**Takeaways for Week 2**

Unresolved*Thread controls*

[Subscribe for email updates.](javascript:void(0);)

* Sort replies by:
* [Oldest first](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=791&sort=oldest)
* [Newest first](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=791&sort=newest)
* [Most popular](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=791&sort=popular)

**Tags**

* [takeaway](https://class.coursera.org/comnetworks-003/forum/tag?id=403) [Delete Tag×](javascript:void(0))

[+ Add Tag](javascript:void(0))

[**[https://coursera-profile-photos.s3.amazonaws.com/df/780fe9a7e8c6457121a8d867ec29af/logo_klein.png](https://class.coursera.org/comnetworks-003/forum/profile?user_id=3067255)Artur Neumann**](https://class.coursera.org/comnetworks-003/forum/profile?user_id=3067255)**·** **[a month ago](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=791" \l "post-2226)**

**Just wanna share with you what I take from week 2:**

1. Transmission delay (time to put the message on the wire) = M (bits) / Rate (bits/sec)
2. Propagation delay (time the signal needs to travel trough the media) = Length / speed of signals
3. Latency = Transmission delay + Propagation delay
4. Bandwidth - Delay Product (amount of data in flight) = Rate x Delay
5. Signals "act" different depending on the media (fiber, cable, wireless)
6. less frequency = less bandwidth
7. What happens to a signal:   
   a) delayed (need time to travel)   
   b) attenuated (it arrives less strong than it was send)   
   c) different frequencies are attenuated differently (   
   Frequencies above a cutoff are highly attenuated)   
   d) noise is added to the signal
8. wireless is complex (interference, high attenuation)
9. different type of modulation (Non-return-to-zero, different levels)
10. clocks need to be in sync. different technic can be used:   
    a) scrambling   
    b) 4B/5B - every 4 bit are mapped to a 5bit value to avoid long runs of 0   
    c) NRZI (a 1 is represented by a invention/change) so log runs of 1s are not a problem for the clock
11. If we cannot change the baseband frequency for caring data, we use always the same baseband frequency but change:   
    a) Amplitude   
    b) Frequency (inside the borders of the baseband?)   
    c) Phase (up-down vs. down-up)
12. The maximum rate on a link can be calculated. without noise the Nyquist Limit is used with noise the Shannon Capacity.
13. On the link-layer data has to be framed. We need to know what the start an end of the frame is. For that we can use:  
    a) bitcount (unreliable)  
    b) byte stuffing (set a flag at the besieging and end of frame - escape the flag if its in the payload - escape the escape if its in the payload)  
    c) bit stuffing (the start/end flag is a number of 1s - if there is the same number of 1s in the payload put a 0 after it - e.g. after 5x1 put a 0)  
    d) PPP uses clever way of bytestuffing (0x7E is the FLAG and 0x7D the ESC -  to stuff/unstuff a byte, add/remove ESC, and XOR the byte you like to escape with 0x20 - with this system there will be never a FLAG in the payload)
14. Error detection and correction is difficult because errors can appear in the payload but also in the checks
15. Hamming Distance is the amount of bits that need to be flipped to change one correct codeword (e.g. checksum) to an other, so the result looks correct even if there was an error.
16. For detection x errors we need a Hamming distance x+1
17. For correction x errors we need a Hamming distance 2x+1
18. Error detection algorithms:  
    a) parity bit - 1 extra bit that is modulo 2 of the data (simple, Hamming distance 2, odd amount of errors will be detected)  
    b) Checksum - e.g. Internet Checksum (more complex than parity but sill simple to calculate, Hamming distance 2, burst errors up to 16bits will be detected, random errors will be detected with high probability )   
    c) CRC (complexer mathematical algorithms, exact protection depends on details)
19. Error correction can be done by different algorithm. e.g:  
    a) Hamming Code: check bits in positions that are powers of 2, start with 1. Check bit in position p is parity of positions with a p term in their values. Hamming distance = 3  
    b) LDPC - Low Density Parity Check
20. It depends on the pattern of the expected errors if a correction or detection is better.  
    a) correction is better when errors are expected (e.g wireless) or when there is no time for retransmission  
    b) detection is better when errors are not expected or large when they occur.

**Anything wrong or important I missed?**  
**Any ideas how to express the stuff better?**

12votes received.

[· flag](https://class.coursera.org/comnetworks-003/help/flag?url=https%3A%2F%2Fclass.coursera.org%2Fcomnetworks-003%2Fforum%2Fthread%3Fthread_id%3D791%23post-2226)

[**Michael Peterson**](https://class.coursera.org/comnetworks-003/forum/profile?user_id=1342108)**·** **[a month ago](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=791" \l "post-2388)**

By "scrambling" in 10a, do you mean XOR'ing of the bits?

1votes received.

[· flag](https://class.coursera.org/comnetworks-003/help/flag?url=https%3A%2F%2Fclass.coursera.org%2Fcomnetworks-003%2Fforum%2Fthread%3Fthread_id%3D791%23post-2388)

[+ Comment](javascript:void(0);)

[**[https://coursera-profile-photos.s3.amazonaws.com/df/780fe9a7e8c6457121a8d867ec29af/logo_klein.png](https://class.coursera.org/comnetworks-003/forum/profile?user_id=3067255)Artur Neumann**](https://class.coursera.org/comnetworks-003/forum/profile?user_id=3067255)**·** **[a month ago](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=791" \l "post-2398)**

Its not explained in detail but the Prof says:  
"you could exclusive all your data with the pseudo random signal, which makes it higher luckily that you get transitions."  
The textbook should have more details in §2.5.1

**Takeaways for Week 3**

Unresolved*Thread controls*

[Subscribe for email updates.](javascript:void(0);)

* Sort replies by:
* [Oldest first](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=991&sort=oldest)
* [Newest first](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=991&sort=newest)
* [Most popular](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=991&sort=popular)

**Tags**

* [takeaway](https://class.coursera.org/comnetworks-003/forum/tag?id=403) [Delete Tag×](javascript:void(0))

[+ Add Tag](javascript:void(0))

[**[https://coursera-profile-photos.s3.amazonaws.com/df/780fe9a7e8c6457121a8d867ec29af/logo_klein.png](https://class.coursera.org/comnetworks-003/forum/profile?user_id=3067255)Artur Neumann**](https://class.coursera.org/comnetworks-003/forum/profile?user_id=3067255)**·** **[a month ago](https://class.coursera.org/comnetworks-003/forum/thread?thread_id=991" \l "post-2835)**

**Here my personal takeaways for Week 3, publishing them in hope they can be useful for somebody**

1. Reliability is implemented in every layer of the network stack. The lower layer try to mask the errors and the higher to run recover actions if an error occurs
2. In situations when errors are common ARQ (Automatic Repeat reQuest) is used to resend a corrupted message
3. ARQ rules:  
   a) receiver acknowledges (ACK) every correct frame  
   b) sender resends a frame if it didn't receive an ACK within a defined time
4. ARQ timeouts in a LAN are choosen by the worst case (what is the maximum signal delay)
5. for avoiding dublicates a sequence number is used  
   a) if there can be just one frame in flight a flag is enough (0,1) - Stop-and-Wait. That could limit the performance  
   b) for having more than one frame in flight a sliding window is used, longer sequence numbers are needed but allows to have more frame to be outstanding
6. Classic ways for multiplexing (sharing a link) - this ways are good for continuous traffic and fixed number of users, but not for bursty network traffic  
   a) Time Division Multiplexing (TDM) - every device gets a defined time when it can send frames  
   b) Frequency Division Multiplexing (FDM) - every device gets its own frequency
7. Multiplexing for bursty traffic  
   a) Randomized Multiple Access. Nodes randomize their resource access attempts (Good for low load situations)  
   b) Contention-free. Nodes order their resource access attempts (Good for high load or guaranteed quality of service situations)
8. Randomized Multiple Access Protocols:  
   a) ALOHA Protocol - everybody sends when they need to send. If there is not ACK, wait a random time and resend. Simple, but not efficient under high load.  
   b) CSMA Carrier Sense Multiple Access. Like ALOHA, but listen before sending.  
   c) CSMA/CD (with Collision Detection). Sending a colision signal and aborting when a colision is detected. In worst case it can take up to 2Delay seconds till a host hears the colision signal. Ethernet does not send ACK, thats why it needs to have a frame size that last min. for 2D seconds (64bytes) to know if its arrived without colisions.  
   d) Binary Exponential Backoff (BEB). When nodes listen and wait to send they que up. If they start sending imediately after the medium is free a collision will happen and end up with a deadlock if everybody sends again after a collision.
9. Rules for BEB:  
   a) After the first collision the host waits 0 or 1 frametime (randomly) before it tries to send again  
   b) after the second collision it waits 0 to 3 frametimes  
   c) doubles this random interval till it can send its frame successfully
10. Classic Ethernet (IEEE 802.3)  
    a) uses 1-persistent CSMA/CD with BEB - 1-persistent =>if someone else is sending, wait for a moment and then they send right away  
    b) addresses for sender and receiver  
    c) CRC-32 for error detection  
    d) no ACK  
    e) no retransmission  
    d) Start of frame is identified with a physical layer preamble
11. CSMA/CD does not work for wireless because:  
    a) different area of coverage - not every node can hear every other node  
    b) node cannot hear while sending because the receiving signal is much weeker that the sending signal
12. MACA (handshake solution) is used for wireless instead of CSMA  
    Rules:  
    a) A sender node transmits a RTS (Request-To-Send, with frame length)  
    b) The receiver replies with a CTS (Clear-To-Send, with frame length)  
    c) Sender transmits the frame while nodes hearing the CTS stay silent
13. 802.11 CSMA/CA - Sender avoids collisions by inserting small random  gaps before it start to send
14. 802.11 Physical layer:  
    a) uses 10 or 40MHz channels in 2.4GHz and 5GHz band (depends on flavor)  
    b) uses OFDM what is a kind of FDM  
    c) different amplitudes / phases for varying SNRs => different rates (6 to 54Mbps)
15. 802.11 Link Layer  
    a) uses CSMA/CA and RTS/CTS optional  
    b) ACK of frames and retransmitting with ARQ  
    c) 3 addresses: soure, transmitter, destination  
    d) 32bit CRC for error detection
16. Under load Random Multiple Access is not efficient, here are Turn-Taking Multiple Access Protocols better:  
    a) Token Ring - all nodes are connected in a ring. A Token is passed from node to node. The node that receives the tocken is allowed to send a frame  
    - good under higher load  
    - easier to give guaranteed quality of service  
    - complex  
    - uneffcient under low load  
    b) node addresses
17. In an Hub all ports are wired together, inside the same system as classic eathernet
18. switches provide a direct conenction between ports without sending data out of the other ports and a small buffer
19. Backward Learning  
    a) if a Switch does not know to what port a particular address is connected to it sends data with that address as destination to all ports  
    b) everytime a host sends data the switch reads its address and knows from now on that this address is connected to this port  
    c) that system works also over multiple switches
20. Spanning Tree Protocol  
    a) without Spanning Tree loops and double pathes causes problems in Ethernet  
    b) the Spanning Tree Protocol avoid the problem of loops by finding a subset of the network that is a tree without loops  
    c) usual networks can have different possible trees  
    d) no central coordination instance  
    e) start with no information -> no configuration needed  
    f) adapts to changes / failures
21. Spanning Tree Algorithm  
    a) Every Switch believes in the beginning it is the root  
    b) Each switch sends periodic updates to neighbours with: its address, address of the root, and distance (in hops) to root  
    c) When a Switch gets a message from an other it decides based on the address who is really root (lowest address is root)  
    d) the tree grows as shortest distances from the root. If two ways have the same distance the lower address is preferred  
    e) ports that are not in the tree are turned off  
    f) after the tree is grown the algorithm continues to run and adapts to changes by timing out information

**I found the week 5 topics not so easy. Here my takeaways. Hope they help somebody and I will use them myself during homeworks :-)**

1. Routing vs. Forwarding
   * Routing is deciding which way to send the packet
   * Forwarding is sending the packet
2. With Routing we would like to find the "best" way for a packet if multiple ways are possible
3. Properties for routing
   * Correctness - Finds paths that work
   * Efficient paths - Uses network bandwidth well
   * Fair paths - Doesn't starve any nodes
   * Fast convergence - Recovers quickly after changes
   * Scalability - Works well as network grows large
4. Rules of Routing Algorithms
   * no central controller
   * odes only know what they learn by exchanging messages with neighbors
   * nodes work concurrently
   * nodes, links or message failures can occur
5. Shortest Paths
   * tries to find the best path according to the "costs"
   * Algorithm:  
     1. assign each link a cost (distance)  
     2. Define best path between each pair of nodes as the path that has the lowest total cost (or is shortest)  
     3. Pick randomly to any break ties
   * Subpaths of shortest paths are also shortest paths
6. Sink tree / Source tree
   * Sink tree for a destination is the union of all shortest paths towards the destination
   * Source tree is the opposite
   * Each node only need to send frames to the next hop
   * Only need to use destination to follow shortest paths
7. Dijkstra's Algorithm
   * Gives complete source/sink tree
   * grows fast
   * Algorithm:  
     1. Mark all nodes tentative  
     2. set distance from source to source to 0  
     3. set distance to every other note to infinity - we don't know yet how to reach them  
     While tentative nodes remain:  
     1. find N, the known node with the lowest distance (at the very beginning its the source node, that has distance 0 and that's the only information we know)  
     2. Add link to N to the shortest path tree  
     3. Check all neighbours of N and lower their distance (to source) if any better distance was found (relaxation).
8. Distance Vector Routing
   * Each node maintains a table (vector) with distances and next hops to all destinations
   * Algorithm:  
     1. Initialize the cost to self with 0 and with infinite to every other host  
     2. Periodically send own information to neighbours  
     3. Update the table for each destination by selecting the shortest distance heard, after adding cost of neighbour link
9. Flooding
   * Broadcasting a message to all hosts in the network
   * Sends an incoming message on to all other neighbours
   * Remember the message so that it is only flood once (we don't want to send messages in loops) - sequence numbers are used for that
10. Link-State Routing
    * Each node floods link state packet (LSP) that describes their portion of the topology - basically the costs to their neighbours
    * Each node computes its own forwarding table by running Dijkstra
    * The forwarding table contains the destination and the next hop to it
    * On changes updated LSPs are flooded
11. Equal-Cost Multi-Path Routing
    * we would like to use multiple shortest path
    * we build not a source/sink tree but a directed acyclic graph (DAG), what is the same but with possible concurrent connections
    * uses Dijkstra algorithm but remembers a set of next hops
12. Hierarchical Routing
    * Route to a region not to a individual host
    * routing tables contain regions and not single hosts
    * result into shorter tables
    * can result in longer path, compared to having a full host tables
    * is done with IP prefix aggregation - join multiple more specific prefixes into one large prefix
13. Routing with Multiple Parties
    * Every party in a network makes their own routing decision according to their own policies
    * can cause asymmetric links (packets take one path from A to B but an other from B to A)
    * the path taken between 2 hosts might not be the overall shortest
14. TRANSIT and PEER service
    * ISPs give TRANSIT service to customers -> traffic from the customer is routed to the rest of the internet and visavers
    * customers pay for TRANSIT service
    * ISPs give PEER service to each other -> Each ISP accepts traffic from the other ISP only for their customers
    * PEER service is usually free of cost -> ISPs do each other a favour
15. BGP (Border Gateway Protocol)
    * computes routes between big networks (ISP, enterprise, etc.)
    * different parties are called AS (Autonomous Systems)
    * Border routers of ASes announce BGP routes to each other
    * Route announcements contain an IP prefix, a list of ASes on the way to the prefix (path vector) and the
    * the path vector is needed to find loops
    * Border routers of ISP announce paths only to parties who may use those paths
    * Border routers of ISP select the best path of the ones they hear