%%%%%%%%%%%%%%%%%%%%%
% H1
%%%%%%%%%%%%%%%%%%%%%

Network purpose: Transfer information from A to B

HIERARCHY IN NETWORKING

Access	----	edge	---	core	---	edge	---	access
almost no aggregation	----	less aggregation	---	more aggregation	---	less aggregation	---	almost no aggregation

Access: ADSL, GSM etc.
Core: Optical backbone network

BASIC ADDRESS TYPES
- -unicast: one-to-one communication
- -multicast: one-to-many communication
- -anycast: one-to-nearest communication
- -broadcast: one-to-all

CONNECTION ORIENTED (CO) AND CONNECTIONLESS (CL) FORWARDING
CO:
- -all packets of a same flow follow the same path
- -example: telephone network
- -advantage: capacity or amount of resources can be guaranteed
- -disadvantage: circuit costs are fixed, higher blocking probability

CL:
- -all possible paths from source to destination can be followed
- -example: computer networks

OPEN SYSTEM INTERCONNECT (OSI)
- -communication slack between two parties
- -horizontal dimension: communication path from one end system to the other over the network
- -vertical dimension: communication is structured in 7 layers:
  1. Physical — Please
  2. Data Link — Do
  3. Network — Not
  4. Transport — Throw
  5. Session — Salami
  6. Presentation — Pizza
  7. Application — Away
- 4 to 7: end-to-end protocols
- 1 to 3: chained or hop-by-hop protocols
- -each layer has well-defined function and is independent from other layers but has interfaces with other layers

	Postal service example:
-SDU: Service Data Unit	letter
-PDU: Protocol Data Unit	mail bag
-SAP: Service Acces Point	post office

-protocol header letter format with name, address and space for stamp

-the lower the layer, the longer the information containers grow
-encapsulation: placing of control and data part of higher layer into data part of lower layer
-Piggybacking in Data Link layer (?)

-Service: defines what the layer does
-Interface: tells the processes above it how to access the service, parameter specification
-Protocol: set of rules for communication between peers
-layer can use any protocol to provide its service

Pros & cons of layering
Pro: simplify design
-divide complex problem into smaller pieces (independent & parallel execution)
-hiding implementation details from other layers: easy to upgrade part of the system
-re-use of functionality: many upper layers can share services of lower layers
Con: poor performance
-limited info exchange between layers

TCP/IP
-Hourglass design

7. Application :	Telnet FTP DNS HTTP SMTP
4. Transport:	TCP UDP RTP
3. Internet:	IP
2&1. Host-to-network:	SDH FR LAN ATM ISDN

-Less layers
-less interconnects because of only one layer 3 protocol

%%%%%%%%%%%%%%%%%%%%
% H2
%%%%%%%%%%%%%%%%%%%%
Local Area Network (LAN)

IEEE 802.3: Carrier Sense Multiple Access (CSMA)
-called Ethernets in short

Ethernet
-Bus (ether): all stations share a single communication channel
-Broadcast
-all transceivers receive every transmission
-host interface filters among packets those intend for the correspondig computer
-Best-effort delivery: no notification about packet receipt

Multiple Access Communication
-Centralized (master-slave)
-Simple network
-Circuit mode: GSM
-Packet mode: Polling reservations

-Distributed (all stations are peers)
	-Robust network
	-Very scalable
	-Circuit mode (?)
	-Packet mode: token passing
		-ALOHA (large delay)
		-CSMA (small delay)

Ethernet Access: CSMA/CD
-Distributed access: no central authority
-Carrier Sense Multiple Access (CSMA)
	-Multiple machines can access the Ethernet simultaneously
	-each machine determines whether the 'ether' is free by sensing carrier wave propagation
	-each transmission is limited in duration to prevent monopolization of the network
-Collision Detection (CD)
	-Collision: when two electrical waves cross, they become scrambled and meaningless
	-Collision detection: each transceiver monitors the cable while transmitting to search for foreign signal interferences
-Maximum collision detection time: 2t with t the maximum propagation delay in the broadcast network

Sending rules
-Non-persistent CSMA
	-sensing is not continuously but repeated after random time
	-If no collisions are sensed, the station sends a packet
-Persistent CSMA
	-continuously sensing, but doens't send packet immediately
	-p-persistent CSMA: send in current time slot with probobility p and in another time slot with probability 1-p
	-Binary exponential back-off policy: a sender delays a random time after the first collision, twice that time after the second and four times as long after the third collision and so on

p-persistent CSMA/CD
-suppose N stations, every station transmits with probability p in any given slot
-Pr[s]= Np(1-p)^N-1
-maximum value of Pr[S]: dPr[S]/dp = 0 --> p - 1/N
	-N --> inf: max(Pr[S]) = 1/e ~= 0.368

Binary exponential backoff
-after each collision j, a station chooses a random time uniformly in [0,(2^j)-1] timeslots:
	-1st: 0 or 1
	-2nd: 0 or 1 or 2 or 3
	-3rd: 0 or 1 or 2 or 3 or 4 or 5 or 6 or 7
	-increase until 10th
	-then contsant until 16th
	-then failure
-balances between prevention of collisions and waiting time

Ethernet MAC protocol:

-1-persistent CSMA/CD with binary exponential backoff
1. wait until channel idle
2. when idle: transmit and keep listening
3. when collision:
-stop transmission
-send jam signal
-wait random time
-go to 1.

Approximate performance analysis of CSMA/CD
-Efficiency of CSMA/CD is defined as

$$n\_CSMA/CD = \frac{E[T]}{E[T] + E[delta]}$$

with E[T]: average total time needed for the protocol in the heavy traffic regime
and E[delta] = 1/p_s-1 <= e-1 (p_s probability of succes)
-also:

$$n = \frac{1}{1 + v*a} \quad \text{where } a = \frac{t}{E[T]} \text{ and } v = 2(1/p\_s - 1)$$

$$n = \frac{1}{1 + f(v)BD/E[L]} \quad \text{with } f(v) = v/(0.8c)$$

L: length of frame (bits)
B: capacity (bits/s)
D: extent of LAN (m)
c: velocity of light

-maximum value of v = 3.5, but from simulations: v = 5

Ethernet address
-48-bit IEEE 802 MAC address:
-24-bit company/manufacturer ID
-24-bit extension/board ID
-Broadcast address
-IEEE Extended Unique Identifier 64 (EUI-64) address:
-24-bit company ID
-40-bit extension ID
-probably larger because of growth in number of devices

%%%%%%%%%%%%%%%%%%%%%%
% H3
%%%%%%%%%%%%%%%%%%%%%%

Phenomena that may cause errors:
-thermal/impulse noise
-signal distortion
-crosstalk
-echoes and reflections
-fading

-synchronization errors
-quantization noise

What to do when an error is detected:
-try to correct errors in the packet
    -Forward Error correction (FEC): packet contians enough redundant information to be able to correct errors
        -used in real-time applications (video, telephony etc.)
-discarding the packet
    -used for non-real-time applications, UDP
-retransmission of the packet
    -used in ARQ/TCP

Principle of error detection
-message M: k bits
-operation O on M gives result R (n-k bits)
-sender constructs codeword C consisting of R and M, C =(M,R)
-receiver checks received codeword C* = (M*,R*) by operation O on M* which gives R**
-if (R* = R**)
    -then receiver assumes that M* = M and R* = R: error free transmission (not sure)
    -else transmisson errors (sure)

-operation O should be fast to compute
-result R should contain only a small number of bits
-check should give sufficient guarantee that it's very likely that M* = M if R* = R

Single parity check
-operation O: sum of the k bits of M modulo 2
    -result R is either 0 or 1
-transmit R

-receiver:
    -compute sum R* of received bits modulo 2
        -if R* = 0, no errors
        -else error(s)
-single bit error is detected

-Hamming distance: number of bits of two vectors which differ from each other
    -minimum Hamming distance d =2 because each message word differs at least by 1 bit in which case the parity bit is also different by 1 bit
    -single parity check code can detect d-1 = 1 error but cannot correct errors because (d-1)/2 = 0
    -to detect d errors: distance >= d + 1
    -to correct d errors: distance >= 2d + 1

-Checksum computations in the Internet
    -M consists of k 16-bit words
    -R consists of a 16-bit checksum
        R = -sum(M_j(mod(2^16-1))) from j=0 to j=k-1
    -transmitted header is multiple 16-bit codeword C which obeys C(mod(2^16-1)) = 0

-Cyclic redundancy check (CRC)
    -M & R binary numbers

    -C = M*2^r + R
    -M*2^r = R(mod G), G is generator, fixed number which both sender and receiver agree upon, must consist of at least r + 1 bits
    -C* = M* * 2^r + R* = C + F, F is error pattern bit string
        -1 in F at position j reflects an error in bit string C* at position j
        -F is a multiple of G

Principles of acknowledgment schemes
-goal: implement reliable packet channel over an unreliable one
-approach: use timers, acknowledgments & retransmits
-sender sends packet, when acknowledgment ACK from the receiver does not arrive in time, the packet is retransmitted
-sequence numbers are assigned to the packets to avoid duplication

-Retransmission protocols: Automatic Repeat Request (ARQ), three variants in increasing order of complexity:
    -stop-and-wait
    -go back n
    -selective repeat

    -efficiency: maximum average rate (throughput)/link rate

-Stop-and-wait protocol
    -packet is retransmitted if the ack of the receiver has not arrived in the worst-case end-to-end delay
    -one-bit sequence number to avoid duplicates
        -sequence bit for sender en receiver must be equal, otherwise errors occurred
    -sometimes called Alternating Bit Protocol
    -disadvantage: it does not operate always correctly in networks where packets may follow different paths
    -fails in networks that serve packets in a non-FIFO (first-in-first-out) order.
    -hardly used in practice
    -round trip time T = t_packet +t_ack + 2(t_prop + t_proc)

    -t_packet = l_packet(bits)/C(bits/s)
    -t_ack = l_ack(bits)/C(bits/s)

    -maximum effective information rate of S&W protocol is
        -Rs&w = (l_packet - l_header)/2

$$\text{-efficiency of S\&W protocol: } n_{S\&W} = \frac{1 - \frac{l_{header}}{l_{packet}}}{1 + \frac{l_{ack}}{l_{packet}} + \frac{2C(t_{prop} + t_{proc})}{l_{packet}}}$$

    -n_S&W;p <= (1-p)n_S&W
        -p probability that a packet or ack contains errors
        -blz. 65&66 voor meer uitleg

    -average transmission time E[T_S&W;p] = T + E[delta]T_RTO = T + p*T_RTO/(1-p) >= T/(1-p)

-Concept of the sliding window
    -allows the sender to transmit multiple packets before awaiting an ack
    -all packets in a window are transmitted, ack makes window shift one place
    -pipelining, the processing of a new task is started before the completion of a previous task
    -protocols using sliding window strategy:
        -Go Back n
        -Selective Repeat

-Go Back n
    -only the next in order packet is aggepted and acknowledged by the receiver
    -when a packet gets corrupted, all arriving packets after are discarded
        - packet transmission process must go n packets back, n is window size W
    -sender maintains two sequence numbers
        -n_S;0 sequence number of the first unacked transmitted packet
        -n_S sequence number of most recently transmitted packet for which n_S >= n_S;0
        -largest possible number for n_S = n_S;0 + W_s -1 (W_s window size)
        -if received packet is error-free but possesses a sequence number different from n_R, the packet is discarded and an ack with seq.no. n_R is transmitted

    -average transmission time E[T_gbn;p] = t_packet + W_s * t_packet * E[delta] = t_packet*(1 + (W_s*p)/(1-p))

$$\text{-efficiency } n_{gbn;p} = (R_{gbn;p})/C = (1 - p) * \frac{1 - \frac{l_{header}}{l_{packet}}}{1 + p(W_s - 1)}$$

-Selective Repeat protocol
    -improves efficiency of Go Back n by changing two features:
        -window size at the receiver is not limited to one packet
        -only individual packets are retransmitted
    -operates best with a large window size at both receiver and sender
    -packets received after a corrupt packet are buffered and only passed to the higher layer if retransmission of lost packet is succesful

    -average transmission time E[T_SR;p] = t_packet + E[delta]*t_packet = t_packet/(1-p)

    -efficiency n_SR;p = (l_packet - l_header)/(C*E[T_SR;p) = (1 - p)(1 - l_header/l_packet)

Comparison of ARQ protocols
-Go Back n, used in situations with:
    -limited buffer space
    -enough channel capacity

-Selective Repeat, used in situations with:
    -enough buffer space
    -enough computational power

-efficiency
    -n_S&W;p = (1-p)/(1+Lc)
    -n_gbn;p = (1-p)/(1+p*Lc)
    -n_SR;p = 1-p
        -link capacity Lc

%%%%%%%%%%%%%%%%%%%%
% H4
%%%%%%%%%%%%%%%%%%%%
Architectural principles
-Slogans:
        -goal: connectivity
        -method: inter-networking layer
        -tool: the Internet Protocol (IP)
        -intelligence: end-to-end rather than hidden in the network
        -faith: all networks are equal
        -freedom: nobody owns the Internet
-Layers:
        -L1: Transmission & access
        -L2: Switching
        -L3: Routing
-Router: a network element that provides internetworking, based on one common protocol IP between different networks

Internet Protocol (IP)
-defines an unreliable, connectionless, best-effort delivery mechanism
        -specification of basic unit of transfer (packet)
        -IP software performs the routing function
        -IP includes the rules for unreliable, connectionless, best-effort delivery

IPv4 packet
-first 'layer'
        -vers (4): version of IP protocol used. For IPv4: vers = 0100
        -hlen (4): header length, datagram header length in 32-bit words
        -ToS (8): type of service. Designed to allow the router to find best matching path for a service defined by requirements (4 bits) for delay, throughput, reliability and cost and a priority level (3 bits). This field is mostly ignored nowadays
        -total length (16): length in octets of of complete information container including header and data. Maximum length 65535 bytes
-Fragmentation 'layer'
        -TCP/IP software chooses initial datagram size and arranges the split-up of large datagrams into smaller pieces
        -identification (16): unique integer that identifies the datagram
        -flags (3): bits that control fragmentation (e.g. first bit = 1: do not fragment)
        -fragment offset (13): offset measured in octets, typical one fragment size
-Other 'layers'

-TTL (8): time to live, specifies how long in seconds the datagram is allowed to remain in the Internet. Measured in 'hops', default TTL set by source is 32 hops.
-protocol (8): specifies upper-layer protocol that createt the message or that is to receive the IP data at the destination host. e.g. 6 = TCP, 17 = UDP and 1 = ICMPs
-header checksum (16):ensures integrity of the IP header. If checksum fails, IP packet is discarded
-options (24): allows the packet to request optional features such as specific path from source to destination, timestamp and router alert
-padding (8): zero padding to multiple of 32 bits words

-Tunneling: similar to encapsulation (?), but an entire L3 packet is enclosed into the data field of another L3 packet: IP header refers to IP
-used where not all nodes understand a newer or advanced protocol

IP addressing (IPv4)
-IPv4 addresses consist of 32 bit or 4 bytes denoted as x.y.z.w in range [0, 255]
-address not related to geographical place: logical binding
-Netid and hostid

-5 address classes, A B C D E with identifier bits 0, 10, 110, 1110 and 11110
-A: netid (7) hostid (24)
-B: netid (14) hostid (16)
-C: netid(21) hostid(8)
-D: multicast address (28)
-E: reserved for future use
Observe that the number of possible netids grows and hostids shrinks

Subnet addressing
-enables large network to be split into several smaller subnetworks
-subnet mask: 1s for every position except for the new hostid
-subnet address = IP address AND subnet mask (bitwise)
-subnet: x.y.z.w/v where v is an integer smaller than 32

Classless Inter-Domain Routing (CIDR)
-possibility to construct an arbitrary length of the netid
-supernetting: allows a host to aggregate multiple class C-addresses to obtain an address space larger than that of a class C subnetwork but smaller than that of a class B
-advantages:
-single routing entry corresponding to the supernet is needed instead of 8 entries for each class C subnet
-copes with the shortage problem of IP addresses
-Consists of:
-21 bit CIDR netid
-8 bit hostid
-3 bit block id (for 8 C class blocks)

Domain Name System (DNS)
-translated names of hosts into IP adresses
-hierarchy in domains, e.g. ewi.tudelft.nl: .nl > tudelft.nl > ewi.tudelft.nl

IPv6
-Header simplifications over IPv4:
	-Fixed format to all headers: no option element, no header length, but extension headers. Extension headers allow:
		-more officient forwarding
		-less stringent limits on length of options
		-greater flexibility for introducing new options in the future
	-No header checksum to diminish cost of processing -> other layers check and correct for errors
	-No hop-by-hop fragmentation fields, unit of transmission should also be the unit of control which can be determined by using the path MTU discovery function

-New features:
	-expanded addressing capability: increases IP address size from 32 bits to 128 bits
		-more levels of addressing hierarchy
		-much greater number of addressable nodes
		-simpler auto-configuration of addresses
	-scalability of multicast routing is improved by adding a scope field to multicast addresses
	-new type of address: anycast address
	-flow labeling capability: labeling of packets which require a special handling requested by user
	-ICMP streamlined
	-multicast IGMP added
	-DCHP improved with enhanced plug & play (server-less) autoconfiguration features
	-end-to-end IP layer authentication & encryption
	-elimination of triangle routing for mobile IP

-Header fields
	-version (4): Internet Protocol version number is 6
	-traffic class (8): Identification field between different classes or priorities of IPv6 packets
	-flow label (20): a source may label several flows or sequences of packets belonging to a certain QoS class
	-payload length (16): length in octets of IPv6 payload including any extension headers
	-next header (8): identifies type of header immediately following the IPv6 header
	-hop limit (8): decremented by 1 every forwarding node -> packet is discarded if hop limit is zero
	-source address (128): address of the originator of the packet
	-destination address (128): address of intended recipient of the packet

-IPv6 header twice as long as IPv4 header without options (40 vs 20 bytes)

-Extension headers [header code: header type: explanation]
	-0: hop-by-hop options: info about packet delivery examined by each router on the path
	-43: routing: similar to IPv4, source routing: explicit path
	-44: fragmentation: similar to IPv4 but only done by source nodes, id field 32-bits

	-51: authentication: specifies authentication rules (Psec(?))
	-52: encrypted payload: indicates that payload is encrypted
	-60: destination options: options that end-systems agree upon

IPv6 addressing
-three types: unicast, anycast and multicast (no broadcast): assigned to interfaces, not nodes
-Text representations:
	-x1:x2:x3:x4:x5:x6:x7:x8 where xk hexadecimal value of 16-bit
	-zero compressed notation: xk's which are zero are left out
	-mixed IPv4 and IPv6: x:x:x:x:x:x:d.d.d.d where d decimal value of 8-bit IPv4 address
	-multicast: FFuv:x2:x3:x4:x5:x6:x7:x8

Host-density ratio (HD supermooi jwz)
-efficiency of address allocation:

$$HD = \frac{\log(n)}{\log(N)} = \log_N(n)$$

with n number of allocated objects in addressing plan and N max. number of allocatable objects

- 0 <= HD <= 1
	-HD = 0.8 reasonable
	-HD = 0.85 painful
	-HD = 0.86 very painful
	-HD = 0.87 practical maximum

Internet Control Message Protocol (ICMP)
-encapsulated in IP packet with protocol field in IP header equal to 1
-two modes: query (echo, timestamp, address mask can be requested) and error (host unreachable, TTL zero, destination network unknown etc.)
- source also obtains the IP address of the error observing router: easy to identify the place and cause of the error
-basic protocol used in many other programs or protocols
-not to enhance reliability but to provide feedback about network problems

Ping
-echo request via IMCP to a host
- Ping -f -l <size in bytes> address
	- -f sets don't fragment flag
	- -l sets payload size

Traceroute
-makes use of ICMP error mode
-sends multiple UDP packets with increasing TTL
-returns path to the address

-Flaws:
	-IMCP messages may be discarded due to security reasons
	-UDP packets might traverse multiple paths (?)
	-Interface and no router addresses

%%%%%%%%%%%%%%%%%%%%%
% H5 TCP
%%%%%%%%%%%%%%%%%%%%%
Flow control: actions taken to control the flow of data to prevent loss, occupation of too many network resources
Congestion control: set of actions taken by the network to recover from overload situations where considerable losses have occurred
Protocol for flow control: TCP

Major protocols in TCP/IP stack on OSI layer 4:
-User Datagram Protocol (UDP)
    -offers unreliable connectionless packet service
    -complements IP at layer L3 with error checking and application distinction
    -checks the integrity of the entire UDP packet, header and data part
    -header consists of four 16-bit fields:
        -source port
        -destination port
        -UDP length
        -UDP Checksum
            -one's complement of the one's complement sum of a pseudoheader
            -pseudo header only constructed to compute the checksum, not transmitted
    -suited for querying and realtime applications

-Transmission Control Protocol (TCP)
    -provides reliable connection-oriented transport over unreliable IP service at L3
    -full duplex (concurrent transfer in both directions)
    -identifies a connection by a pair of endpoints
        -result: given TCP port can be shared by multiple connections on the same machine

Operation of TCP
-selective repeat ARQ mechanisms for error correction
-data stream viewed as a sequence of octets: octet is the basic unit
-header consists of:
    -source port (16)
    -destination port(16)
    -sequence number (32)
        -corresponds to the sequence number of the first octet in the sender's TCP payload
    -Acknowledgement number (32)
        -sequence number of the next in order octet that the sender expects
    -header length (?)
        -specifies the length of the TCP header in 32-bit words
    -flags (6) one bit to set each of the six flags (nee niet walibi):
        -URG: activates the urgent pointer field
        -ACK: activates the acknowledgment field
        -PSH: requests the receiving TCP side to pass the data to the application immediately
        -RST: requests to abort the connection due to abnormal operation
        -SYN: requests a connection
        -FIN: informs the receiver no more data will be sent
    -window size (16)
        -corresponds to the available buffer space at the receiver

```
    -TCP checksum (16)
        -end-to-end checksum on header and data
        -computed the same way as the UDP checksum by constructing a TCP
pseudoheader of 96 bits prefixed to the TCP header, zero padded to multiple of
32 bits
    -urgent pointer (?)
    -options (32)
        -provides additional functions, most important:
        -maximum segment size
        -window scale, allows use of a larger advertised window size
            -multiplier not larger than 2^14
        -timestamp

Connection setup and connection termination
-setup: three-way handshake
       1. sender requests destination to open connection
            -SYN bit is set by sender
            -initial sequence number n_seq = x
       2. receiver replies with ACK
            -sets acknowledgment number n_ACK = x + 1
            -first segment of receiver has n_seq = y
       3. source sets ACK flag and acknowledgment number n_sec = y + 1

-termination
    -abort via the RST flag
    -graceful close
        -sender sends FIN flag to receiver, receiver acknowledges by returning
n_ack
        - receiver --> sender still continues until receiver sends FIN

TCP is "self-clocking"
-amount of data in transit between sender and receiver: bandwidth * delay
    -referred as bandwidth-delay product of a connection
-when dealing with a link bottleneck, packets take more time to transmit

-self clocking: a receiver cannot generate acks faster than data packets
travel through the network
    -packets and acks clocking each other

-slow start algorithm
    -developed to start the self-clocking
    -imposes the congestion window W to the sender
        -amount of data allowed to send = min(advertised window, W)
    -congestion window doubles every RTT (round trip time)
        -increase continues until slow start threshold (sstresh) is reached

-congestion avoidance
    -phase with a linear increase of W by 1 every RTT
    -entered when crossing sstresh
    -if(W < sstresh)
        w++;
     else
        W += 1/W;
    -informing end-systems that congestion is occurring
    -a rule to adapt the sending rate on congestion notification
```

Retransmission
-TCP sender retransmits all unacknowledged segments after RTO is exceeded or consecutive duplicate acks have been received
-when packet loss is detected by RTO:
    -sstresh is set to half the current congestion window W
    -congestion window is set to W = 1
    -slow start is initiated

-fast retransmission
    -retransmitting lost packet after a number of duplicate acks without awaiting time-out

Fast recovery
1. at receipt of 3rd duplicate ack:
    -sstresh <- W/2
    -retransmit missing segment L
    -W <- sstresh + 3

2. at receipt of another duplicate ack:
    -W++
    -transmit new, not in transit segment

3. at receipt of ack L
    -W <- sstresh
       start congestion avoidance phase

-explanation blz 128-130

Throughput of TCP
-during a periodic cycle of D seconds or D/T = W/2 RTT's, TCP connection sends M packets:

    M = (D/T)*(W/2) + (D/T)*(W/4) = (3/8)*W^2

-transmission rate R or throughput:

    R = M/D) = (3W)/(4T) (packets/s) = sqrt(3/2)/(T*sqrt(p)) (inverse square root law)   p chance of packet loss

    -general version for sawtooth ranging from a*W to W

        R = sqrt((1+a)/2(1-a)) * 1/(T*sqrt(p))

        -a = 0.5 gives previous and most used equation

Additive Increase, Multiplicative Decrease (AIMD)
-sawtooth-like behaviour of the window size
-results in equal division of the available capacity aming identical competing TCP sources

%%%%%%%%%%%%%%%%%%%%%
% H6
%%%%%%%%%%%%%%%%%%%%%
Forwarding: action of placing an incoming IP packet to an outgong interface
-table look-up part
-packet transfer part

Network topologies
-graph G
	-contains V vertices
	-connected by a set of E edges
	-number of links L
	-number of nodes N
-full mesh: N(N-1)/2 links

-link weights
	-characterized by a vector w(i -> j)
	-assumption: symmetry in both directions w(i -> j) = w(j -> i)

-degree of a node in a graph (d_j)
	-number of neigbouring nodes, 0 <= d_j <= N-1
	-basic law: sum(d_j from j=1 to N) = 2L
	-power law: Pr[d_j=k] = c*k^a
		-a =~ -2.2, c normalization factor -> Pr = 1

-random graphs
	-G_p(N)
		-all graphs with N nodes in which the links are chosen independently with probability p
		-total number of links known on average as p*L_max
		-refinement: geometric G_{pij}(N), pij = exp(-a|r_i - r_j|)
	-G(N,L)
		-set of graphs with N nodes an L links
		-number of different network topologies: nCr(L_max,L)
	"L_max boven L"
			-distribute set of L ones in L_max possible places in upper triangle of matrix A
				-aij = 1; p = L/R_max
				-aij = o; q = 1-p

-connectivity
	-redundancy D: link-to-node ratio
	-edge connectivity l(G)
		-smallest number of edges (links whose removal disconnects the graph
	-vertex connectivity k(G)
		-smalles number of vertices (nodes) whose removal disconnects the graph
		-k(G) cannot exceed l(G)
	-delta(G) minimum degree of node

	-Menger's theoram: the maximum number of link(node)-disjoint paths between A and B is equal to the minimum number of links(nodes) separating A and B
		-best possible reliability acheved if k(G) = (2L)/N

shortest path routing
-path weight is equal to the sum of weight of containing links

-single parameter shortest path problem
	-subsections of a shortest path are also shortest paths

-relaxation
	RELAX(u,v,w,d,pi)
	1. if d[v] > d[u] + w(u -> v)
	2. then d[v] <- d[u] + w(u ->v)
	3. pi[v] <- u

	-found paths decrease in length until shortest path is found
	-analogy of a spring: it is stretched initially when a long path is found. As shorter paths are found, the spring 'relaxes'.

Dijkstra's algorithm
-assumes only positive weights
1. find next connected node with least weight
2. find connected nodes to that first connected node and determine the shortest path from starting point (relaxation)
3.on to the next node, repeat step 1 & 2 until shortest path is found

-complexity: O(N^2)

Bellman-Ford algorithm
-allows negative link weights
-less complex (computation wise) than Dijkstra's, used in the early Internet
1. give value to each node connected to starting/previous node
2. relaxation: if multiple paths are formed, determine which one is the shortest and assign that weight value to the reached node
3. repeat step 1 & 2 until end node is reached

Alternating path search
-alternate scanning from the source node and scanning from the destination node
	-results in higher efficiency
	-performance: 1/T*sum(n_alternating_i/n_dijkstra_i from i = 1 to T)

Kruskal's algorithm
1.connect al nodes with the lowest link weight
2. increase link weight with 1 and repeat step 1 until all nodes are connected
3. one path from begin to end has been formed, this is the shortest path

Prim's algorithm
-very similar to Dijkstra's
-updating with relaxation process does not maintain smallest value so far but the smallest link weight so far that connects the node in question to the root tree

%%%%%%%%%%%%%%%%%%%%%
% H7
%%%%%%%%%%%%%%%%%%%%%

Flooding
-advantages:
	-simple and robust
	-use of all paths assures shortest time

-disadvantage: overhead, many duplications

-simple flooding: each incoming packet is sent out on every outgoing link or interface except for the interface it entered
-ways to stop flooding:
	-wait until TTL is zero
	-compare sequence numbers and discard older ones
-selective flooding: only via a minimum spanning tree, topology update information is distributed to all nodes

-multi-hop wireless networks/ad hoc networks:
	-routing based on flooding proposed
	-application of QoS routing
		-each intermediate router modifies the QoS fields and appends his IP address to the path list.
			-updating consists for min-max metrics and additive metrics

Topology changes
-changes that occur infrequently: joins/leaves of nodes (1)
-changes that rapidly change in time: metrics coupled to state of resources (2)
-mobile network: variations in location of the nodes

Classification of routing protocols
-static (non-adaptive) vs dynamic (adaptive) routing
	-static: routing tables manually constructed
		-paths are pre-computed
		-useful in small networks with slow groth and slow changes
	-dynamic: routing tables are created by automated construction and updating
		-enables large networks to cope with rapid topology changes
		-enables to respond more quickly to failures
		-automation eliminates human errors

-centralized vs distributed routing
	-centralized: paths in the network are computed in one location
		-computed on route server
		-suffers from single point of failure
	-distributed: computations executed in a set of separate nodes, no single authority
		-if one node fails, routing is still in operation

-source routing
	-source computes entire path
	-signaling required (CO) or storage of complete path in every packet
	-loop free
	-general multiple parameter routing
	-QoS routing (PNNI)

-hop-by-hop routing
	-each node computes (sub)path from itself to destination

	-routing consistency requires that all nodes use same routing algorithm
	-flexible, robust, connectionless
	-only single parameter routing
	-best effort (OSPF, RIP)
	-used in the Internet

-Pre-computation vs on-demand routing
	-pre-computation
		-single parameter:
			-slow topology updating rate
			-forwarding table contains only destinations: O(N)
			-computation of total table: O(N^2)
		-multiple parameter:
			-table contains
				-all source-destination pairs: O(N^2)
				-all combinations of QoS measures q
			-fast topology updating rate due to resource coupling
	-topology changes of (2): only on-demand routing feasible

two levels of routing:
-Intra-domain routing
-Inter-domain routing

Intra-domain routing
-protocol families in the Internet:
	-distance vector protocols (RIP, Bellman-Ford)
		-flood list of distances to neighbours
		-maintain list of shortest distances
		-protocol itself constructs forwarding table
		-simple but vulnerable
	-link state protocols (OSPF, Dijkstra's)
		-flood topology information
		-nodes maintain entire map of network
		-local routing algorithm computes forwarding table
		-more robust but more complex
		-first developed for ARPANET to overcome the problems with distance vector protocols
		-if all nodes have the same map and algorithm, loops cannot occur
		-OSPF developed by IETF for Internet

-Routing Information Protocol (RIP)		~rest in pieces
	-first dynamic and distributed routing algorithm used in the early Internet
	-cold start: fully distributed routing
		-all routers power up simultaneously
	-artifacts: bouncing effect and count to infinity
	-uses asynchronous version of the Bellman-Ford algorithm

	-bouncing effect
		-A connected to B, B connnected to C (example)
		-connection B to C breaks

		-node A sends before B floods its distances, B to C link falsely updated to flooded value of A instead of INF
		-repeats until A to C reaches 10

	-count to infinity
		-when a pair of nodes gets disconnected because two or more links break, they keep upping their distance to the disconnected root tree until infinity
		-links have to be repaired to stop this

	-heuristic remedies for looping
		-split horizon: if node A is routing packets to X through B, it makes no sense for A to announce to B that X is only a short distance from A
		-triggered updates: flood information as soon as a change has occurred
		-hold-down: after the metric for a route entry changes, the router accepts no updates for the route until the hold-down timer expires

	-RIP summary:
		-simple and adequate for small, reliable (link failures are rare) networks
		-inadequate for large and complex networks:
			-slow convergence
			-network is left too long in transient state
			-loops Mai occur
			-leading to temporary congestion

-Open shortest path first (OSPF)
	-replacement of RIP to alleviate artifacts present
	-main purpose to quickly update the routing tables after a topology change in a consistent way
	-invokes Dijkstra's shortest path algorithm
	-all nodes maintain a map of the network
	-Quickly update after any topology change
		-purpose: synchronized copy of the link state in all nodes of network
	-properties:
		-secured flooding and fast, loopless convergence of link state
		-precise (multiple) metrics
		-multiple paths (any routing problem can be computed)
		-can take the QoS/ToS (welke is niet duidelijk??) into account

Inter-domain routing
-Autonomous System (AS): topology subnet under single administrative control
-Border Cateway Protocol (BGP)
	-fundamental Internet routing protocol
		-'glue' between ASs
		-complex protocol
		-only unicast
	-distance vector protocol enhanced with path vectors
		-path vector contains entire path
	-shortest path: based on policies
	-exchange of routing table via TCP
	-current scaling: problematic
		-RIB grows large

-GIDR looses initial efficiency (aggregation)
-solves stable path problem