Lab: The Hub

Can't tie the outputs of several devices together, unless they have a high-impedance state. Thus, we'll be using a hub to connect the devices. The signals coming in to the buffer are low-enable.

HDLC

Introduction

High Level Data Link Control

The predominant data link layer protocol that is used and the basis for nearly all other protocols that are in use. Another common one is X.25 that is used on packet switched networks. It is a derivative of HDLC. PPP is also based off of HDLC. LLC (Logical Link Control) is based off of HDLC.

Remember - at the data link layer, we're only concerned with transmission from node to node.

End to end vs. node to node error control.

UDP doesn't have error correction. If all your error correction is done at a lower level, then maybe you don't need it up at the higher level. This is the approach that UDP takes.

HDLC Phases

HDLC is a **connection oriented** protocol. That means that when you want to transfer data from one node to the next, a connection must be established first on each end, just like when making a protocol. Each node has to agree to certain terms. There is also a termination phase.

Establishment Phase
Data Transfer
Disconnect

A **connectionless** protocol, on the other hand, does the data transfers but does not have any sort of establishment or termination phases. It would just discard bad frames. The only place these bad frames would be discovered would be at the other end.

When HDLC was first developed, it was intended at handling point-to-point connections and keeping the errors down.

Node Types

Primary – Acts as the master device.
Secondary – Can't initiate a data transfer. Acts a slave device.
Combined – Can act as either a primary or secondary node. This is more typically used now a days. They can initiate data transfers and usually work in full-duplex mode.

Link Configurations

Asynchronous Balanced Mode (ABM) – typically used
All nodes are combined nodes

There is some handshaking that takes place when the link is established.

Asynchronous Balanced Mode Extended (ABME)

Frame Types

I-Frames (Information)
Actually carry the data

S-Frames (Supervisory)
Receiver ready, ACK frames, etc.

U-Frames (Unnumbered)
Used for link establishment and termination. Just used for managing the data link.

Flag
01111110

Frames have this flag at the beginning and the end.

Bit Stuffing
Transmitting device looks at the data it is sending. It knows when it is sending data and when it is in the middle of sending data. If the information field has the same pattern, after 1 consecutive 1s, it will insert a 0 right after the 5^{th} one. That will keep us from getting more than 5 1s in a row unless the transmitter wants to have more than 5 1s in a row. When the receiver receives this data, after seeing 5-1s, it will check the next bit. If it is 1, then it knows it is the flag. If it is a 0 it simply removes it. This removes restrictions on the data that gets sent.

Control Field of I-Frame

When the first bit is 0, it identifies the frame as being an I-Frame.
N(S) - The next three bits specify the sequence number. However, this can be negotiated to be 6. It is agreed upon at the time the link is established.
P/F bit – only means something if it is a 1. When set to a 1, it specifies the frame as a polling frame. It is only needed when the transmitter's timer times out and it is prompting the receiver to let it know what to send next.
N(R) – Used to specify which frame is expected next.

Can be used for **piggybacking –** ACK frames already received while also sending data back.