Lecture 18: Physical & Data Link Layers

Adam Hawley

April 1, 2019

Contents

1 Physical Layer

Deals with the transmission of bits over physical medium (twisted pair/coaxial cables, fibre optics, wireless etc.). The layers usually have a standardised interface to transmission media (e.g. how 1/0 signals are carried over a link).

Requirements of the physical layer include both **synchronisation** and **flow control** as well as multiplexing (FDM (frequency division multiplexing), TDM (time division multiplexing) and CDM (code division multiplexing)). Multiplexing is where you use the same connection for multiple transmissions.

2 Data Link Layer

Deals with the framing of raw data, as well as flow control and error correction (i.e. if they cannot be solved at the physical level). The data linke layer is often divided into two sublayers:

- **Logical Link Control (LLC)**: Multiplex protocols running over the data link layer, error and flow control.
- **Media Access Control (MAC)**: Control channel access, append and check FCS (frame chack sequence), discard malformed frames, addressing.

2.1 Example: Ethernet

- Standard: IEEE 802.3

Uses CSMA/CD local area network technology:

- Multiple-Access (MA): Several nodes are connected to the same cable (cf. data bus).
- Carrier Sense (CS): A node can distinguish between a busy and an idle link.
- Collision Detect (CD): A node listens when it transmits a frame in order to detect whether the frame interferes (collides) with a frame transmitted by another node.

Multiple Ethernet segments are joined by **repeaters** that forward the signals. In an Ethernet, every signal is propagated in all directions over the entire network, even crossing repeater boundaries.

- Problem: All hosts compete for access to the same link; they are said to be in the same **collision detection**.

The problem is solved by intelligently partitioning the **collision domain** using the following:

Hub Multiway repeater supporting several point-to-point segments - still only one collision domain.

Bridge Each port is connected to a different collision domain. Transmissions within separate domains are allowed to happen in parallel.

Switch Frames are sent only to their destinations — no collisions.

How can we know that a frame is going between certain nodes?

Each Ethernet adaptor has a unique Ethernet (MAC) address, which is 6 bytes long and read as a sequence of six numbers, each given as a pair

of hexadecimal digits (e.g. 08:00:2B:E4:B1:02). Each connected adaptor receives all frames transmitted over an Ethernet, but passes to the upper protocol layers only those matching:

- The adaptor's own address (unicase)
- The Ethernet broadcast address (FF:FF:FF:FF:FF:FF).
- Multicase addresses (not used in practice just in the standard) (least significant bit of the first byte set).

See lecture for frame format example.

Other Data Link Layer protocols include:

- IEEE 802.11 - Wireless Lan
- IEEE 802.16 - Wireless Broadband (e.g. WiMax)
- IEEE 802.15.4 - Low-rate Personal Area Networks (e.g. ZigBee, WirelessHart)

3 Learning Bridges

How does a bridge know to which port to forward a packet, i.e. on which port a destination host resides?

A bridge inspects the source address of each received frame and notes the number of the incoming port. The pair </port number/, /source address/> is added to the bridge's forwarding table. If there exists no entry for a destination address (yet), the packet is forwarded to all ports, except the packet's incoming port.

Learning algorithm:

1. The forwarding table is empty when the bridge boots.
2. The forwarding table it updated according to the above scheme.
3. Table entries are discarded after some amount of time to protect against the possibility of LAN addresses changing segments.

The table is updated when messages are sent as even though they cannot know which domain the receiver is in, they can know which the sender is.

4 Problems with Loops in Extended LANs

Loops in extended LANs:

- Provide **redundancy** in case of failures of links/bridges.
- Potentially enable frames to **loop forever**.

To stop them looping forever, a **distributed spanning tree algorithm** is run, which selects, for each bridge, the ports over which it should forward frames such that loops are avoided.

4.1 Spanning Tree

Think of an extended LAN (with loops) as a graph (with cycles). The **spanning tree** of a graph is a **subgraph** that emerges from the original graph by leaving out edges such that no cycles/loops remain. There is a spanning tree algorithm included in the IEEE 802.1 specification for LAN bridges. Each bridge decides the ports over which it is (not) willing to forward frames.

4.2 Distributed Spanning Tree Algorithm

4.2.1 Basic Idea

1. The algorithm first elects the bridge with the smallest id (address) as the **root** of the spanning tree.
2. Each bridge computes the **shortest path** to the root and notes its ports that lie on this path (*preferred port towards root*)/
3. Each LAN elects a single designated bridge (one of the closest to the root — smallest id wins in tie) which is made responsible for forwarding frames towards the root bridge.

Afterwards each bridge just forwards frames over those ports (i.e. to those LANs) for which it is the designated bridge.

Implemented by bridges exchanging configuration messages with each other; these messages include:

id For the bridge that the sending bridge believes to be the **root**.

distance (measured in hops) from the sending to the root bridge.

id for the bridge sending the message.

Each bridge:

- Initially thinks that it is the root and sends over all of its ports the message (id,0,id), where id is the bridge's identifier.
- May receive a message over one of its ports and checks whether:
  - It identifies a root with a smaller id.
  - It identifies a root with an equal id but with shorter distance.
  - Root id and distance are equal but the sending bridge has a smaller id.
- If so, it adds 1 to the distance, saves this info & discards old info.

See slides ror example.