:

We conclude that if the sliding window visualization system is properly and effectively used than it can be proved useful at the community and college level for studying and understanding various protocols and also for creating, examining and formulating new algorithms as well as studying and refining the old algorithms.

REFERENCES:

1) Google.com
2) Wikipedia.com
3) Computer Networks-Andrew S. Tanenbaum