ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικό και Καποδιστριακό
Πανεπιστήμιο Αθηνών

ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ &
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Advanced Network Architectures (M132)

Assignment 2

Link Layer control mechanisms and performance.

Name:	**Evangelos Siatiras**
e-mail:	**EN2190001@di.uoa.gr**

1. Importance of including flow control of data units (packets) at the link layer

Flow control [1] in the transport-layer protocol is provided on an end-to-end basis, whereas it is provided in a link-layer protocol on a node-to-adjacent-node basis. Flow control is a service offered by a link layer protocol where the nodes on each side of a link have a limited amount of packet buffering capacity. This is a potential problem, as a receiving node may receive frames at a rate faster than it can process the frames (over some time interval). Without flow control, the receiver's buffer can overflow, and frames can get lost. Similar to the transport layer, a link-layer protocol can provide flow control in order to prevent the sending node on one side of a link from overwhelming the receiving node on the other side of the link.

2. Link layer flow control mechanisms

Further to the rfc3366 [2], a link ARQ protocol uses a link protocol mechanism to allow the sender to detect lost or frames and to schedule retransmission. Detection of frame loss may be via a link protocol timer, by detecting missing positive link acknowledgement frames, by receiving explicit negative acknowledgement frames and/or by polling the link receiver status. Whatever mechanisms are chosen, there are two easily-described categories of ARQ retransmission process that are widely used:

2.1 Stop-And-Wait ARQ

A sender using stop-and-wait ARQ (sometimes known as 'Idle ARQ' [LIN93]) transmits a single frame and then waits for an acknowledgement from the receiver for that frame. The sender then either continues transmission with the next frame, or repeats transmission of the same frame if the acknowledgement indicates that the original frame was lost or corrupted. Stop-and-wait ARQ is simple, if inefficient, for protocol designers to implement, and therefore popular, e.g., tftp [RFC1350] at the transport layer. However, when stop-and-wait ARQ is used in the link layer, it is well-suited only to links with low bandwidth-delay products.

2.2 Sliding-Window ARQ

A protocol using sliding-window link ARQ [LIN93] numbers every frame with a unique sequence number, according to a modulus. The modulus defines the numbering base for frame sequence numbers, and the size of the sequence space. The largest sequence number value is viewed by the link protocol as contiguous with the first (0), since the numbering space wraps around TCP is itself a sliding-window protocol at the transport layer [STE94], so similarities between a link-interface-to-link-interface protocol and end-to-end TCP may be recognizable. A sliding-window link protocol is much more complex in implementation than the simpler stop-and-wait protocol, particularly if per-flow ordering is preserved. At any time, the link sender may have several frames outstanding and awaiting acknowledgement, up to the space available in its transmission window. A sufficiently large link sender window (equivalent to or greater than the number of frames sent, or larger than the bandwidth*delay product capacity of the link) permits continuous transmission of new frames. A smaller link sender window causes the sender to pause transmission of new frames until a timeout or a control frame, such as an acknowledgement, is received. When frames are lost, a larger window, i.e., more than the link's bandwidth*delay product, is needed to allow continuous operation while frame retransmission takes place. The modulus numbering space determines the size of the frame header sequence number field. This sequence space needs to be larger than the link window size and, if using selective repeat ARQ, larger than twice the link window size. For continuous operation, the sequence space should be larger than the product of the link capacity and the link ARQ persistence, so that in-flight frames can be identified uniquely. As with TCP, which provides sliding-window delivery across an entire end-to-end path rather than across a single link, there are a large number of variations on the basic sliding-window implementation, with increased complexity and sophistication to make them suitable for various conditions. Selective Repeat (SR), also known as Selective Reject (SREJ), and Go-Back-N, also known as Reject (REJ), are examples of ARQ techniques using protocols implementing sliding window ARQ.

3. Performance of Link layer flow control mechanisms

The key parameters to determine system performance for Stop-and-Wait, Go-Back_N and Selective Repeat ARQ protocols are the delay-bandwidth product, the frame error rate and the frame length.

Stop-and-Wait ARQ becomes inefficient when the propagation delay is much greater that the time to transmit a frame. This severe inefficiency is due to the requirement that the transmitter wait from the acknowledgement of a frame before proceeding with other transmissions. Note that the situation becomes much worse in the presence of transmission errors that trigger retransmissions. In stop-and-Wait ARQ the delay*bandwidth product can be viewed as a measure of lost opportunity in terms of transmitted bit. The inefficiency of Stop-and-Wait ARQ can be overcome by allowing the transmitter to continue sending enough frames so that the channel is kept busy while the transmitter waits for acknowledgements. For such that purpose the Go-Back-N ARQ protocol has been developed where the transmitter has a limit on the number of frames W_s that can be outstanding. W_s is chosen larger than the delay*bandwidth product to ensure that the channel can be kept busy. In channels that have high error rates the Go-Back-N ARQ is inefficient because of the need to retransmit the frame in error. A more efficient ARQ protocol called as Selective Repeat ARQ can be obtained by adding two new features: first, the receive window is made larger than one frame so that the receiver can accept frames that out of order but error free and second, the transmission mechanism is modified so that only individual frames are transmitted. Optimally our goal is that the selected ARQ protocol must deliver an error-free and ordered sequence of packets in the destination.

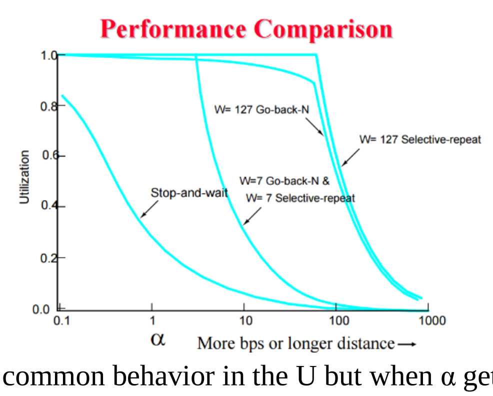
The weakness of Stop-and-Wait ARQ with respect to bandwidth*delay product can been easily seen in a higher-speed line where the efficiency quickly drops essentially to zero. The efficiency can be improved by going to larger frame sizes.

Go-Back-N and Selective Repeat ARQ are not affected explicitly by the delay-bandwidth product of a channel. Thus both of them can achieve the same efficiency $1 - n_0/n_f$ ($n_0 = \#\ of\ overhead\ bits\ , n_f = \#\ of\ bits\ in\ the\ onformation\ frame$) in the absence of errors. In details if the error rate is very low there will be only a very low number of frames being damaged and very few retransmissions of these frames which will be arrived after a damaged frame wasting a lot of bandwidth. So the performance will be almost the same with the Selective Repeat protocol which retransmits only the damaged frame due to very few damaged frames.

As it is explained in [3] we define a link parameter α = $\frac{propagation\ time}{frame\ time}$ and according to the efficiency principle For all protocols, the maximum utilization (efficiency) is a non-increasing function of α.

So utilization is defined as follows : Go-back-N ARQ : $U = \begin{cases} \frac{1-p}{1+2\alpha p}, w \geq 2a+1 \\ \frac{W(1-p)}{[(2a+1)(1-p+wp)]}, w < 2a+1 \end{cases}$

And Selective Repeat ARQ: $U = \begin{cases} 1-p, w \geq 2a+1 \\ \frac{W(1-p)}{2a+1}, w < 2a+1 \end{cases}$ and w:buffer space, window size



In case that the link is short so that the distance is small means that the propagation time is small thus a is small. If w is chosen with respect to the limitations of the protocol and w $\geq 2a+1$ then the utilization is affected only from the probability of a non-damaged frame in Selective Repeat and the same instance divided by (1+2p) (as a is small) in Go-back-N. So either for w = 7 or w = 127 the utilization of both protocols is very close to each other and very close to 1.Now in case that α is big (thus long link). At first as long as the window size $w \geq 2a+1$ both protocols have a common behavior in the U but when α gets bigger such that $w < 2a+1$ the utilization starts falling in both protocols due to limited buffer space. We evaluate it from the plot for w=7 or w =127 from where the utilization starts to decrease similarly for both protocols.

References

[1] Ross, J. F. *The Data Link Layer: Introduction, Services.* Retrieved from https://www.net.t-labs.tu-berlin.de/teaching/computer_networking/05.01.htm

[2] *RFC_3366*. Retrieved from https://tools.ietf.org/html/rfc3366#page-7

[3] Jain, R. *Data Link Control Protocols.* Retrieved from https://www.cse.wustl.edu/~jain/cse473-05/ftp/i_7dlc.pdf

[4] Bertsekas, D., & Gallager, R. *Data Networks.* Retrieved from https://web.mit.edu/dimitrib/www/Flow_Control_Data_Nets.pdf