Počítačová sieť

- skupina počítačov a zariadení, vzájomne prepojených komunikačnými kanálmi, ktoré uľahčujú komunikáciu medzi užívateľmi a umožňujú zdieľanie zdrojov(HW, SW, dáta ...)

Connection-oriented networks (circuit switching networks)

- medzi dvoma zariadeniami je vytvorené komunikačné spojenie, ktoré je udržiavané počas celej doby komunikácie
- eviduje sa stav spojenia
- jednoduchá implementácia **QoS**(Quality of Service)
- napr: telefónny systém pracuje na takýchto spojeniach

Connection-less networks (packet switching networks)

- pre komunikáciu medzi dvoma zariadeniami nieje určena žiadna špecifická cesta, dáta sú rozdelené do packetov a cez sieť môžu putovať do cieľa rôznymi cestami
- neeviduje sa žiadny stav spojenia
- u príjemcu sa packety skladajú späť do pôvodných dát
- zložitá implementácia QoS
- napr: Internet je tiež connection-less sieť

Network protocols

- forma, v akej bude prebiehať komunikácia medzi zariadeniami, musí byť známa všetkým účastníkom komunikácie
- protokol je množina pravidiel, ktorá definuje formát komunikácie medzi komunikujúcimi zariadeniami a tiež definuje operácie, ktoré sa majú vykonávať počas posielania dát alebo po ich prijatí
- protokol definuje **Čo** je predmetom komunikácie, **syntax** = štruktúra/formát posielaných dát
- protokol definuje **Ako** bude komunikácia prebiehať, **sémantika** = špecifikuje význam každej sekcie bitov
- protokol definuje **Kedy** bude komunikácia prebiehať, **timing** = kedy a ako rýchlo môžu byť dáta poslané

Standardization

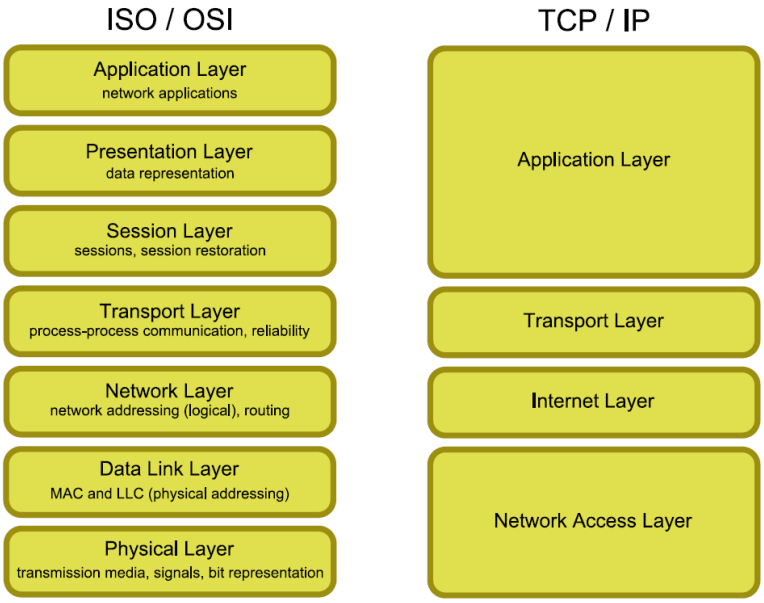
- **de facto** – štandardy, ktoré neboli schválené oficiálnou štandardizačnou organizáciou, ale sú rozšírené a vo veľkom sa používajú, často štandardy výrobcov
- **de jure** – štandardy schválené oficiálnou štandardizačnou organizáciou
- najznámejšie oficiálne IT štandardizačné organizácie: ISO, ANSI, IEEE, IETF (RFCs), IEC, ITU-T ...

ISO/OSI Model

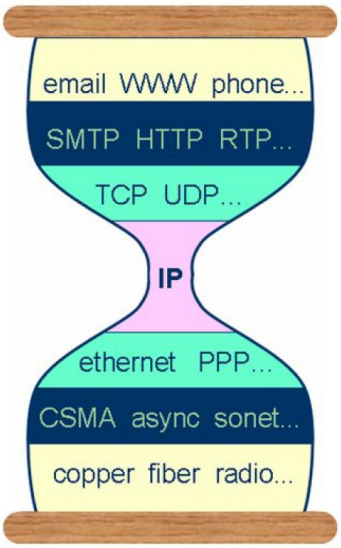
- 7 vrstvový sieťový model navrhnutý organizáciou OSI, popisujúci fungovanie komunikačných systémov
- každá vrstva je zodpovedná za určitú funkcionalitu a k prenášaným dátam pridáva riadiace dáta vrstvy
- každá vrstva komunikuje len so susednými vrstvami, každá vrstva využíva služby vrstvy nižšej a poskytuje svoje služby vrstve vyššej
- vrstvy tohto modelu sú len abstrakciou

TCP/IP Internet Protocol Suite

- protokolová sada, ktorú používa aj Internet, vychádza z ISO/OSI sieťového modelu ale TCP/IP niektoré abstraktné vrstvy ISO/OSI modelu spája do jednej a jeho model má preto len 4 vrstvy



ISO/OSI a TCP/IP sieťové modely



TCP/IP model presýpacích hodín

L1 – Physical layer

- **PDU: bit**
- spravuje prenosové médium a poskytuje svoje služby Data Linkovej vrstve
- stará sa o samotný prenos dát od odosielateľa k príjemcovi cez prenosové médium
- na fyzickej vrstve neexistuje žiadny proces adresovania/smerovania, má len point to-point spojenie
- služby fyzickej vrstvy:
    - **Bit-to-Signal Transformation**: transformácia binárnych dát do elektrického signálu
    - **Bit-Rate Control**: kontrola posielana max. množstva dát za sekundu
    - **Bit Synchronization**: synchronizácia odosielateľa a prijímateľa
    - **Multiplexing**: rozdelenie fyzického média do viacerých logických kanálov
    - **Circuit Switching**: väčšinou je to funkciou fyzickej vrstvy

L2 – Data Link layer

- PDU: frame
- poskytuje svoje služby vrstve Sieťovej(L3) a využíva služby Fyzickej vrstvy (L1)
- zo Sieťovej vrstvy prijíma pakety a tie transformuje do rámcov (frames)
- riadi prístup k prenosovému médiu (Media Access Control)
- zabezpečuje kontrolu toku (Flow Control), ktorým sa predchádza zahlteniu príjemcu
- za pomoci fyzickej vrstvy zaručuje prenos rámcov medzi zariadeniami v rámci jednej LAN
- služby data linkovej vrstvy:
    - **Framing**: enkapsulácia paketov do rámcov
    - **Addressing**: poskytuje fyzické MAC adresy zariadení , rámce obsahujú zdrojovú a cieľovú MAC
    - **Error Control**: kontrola a oprava vzniknutých chýb na fyzickej vrstve
    - **Flow Control**: zabraňuje zahlteniu príjemcu, stop-and-wait, sliding-window mechanizmy...
    - **Medium Access Control (MAC)**: riadi prístup zariadení k zdielanému prenosovému médiu

Error Control:

- odosielateľ pridáva k posielaným dátam kontrolné bity, ktoré sú výsledkom funkcie na samotných prenášaných dátach, príjemca na dátach vyráta tú istú funkciu, ak sa kontrolné bity nezhodujú, znamená to chybu prenosu, príjemca sa môže pokúsiť dáta opraviť sám(Forward Error Correction (FEC)) alebo požiadať o znovuposlanie (Automatic Request for Retransmission(ARQ))

Media Access Control(MAC):

- zabraňuje kolíziám vznikajúcim pri súčasnom prenose dát viacerými zariadeniami po zdieľanom prenosovom médiu
    - **Random-Access protokoly**: Aloha, CSMA/CD, CSMA/CA
    - **Controlled-Access protokoly**: rezervácie, polling, tokens ...
    - **Channelization protokoly**: založené na multiplex technológii, napr: FDMA, TDMA

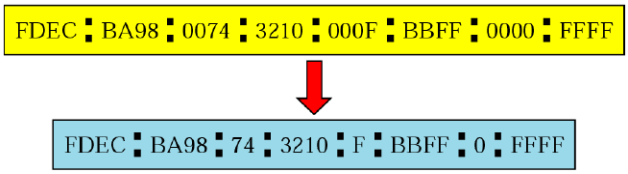
L3 – Network layer

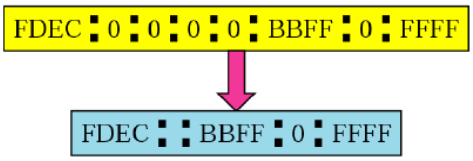
- PDU: packet
- poskytuje svoje služby vrstve Transportnej(L4) a využíva služby vrstvy Data Linkovej(L2)
- za pomoci data linkovej vrstvy zaručuje prenos paketov medzi zariadeniami aj v rôznych LAN
- logicky prepája nezávislé LAN siete, vyššie vrstvy pracujú s ilúziou jednej WAN siete
- umožňuje jednoznačne identifikovať každé zariadenie v sieti
- zabezpečuje smerovanie paketov
- v spolupráci s data linkovou vrstvou vie preložiť L3 adresu na L2 adresu a opačne
- služby sieťovej vrstvy:

- **Internetworking**: vytvorenie ilúzie jednej WAN siete poprepájaním jednotlivých fyzicky vzdialených sietí.
- **Packetizing**: segmenty/datagramy z transportnej vrstvy sú zabalené do paketov
- **Fragmenting**: rozdelenie datagramu na menšie fragmenty, ktoré sú poslané nezávisle
- **Addressing**: IP adresy unikátne v celej sieti, paket obsahuje zdrojovú a cieľovú
- **Address Resolution**: ARP, RARP protokoly
- **Routing**: proces smerovania, výberu cesty ktorou poslať paket k príjemcovi
- **Control Messaging**: poskytuje informácie o nedostupnosti siete/príjemcu ICMP protokol

Addressing:

- IPv4 adresy (32 bits) vs. IPv6 adresy (128 bits)
- IPv4 addresses:
  - **Unicast Address**: identifikácia jedného sieťového interfaceu, jedného zariadenia
  - **Broadcast Address**: všetkých zariadení v danej LAN
  - **Multicast Address**: identifikácia skupiny zariadení
- IPv6 addresses:
  - riešením vyčerpania adresného priestoru IPv4
  - IPv6 adresa má 128 bitov => 2^{128} možných IP adries, používa sa hexadecimálna notácia
  - nuly vyskytujúce sa na začiatku čísla môžu byť vynechané v každej skupine
  - po sebe idúce nulové skupiny čísel môže byť nahradené symbolom "::" vždy ale len jedna sekvencia takýchto skupín v IP adrese





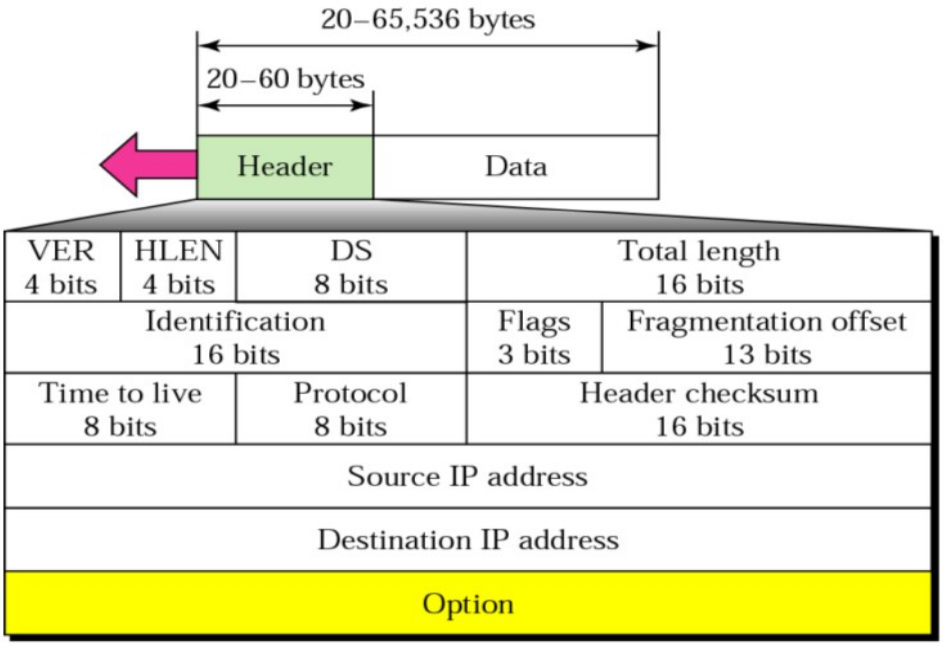
  - unicast a multicast adresy fungujú rovnako ako v IPv4
  - broadcast adresa sa v IPv6 nepoužíva, boli na to vyhradené multicast adresy
  - **Anycast Address**: nový typ adresy oproti IPv4, rovnako ako multicast identifikuje skupinu príjemcov dát, ale samotné dáta pošle len jednému členovy tejto skupiny, najbližšiemu

Internet Protocol (IP):

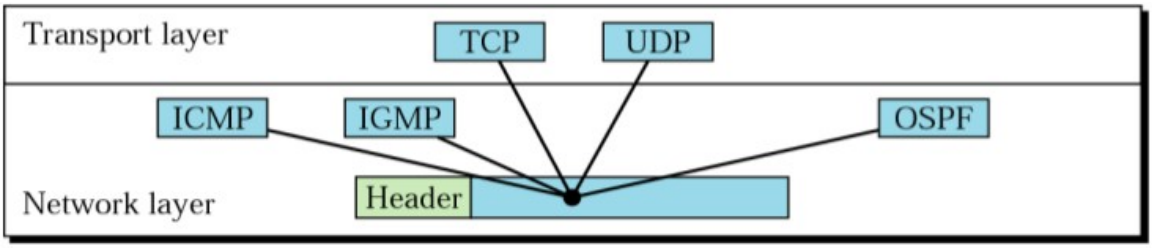
- najrozšírenejší protokol sieťovej vrstvy
- stará sa o doručovanie dát vo forme datagramov
- poskytuje unreliable(nespolahlivú) službu
- využíva doplnkové protokoly ako ICMP, ARP, RARP, IGMP pre neštandardné situácie

a pre L2 identifikáciu sieťových rozhraní
- identifikácia zariadení prebieha pomocou IP adries

IPv4 Datagram:



- **Version(VER):** verzia IP protokolu
- **Header length(HLEN):** veľkosť hlavičky datagramu, vďaka polu Option sa mení
- **Differentiated service(DS) alebo Type of Service(TOS):** určuje dôležitosť(prioritu) datagramu pre potreby zaručenia QoS
- **Total length**: veľkosť celého IP datagramu(max. $2^{16} - 1 = 65535$ bytov)
- **Identification, Flags, Fragmentation offset:** polia používané pri fragmentácii
- **Time to Live(TTL)**: číslo udávajúce maximálny počet hopov datagramu, každý router, ktorý datagram navštívi, zníži toto číslo o 1, po dosiahnutí TTL = 0 sa datagram zahodí, je to prevencia pred nekonečným potuvaním datagramu
- **Protocol:** protokol vyššej vrstvy, udáva ktorému protokolu vyššej vrstvy má byť datagram odovzdaný



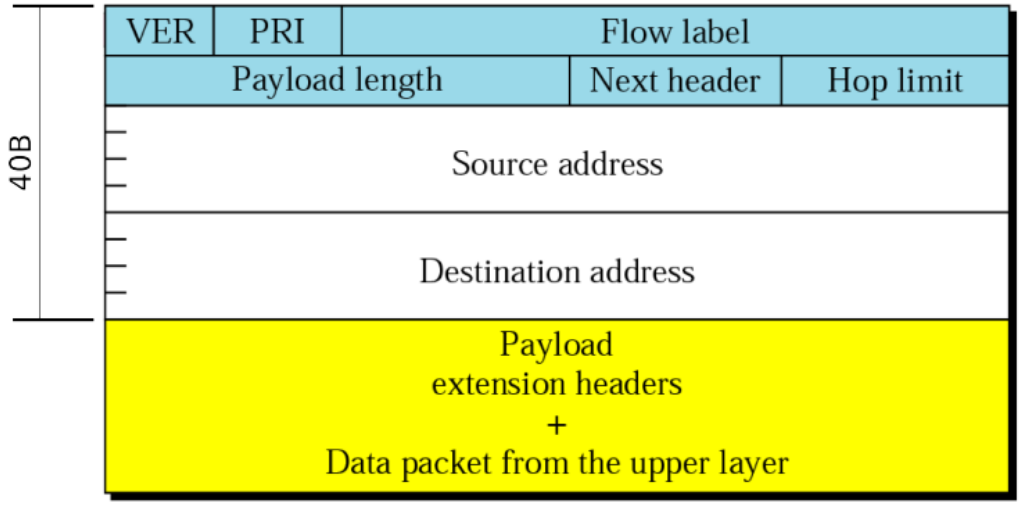
- **Header checksum**: checksum hlavičky IP datagramu, checksum musí byť vypočítaná na každom smerovači, ktorý datagram navštívi, kvôli meniacim sa

poliam hlavičky ako napr. TTL pole
- **Source/Destination IP address:** 32-bitová IPv4 adresa ako identifikátor uzlov
- **Options**: voliteľná časť IP datagramov, vhodné pre testovanie a debugging
- **Data**: samotné prenášané dáta

Internet Control Message Protocol (ICMP):

- dodatočný protokol pre IP protokol
- poskytuje informácie o vzniknutých chybách počas dátového prenosu
- poskytuje informácie o stave siete
- napr:
    - Destination Unreachable: "Destination" môže byť: protokol, port, host, sieť
    - Time exceeded: vypršanie TTL, alebo všetky fragmenty pôvodnej správy nedorazili do cieľa v stanovenom časovom limite
    - Echo request/reply: požiadavka o zopakovanie zaslania

IPv6 Datagram:



- fixovaná veľkosť hlavičky = 40 bytov
- pole Options a polia Fragmentácie sú dostupné cez extension headers
- pole checksum už nieje pri IPv6 podporované vôbec

- **Version(VER):** verzia IP protokolu, v tomto prípade 6
- **Priority(PRI)**: priorita datagramu vzhľadom k zahlteniu
- **Flow label**: pole určujúce špeciálne zaobchádzanie s tokom dát
- **Payload length**: veľkosť datagramu bez základnej hlavičky
- **Next header**: určuje hlavičku, ktorá nasleduje za touto základnou (extension header alebo transport header)
- **Hop limit:** pole totožné s polom TTL v IPv4
- **Source/Destination address**: IPv6 adresa ako identifikátor uzla

Internet Control Message Protocol v6(ICMPv6):

- narozdiel od ICMPv4 zahŕňa aj funkcionalitu protokolu ARP a IGMP

Routing(smerovanie):

- proces smerovania, nájdenia cesty zo zdroja do cieľa

Routing table(smerovacia tabuľka):

- reprezentuje lokálny pohľad na sieťovú topológiu
- používa sa pre hop-by-hop smerovanie, vtedy sa router nerozhoduje o celej ceste packetu do cieľa, ale len najbližším hopom, tj. kam má poslať packet on, niekomu bližšie k cieľu
- vytváranie/udržovanie routovacích tabuliek:
    - **static:** manuálne upravované záznamy v smerovacej tabuľke
    - **dynamic:** reagujú na zmeny v sieti
        - centralized: nejaký centrálny uzol kontroluje celé smerovanie
        - isolated: každý uzol sám za seba
        - distributed: uzly spolupracujú a vymieňajú si navzájom informácie

Distributed routing:

Distance Vector(DV) – Bellman Ford algorithm:
- susediace routre si periodicky alebo pri zmene topológie vymieňajú svoje celé smerovacie tabuľky
- na základe prijatých tabuliek od susedov si updatuje údaje vo svojej tabuľke
- update o zmene v sieti zasielaný len priamym susedom, propagovaný ďalej je len vtedy, keď zmena vedie k zmene najkratšej cesty
- problémy so smerovacími slučkami
- falošné informácie sa môžu rozšíriť do celej siete
- **vhodné pre menšie siete**

platí pravidlo: VŠETKY INFORMÁCIE O SIETI LEN SVOJIM SUSEDOM

Link State(LS) – Djikstra algorithm:
- smerovače si periodicky vymieňajú informácie o stavoch liniek, ku ktorým sú priamo pripojené
- každý router si udržuje tabuľku kompletných informácií o celej sieti, každý router vie o každom routri v celej sieti
- O(nE) správ pri zasielaní updatov
- router môže šíriť falošné informácie len o svojich priamych susedoch
- na smerovanie sa používa Djikstrov algoritmus pre nájdenie najkratšej cesty
- **vhodné pre veľké siete**

platí pravidlo: INFORMÁCIE LEN O SVOJICH SUSEDOCH VŠETKÝM

Distance Vector protocol – RIP

- RIPv1, RIPv2(pridáva viacero funkcionalít k pôvodným funkciám RIPv1), RIPng(pridanie podpory IPv6 adries)
- siete sú identifikované pomocou CIDR mechanizmu
- ako metrika je použitý počet hopov, nekonečno = 16 hopov => tieto protokoly nemôžu byť použité v sieťach s najkratšou cestou dlhšou ako 15 hopov medzi ľubovolnými 2 routrami
- routre si vymieňajú tabuľky každých 30 sekúnd, alebo pri zmene v sieti pomocou UDP transportného protokolu
- timeout = 180s, po uplynutí sa linka považuje za mŕtvu
- protokoly použiteľné pre menšie siete
- RIPv1 updaty sú posielané na broadcast adresu, RIPv2 na multicast: 224.0.0.9

Link State protocol – OSPF

- najpoužívanejší Link State protokol
- ako metriku používa cenu(cost), ktorá je daná defaultne:
    - cost = 100000000/bandwidth, môže byť ale manuálne editovaná
- poskytuje autentizáciu správ, ale iba do verzie OSPFv2, od OSPFv3 sa spolieha na IPv6 IPSec
- umožňuje rozdelenie autonómnych systémov do podsystémov
- umožňuje použiť viacero východzích liniek routra s rovnakou najnižšou cenou pre prenos dát, tzv Equal-Cost MultiPath(ECMP)
- podporuje VLSM
- protokolové číslo = 89
- pre protokolové updaty používa multicast adresy 224.0.0.5 a 224.0.0.6 pre IPv4 FF02::5 a FF02::6 pre IPv6
- OPSF správy:
    - Hello Packet
    - Database Description Packet
    - Link State Request Packet
    - Link State Update Packet
    - Link State Acknowledgement Packet
- pre nájdenie najkratších ciest sa používa Djikstrov algoritmus

Autonómny systém(AS):

- skupina zariadení / časť siete, pod rovnakou správou nejakej organizácie
- zjednodušuje správu siete, redukuje počet vymieňaných informácií
- 16 bitový identifikátor je priradený každému AS
- rozlišujeme spôsoby pripojenia AS do siete Internet:
  - Stub AS
  - Multihomed AS
  - Transit AS

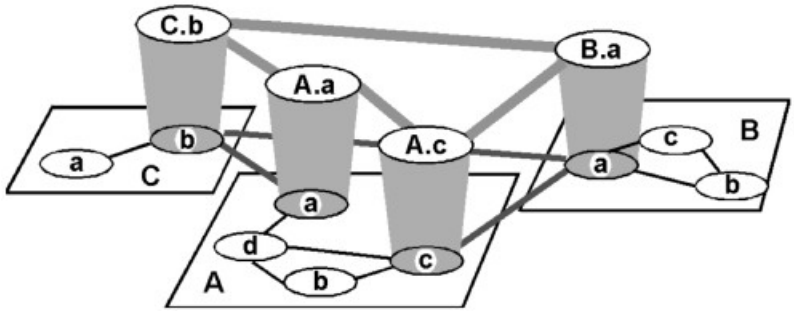
Použitím AS rozlišujeme smerovanie na:

interior routing:

- smerovanie vo vnútri AS
- pod správou admina AS
- primárnym cieľom je výkon
- používajú sa Interior Gateway Protocols(IGP): napr: RIP, OPSF

exterior routing:

- smerovanie medzi AS
- primárnym cieľom je podpora definovanej politiky a škálovateľnosť
- používajú sa Exterior Gateway Protocols(EGP): napr: EGP, BGP-4



L4 – Transport layer

- PDU: segment/datagram
- poskytuje svoje služby vrstve aplikačnej a využíva služby sieťovej vrstvy
- získané dáta z aplikačnej vrstvy zabalí do segmentov, tie doručí cieľovej aplikácii
- process – to – process delivery
- routre nemajú prístup k dátam transportnej vrstvy obsiahnutým v packete
- služby transportnej vrstvy:
    - **Packetizing**: transformácia dát do packetov
    - **Connection Control**: connection-oriented a connectionless služby
    - **Addressing**: adresy transportnej vrstvy sú porty, packety obsahujú zdrojový a cieľový port, aplikácia je jednoznačne identifikovateľná v celej sieti pomocou IP adresa:port
    - **Connection Reliability**: Flow Control a Error Control, pri nižších vrstvách to je poskytované na node-to-node princípe, na transportnej vrstve na end-to-end princípe
    - **Congestion Control a Quality of Service**

Addressing – ports:

- adresy transportnej vrstvy sú porty
- sú to 16 bitové identifikátory (0 – 65535)

L4 – Connection-oriented:

- spojenie je ustanovené a udržiavané počas celej doby komunikácie
- pakety sú číslované
- o doručení, resp. nedoručení paketov sa presne vie, tj používa spätnú väzbu

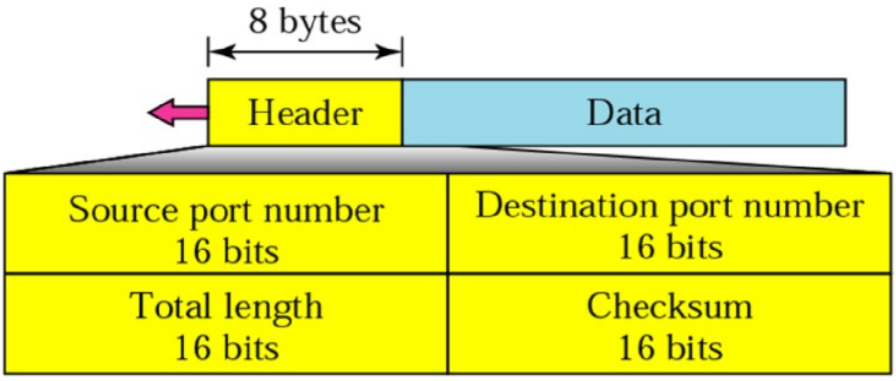
L4 – Connection-less:

- packety sú do cieľa poslané bez ustanovenia a udržiavania špecifického spojenia
- packety niesú číslované
- o doručení, resp. nedoručení packetov sa nevie, tj nepoužíva spätnú väzbu

User Datagram Protocol(UDP):

- protokol transportnej vrstvy, poskytuje connection-less služby
- služby IP vrstvy obohacuje len o process-to-process komunikáciu a jednoduchú error control
- používa malú hlavičku(header)
- používaný najmä pre komunikácie, kde sú povolené isté straty a ide najmä o rýchlosť komunikácie, ako real-time prenosy a multicast prenosy
- napr: DNS server

UDP header:

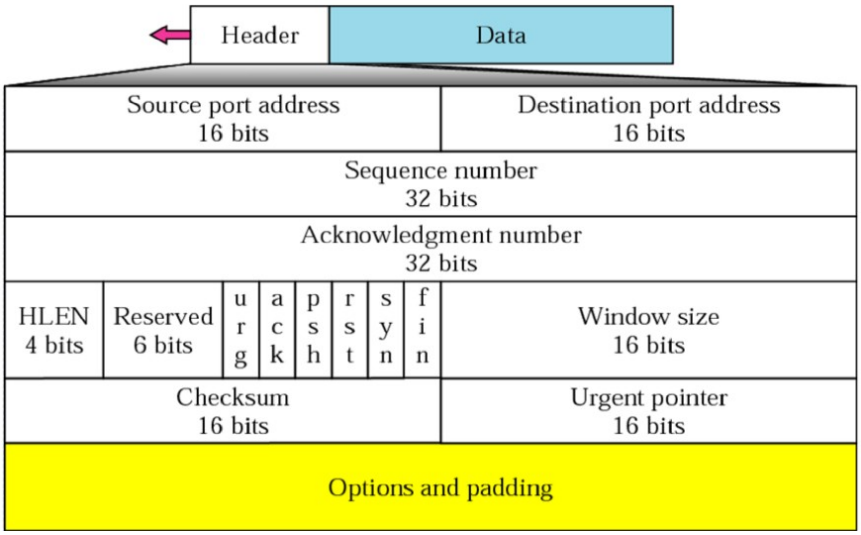


- **Source port/Destination port**: identifikácia zdrojovej/cieľovej aplikácie
- **Total length:** veľkosť UDP packetu (hlavička + dáta)
- **Checksum**: kontrolná suma UDP packetu (hlavička + dáta)

Transmission Control Protocol(TCP):

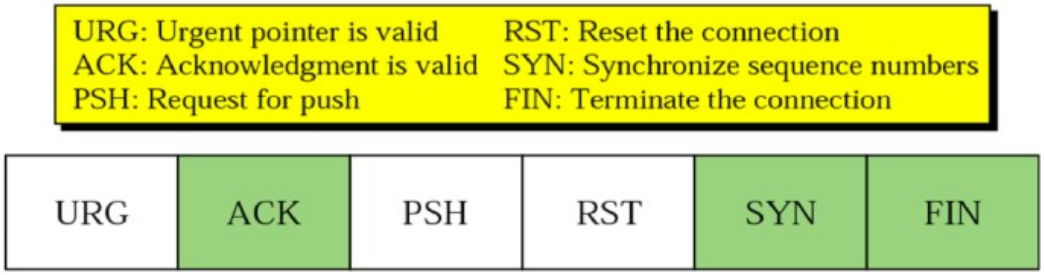
- protokol transportnej vrstvy, poskytuje connection-oriented služby
- packety sú v cieli prijaté všetky a v správnom poradí
- pred samotným prenosom je ustanovené spojenie medzi zdrojovým a cieľovým uzlom (three-way handshake)
- spojenie je rozpoznateľné len na koncových uzloch, routre ho nedetekujú
- podporuje len point-to-point spojenie, multicast nieje možný
- error control a multiplexing/demultiplexing je rovnaký ako v prípade UDP protokolu

TCP header:



- **Source/Destination port:** identifikácia zdrojovej/cieľovej aplikácie
- **Sequence number:** číslo priradené prvému bytu dát v segmente

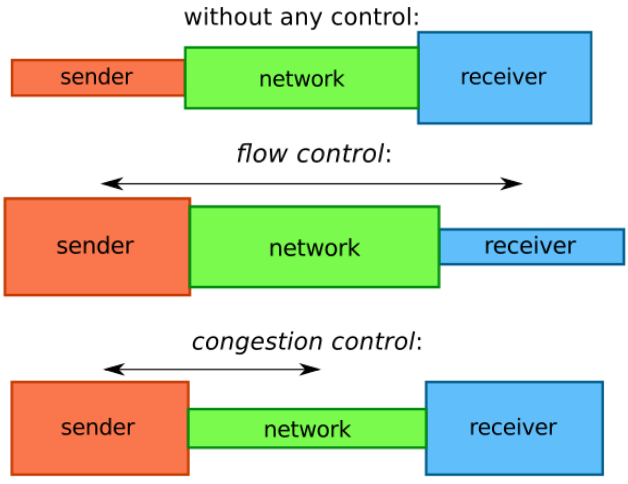
- **Acknowledgement number:** ak je nastavený kontrolný bit ack, potom toto číslo udáva sequence number segmentu, ktorý očakáva najbližšie
- **Header length(HLEN)**: veľkosť TCP hlavičky
- **Reserved:** vyhradené pre budúce účely
- **Kontrolné bity(urg, ack, psh, rst, syn, fin):** udávajú validitu daných polí



- **Window size**: veľkosť jednotky, ktorú je odosielateľ daného segmentu schopný prijať
- **Checksum**: kontrolná suma TCP segmentu (header + data)
- **Urgent pointer**: použitý, keď segment obsahuje urgentné dáta, 16 bitové číslo, ktoré udáva offset od sequence number, udávajúce posledný urgentný byte dát
- **Options:**

Flow Control: chráni príjemcu pred zahltením pomocou feedbacku od príjemcu s používaním rwnd(receiver window)
Congestion Control: chráni sieť pred zahltením pomocou odhadu dostupnej priepustnosti siete vypočítaného odosielateľom

Množstvo dát, ktoré je možné naraz poslať do siete je dané:
1. veľkosťou okna príjemcu (receiver's window size) = flow control
2. veľkosťou okna zahltenia (congestion window) = congestion control
Povolené množstvo je dané **nižšou** hodnotou z týchto dvoch:
- ownd = min{rwnd, cwnd}



L7 – Application layer

- poskytuje svoje služby užívateľom, využíva služby transportnej vrstvy
- slúži ako interface medzi užívateľom a sieťou
- aplikácie sú hlavným dôvodom existencie počítačových sietí
- obsahuje sieťové aplikácie a aplikačné protokoly
    - protokoly sú súčasťou sieťových aplikácií (web, email)
    - protokoly definujú formu komunikácie jednotlivých aplikácií

Komunikačné modely:
- Client-Server model
- Peer-to-peer model

Prístup k informáciám:
- pull model: prenos dát začína klient
- push model: prenos dát začína server

IPv6 adresy:

16 bits	16 bits	16 bits	16 bits	64 bits
2001	assigned by RIR	assigned by LIR	subnet address	interface address

- používa sa CIDR notácia, adresy sú classless
- Unicast adresa: adresa jedného sieťového interfaceu
- Multicast adresa: adresa skupiny sieťových interfaceov, prefix **ff00::/8**
- Anycast adresa: adresa skupiny sieťových interfaceov, dáta sú odoslané len najbližšiemu z tejto skupiny
- Broadcast adresa: v IPv6 už neexistuje, nahradená špeciálnymi multicast adresami

Path MTU Discovery:

- špeciálna technika rozhodovania o veľkosti fragmentov
- používa feedback(spätnú väzbu) vo forme ICMPv6 správy Packet Too Big
- zdrojový uzol pošle datagram do cieľového uzla, ak je datagram menší alebo rovný MTU linky cieľového uzlu => prejde a veľkosť datagramu sa použije aj pre budúce prenosy dát do daného cieľového uzlu, ak zdrojový uzol obdrží správu Packet Too Big, poslanie dát opakuje s menšou veľkosťou datagramu, ktorá je špecifikovaná v správe Packet Too Big

Neighbor Discovery Protocol:

- náhrada za IPv4 ARP protokol
- je súčasťou ICMPv6
- bolo mu pridaných viac funkcionalít:
 - autokonfigurácia IPv6 adresy
 - zisťovanie sieťových prefixov a iných konfiguračných informácií
 - detekcia duplikácie IP adresy (Duplicate Address Detection = DAD)

    - zistenie fyzickej MAC (L2) adresy zariadenia na základe IP adresy
    - nájdenie susedných routrov
    - zisťovanie dosiahnuteľných a nedosiahnuteľných susedov (NUD)
    - detekcia zmien fyzických adries
- pozostáva z piatich ICMP správ:
    - Router Solicitation(RS)
    - Router Advertisement(RA)
    - Neighbor Solicitation(NS)
    - Neighbor Advertisement(NA)
    - ICMP Redirect

Neigbor Discovery – L2 address resolution:

- proces podobný protokolu ARP
- využíva NS a NA správy
- bežný multicast prefix je daný: **FF02:0:0:0:0:1:FF00::/104**
- uzol, ktorý zisťuje L2 adresu zariadenia s danou IP adresou, vezme posledných 24 bitov IP adresy zisťovaného zariadenia a pridá ich k multicast prefixu:
- napr: hľadá L2 adresu zariadenia s IP adresou:
2AC0:56:A319:15:022A:FFF:FE32:5ED1 => vezme posl. 24 bitov = 32:5ED1
priradí k multicast prefixu => **FF02:0:0:0:0:1:FF32:5ED1**
- na túto multicast adresu zašle Neighbor Solicitation správu, ktorá obsahuje:
    - IPv6 adresu uzlu(target adresa), ktorého L2 adresu chce získať
    - L2 adresu zdrojového uzlu (ten čo zisťuje)
- následne uzol s danou IPv6 adresou odpovedá pomocou Neighbor Advertisement správy, ktorá obsahuje:
    - všetky IPv6 a L2 adresy, ktoré uzol vlastní
    - atribúty: R(Router): zariadenie je smerovač
      S(Solicited): udáva či NA bola vyžiadaná/nevyžiadaná(unsolicited)
      O(Override): udáva, či nová informácia by mala prepísať starú uloženú informáciu

Unsolicited Neighbor Advertisement:

- nevyžiadaná NA správa
- uzol odošle na multicast adresu všetkých uzlov(FF02:1), pri zmene jeho L2 adresy

Neighbor Discovery - Duplicate Address Detection(DAD):

- proces odhalenia duplikácie IPv6 adresy
- používa sa počas procesu autokonfigurácie
- podobne ako pri L2 adress resolution, uzol pošle NS správu na multicast adresu ale ako target adresu zvolí svoju vlastnú IPv6 adresu a zdrojovú adresu nechá nešpecifikovanú, ak zariadenie s rovnakou IPv6 adresou už existuje, pošle NA odpoveď na multicast adresu všetkých uzlov = FF02::1

Neighbor Discovery - Neigbor Unreachability Discovery(NUD):

- uzol periodicky kontroluje dostupnosť svojich susedov s ktorými komunikuje
- detekcia môže byť poskytnutá z vyššej vrstvy, napr. Protokolom TCP, alebo to musí IPv6 detekovať sám
- sused s IP adresou môže byť v jednom z nasledujúcich stavov:
    - Incomplete: bol odoslaný NS ale NA ešte nebolo prijaté
    - Reachable: pozitívna NA správa bola prijatá v rámci ReachableTime
    - Stale: viac ako ReachableTime času ubehlo od posledného pozitívneho NA
    - Delay: vypršal reachable time, protokol vyššej vrstvy môže potvrdiť dostupnosť
    - Probe: potvrdenie dostupnosti sa vykonáva

Neighbor Discovery – Autoconfiguration:

- možnosť autokonfigurácie zariadenia, IPv6 podporuje 2 typy autokonfigurácie:
1. Stateful autoconfiguration: pomocou DHCP ako v IPv4, používa ale DHCPv6
2. Stateless autoconfiguration: nový spôsob autokonfigurácie

Stateless autoconfiguration:

1. Link-Local Address Generation: samotné zariadenie vygeneruje tzv. link-local(tentative) adresu, ktorá má 10 bitový prefix = FE80, za ktorým nasleduje 54 núl a za tým nasleduje 64 bitový identifikátor interfaceu, ktorý predstavuje buď MAC adresu interfaceu, alebo náhodne vygenerované ID

2. Link-Local Address Uniqueness Test: použije Duplicate Address Detection(DAD) proces pre zistenie unikátnosti vygenerovanej IP adresy v lokálnej sieti

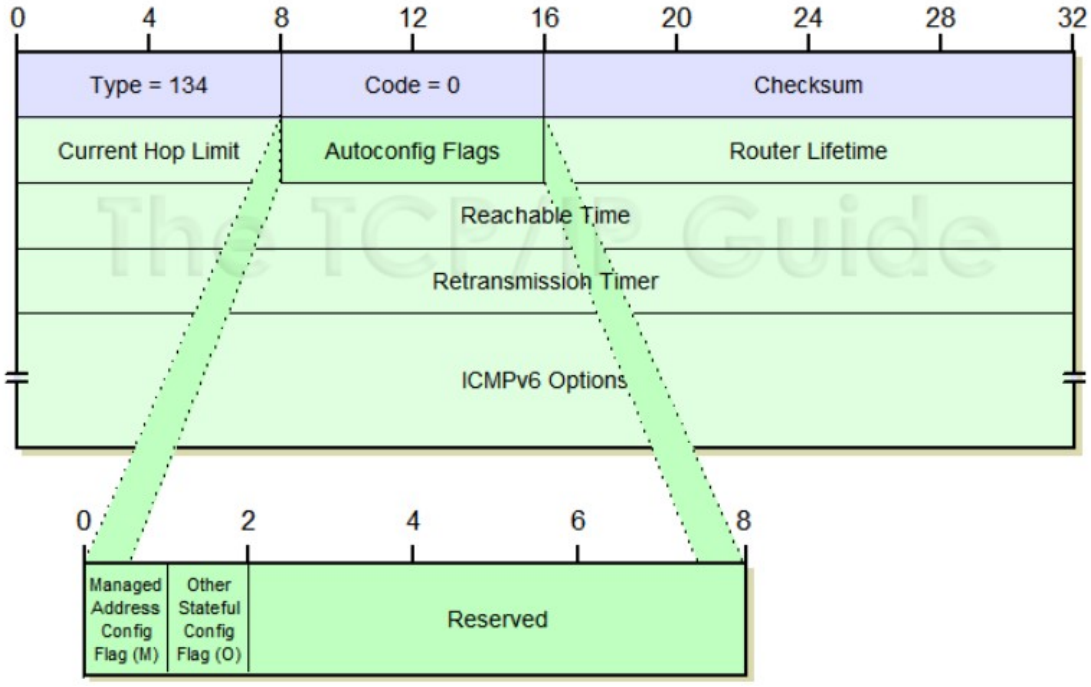
3. Link-Local Address Assignement: ak prejde adresa testom unikátnosti, danému interfaceu zariadenia je priradená daná link-local adresa, ktorá môže byť použitá pre komunikáciu v rámci lokálnej siete

4. Router Contact: Zariadenie načúva pre RA správy od routra, ktoré sú posielané periodicky alebo samotné zariadenie požiada o RA správu od routra pomocou RS správy

5. Router Direction: router informuje zariadenie ako má pokračovať v procese autokonfigurácie:
    - môže mu povedať, že “stateful” autokonfigurácia je v procese a poskytne mu IP adresu DHCP serveru
    - poskytne mu kroky na vygenerovanie globálnej IP adresy

6. Global Address Configuration: proces vygenerovania globálnej IP adresy samotným zariadením, väčšinou za pomoci sieťového prefixu, poskytnutého RA správou od routra a 64 bitovým identifikátorom interfaceu zariadenia(MAC adresa) alebo náhodne vygenerované ID

Router Advertisement(RA) správa:



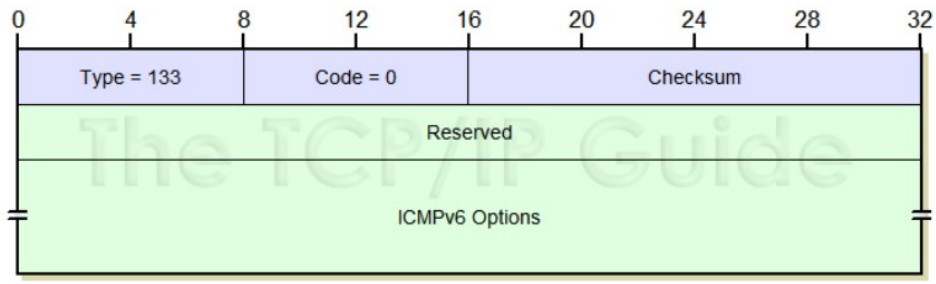
Autoconfiguration flags:

- M: informuje zariadenie aby použilo stateful metódu autokonfigurácie
- O: informuje zariadenie, aby použilo stateful metódy pre získanie dodatočných informácií okrem IP adresy

Router lifetime: informuje zariadenie, ako dlho by mal považovať router za default
Reachable time: informuje zariadenie, ako dlho by malo považovať suseda za dosiahnuteľného po obdržaní potvrdenia dosiahnuteľnosti
Retransmission timer: čas v milisekundách, koľko má zariadenie počkat pred preposlaním dát
ICMPv6 Options:

- source L2 Address: L2 adresa interfaceu routra, z ktorého je odoslaná RA
- MTU
- prefix information

Router Solicitation(RS) správa:



ICMPv6 options by mali byť zahrnuté ak zariadenie pozná svoju L2 adresu

Mobility Support:

- hlavná myšlienka: aj mobilné(prenosné) zariadenie je niekde “doma”
- používané adresy:
    - **Home Address**: globálna unicast adresa, cez ktorú je zariadenie vždy dostupné (aj keď nieje práve v domácej sieti)
    - **Care-of Address**: globálna unicast adresa mobilného zariadenia v cudzej sieti, adresa má tvar na základe siete v ktorej sa práve host nachádza
- **Correspondent Node(CN)**: zariadenie, ktoré komunikuje s mobilným zariadením
- **Home Agent(HA)**: router v domácej sieti, cez ktorý je mobilné zariadenie vždy dostupné
- **route optimization**: priama komunikácia mobilného zariadenia s CN (komunikácia môže inak prebiehať cez domáceho agenta = HA)
- prepojenie home adresy a care-of-adresy zariadenia sa nazýva binding

Return Routability Procedure(RRP):

- umožňuje Correspondend nodu(CN) overiť si že Mobile node(MN) je dostupné prostredníctvom home adresy a care-of-adresy, ktoré MN uvádza

1. MN pošle **Home Test Init(HoTI)** správu cez HA pre CN, čím sa CN dozvie o home adrese MN
2. MN pošle **Care-of Test Init(CoTI)** správu pre CN, čím sa CN dozvie o care-of adrese MN
3. CN odpovedá na HoTI správu so správou **Home Test(HoT)**, ktorú posiela cez HA
4. CN odpovedá na CoTI správu so správou **Care-of Test(CoT)**, ktorú posiela na care-of-adresu MN
5. MN aj CN vypočítajú 20 bytový Management Key, ktorý sa používa pre zabezpečenie Binding Update správ
    - keď zariadenie disponuje Management key, znamená to, že úspešne prešlo procesom RRP => preukázalo, že je dostupné cez obe adresy

General Security:

- sieťová bezpečnosť je založená na tzv. triádach **CIA** a **AAA**

CIA:
- **C**onfidentiality: čítať alebo meniť dáta nemôže neautorizovaná osoba
- **I**ntegrity: každá zmena dát musí byť zaznamenaná
- **A**vailability: dáta sú dostupné pre čítanie autorizovaným osobám stále

AAA:
- **A**uthentication: overenie identity
- **A**uthorization: overenie, že autentizovaná osoba má potrebné práva pre prístup k dátam, ku ktorým sa snaží pristúpiť
- **A**ccounting: zbieranie informácií za účelom analýzy

Bezpečnostné sieťové požiadavky musia byť zabezpečené pomocou základných bezpečnostných procesov:
- **encryption**: pre confidentiality
- **secure checksum**: pre integritu
a ich kombináciou.

Používané encryption metódy:
1. Secret Key Cryptography(symetrické šifrovanie): zdieľané secret
2. Public Key Cryptography(asymetrické šifrovanie): public/private kľúče

Security Association(SA):

- množina informácií, ktoré definujú použité šifrovacie algoritmy a kľúče pri zabezpečenej komunikácii
- jedno SA definuje zabezpečenú komunikáciu v jednom smere => pri duplex zabezpečenej komunikácii je potrebné použiť najmenej 2 SA
- SA je definované troma parametrami:
    - 1. Security Parameter Index(SPI): 32 bitový identifikátor SA
    - 2. IP Destination Address: adresa zariadenia, s ktorým bude prebiehať IPSec
    - 3. Security Protocol Identifier: udáva použitie AH alebo ESP

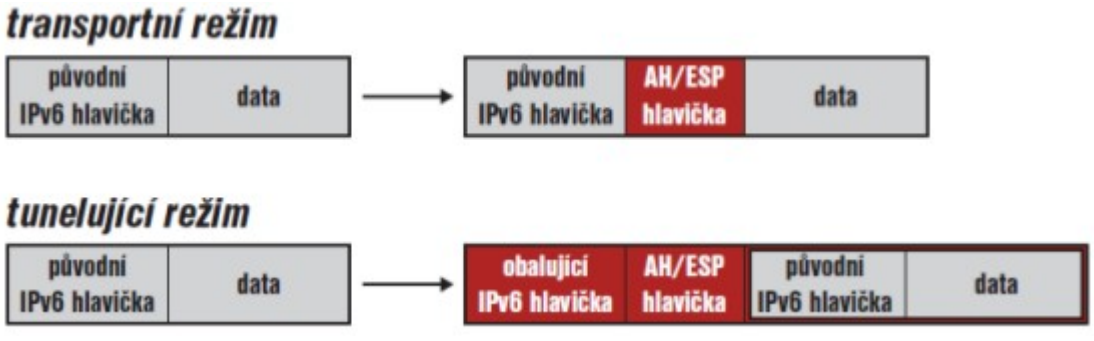
Internet Key Exchange version 2(IKEv2):

1. vytvorí zabezpečený kanál medzi komunikujúcimi zariadeniami pre potreby výmeny informácií o zabezpečení komunikácie a výmeny kryptografického materiálu
2. vytvorí zabezpečený kanál pre prenos dát (vytvorí pár IPSec AS => oba smery)
- predtým sa na tieto účely používali ISAKMP a IKEv1 protokoly

IPSec modes:

- IPSec rozlišuje medzi dvoma spôsobmi prenosu
1. **Transport mode:**
 - protokol zabezpečuje dáta z transportnej vrstvy
 - správa je vybavená AH/ESP hlavičkou, ktoré sú pridané pred hlavičku transportnej vrstvy (pred TCP/UPD hlavičku)
 - pred túto hlavičku sa vloží IP hlavička protokolom IP

2. **Tunnel mode:**
- zabezpečovaný je celý IP datagram, teda aj s už pridanou IP hlavičkou
- IPSec hlavičky sa pridajú pred už vloženú IP hlavičku a následne sa pridá ešte ďalšia IP hlavička pred IPSec hlavičky
- celý pôvodný IP datagram je obalený v ďalsom IP datagrame



Authentication Header(AH):

- IPSec protokol, poskytujúci authentication celého alebo časti IP datagramu
- hlavička obsahuje kontrolné dáta vypočítané zo skutočných prenášaných dát
1. medzi komunikujúcimi uzlami musí byť SA spojenie, len zdrojový a cieľový uzol vedia ako vypočítať kontrolné dáta
2. zdroj vykoná výpočet kontrolných dát, ktoré sa nazývajú Integrity Check Value(ICV) a vloží ich do hlavičky
3. cieľový uzol vykoná ten istý výpočet na prijatých dátach, na základe zdieľaného kľúča medzi týmito dvoma zariadeniami, čím hneď zistí prípadnu zmenu dát
- Authentication Header poskytuje overenie neporušenia dát ale nie súkromie dát

Encapsulating Security Payload(ESP):

- IPSec protokol, poskytujúci šiforavanie dát a tým pádom súkromie dát
- používa tri kompomenty:
    1. ESP Header: obsahuje SPI a Sequence Number a vkladá sa pred šifrované dáta
    2. ESP Trailer: vložené za šifrované dáta(zarovnáva šifrované dáta)
    3. ESP Authentication Data: keď sa použije autentizačná funkcia ESP protokolu, počíta sa vtedy ICV hodnota podobným spôsobom ako pri AH

AH vs ESP:
- AH vyžaduje menšiu výpočtovú silu ako ESP
- ESP používa silnejšie šifrovacie algoritmy
- AH autentizuje celý datagram
- ESP neautentizuje vonkajšiu IP hlavičku

Architektúry pre poskytnutie Quality of Service(QoS) vo forme prioritných dátových prúdov a garancie kvality prenosu:
1. **Integrated Services**
2. **Differentiated Services**

Integrated Services:
- aplikácia oznámi sieti svoje kvalitatívne požiadavky
- sieť zisťuje, či bude schopná splniť požiadavky
- ak nie, spojenie je zamietnuté (ak aplikácia nezníži svoje požiadavky)
- ak je sieť schopná vyhovieť požiadavkam aplikácie, informujú sa všetky zariadenia na ceste zo zdroja do cieľa o požiadavkách prenosu, používa sa protokol Resource reSerVation Protocol(RSVP) alebo YESSIR
- pri tejto architektúre je nutné udržiavať stav na vnútorných uzloch siete

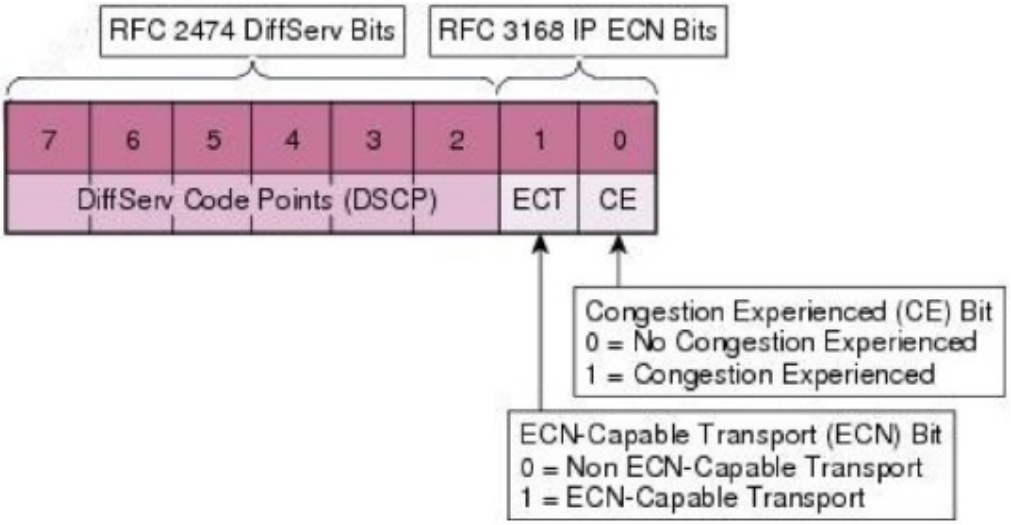
Differentiated Services:

- nepoužíva rezervačné protokoly a sieť sa neinformuje o požiadavkách na prenos
- každý packet je označený triedou priority

- packety sú označované pri vstupe do siete
- označenie sa vkladá do pola Type of Service (IPv4) alebo Traffic Class (IPv6)
- packety sú na vnútorných uzloch siete spracovávané podľa priorít
- dobrá škálovatelnosť, keďže sa nemusí udržiavať stav na vnútorných uzloch siete
- jednoduché na implementáciu
- žiadne inicializačné oneskorenie (initial delay) pre potreby rezervácií

V IPv6 hlavičke sa používajú 2 polia pre QoS:
1. **Traffic Class pole**
2. **Flow Label pole**

Traffic Class:
- označované niekedy ako Packet Priority(PRI)
- 1 bytové pole
- prvých 6 bitov je identifikátor tzv DS rutín, ktoré špecifikujú ako má byť packet poslaný
- posledné 2 bity sú tzv Explicit Congestion Notification(ECN) hodnoty, ktoré router posiela pri preťažení, predtým, ako nastáva packet loss(strata packetov)



Flow Label:
- 20 bitové pole, ktoré určuje príslušnosť packetu do dátového toku
- router si pri prvom packete toku uloží do cache pamäte IPv6 header, Routing header a Hop-by-Hop header, následne ich používa pre smerovanie ostatných packetov toho istého toku a nemusí stále čítať hlavičky z packetov
- IPv6 používa na overenie príslušnosti packetu do dátového toku len trojicu: Flow Label, Source address a Destination address, polia hlavnej IPv6 hlavičky, ktoré majú v hlavičke fixovanú pozíciu
- ak packet neprislúcha do žiadneho dátového toku, Flow Label je nastavený na 0

Mechanizmy koexistencie IPv4 a IPv6

1. **Dual Stack**: zariadenia a siete podporujú IPv4 aj IPv6 súčasne
2. **Tunneling**: IPv6 datagram je zabalený do dátového pola IPv4 datagramu
3. **Translators(NAT-PT)**: zariadenie preloží IPv6 datagram do IPv4 datagramu

1. Dual Stack:

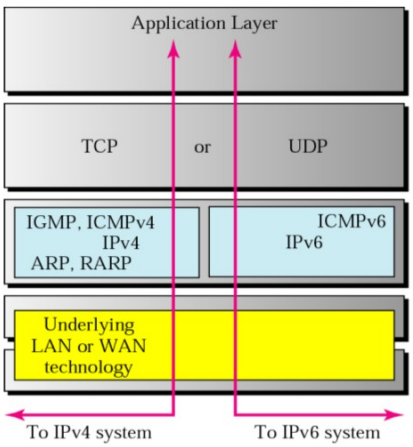
- jednoduché a flexibilné
- zariadenie môže komunikovať s IPv4 aj IPv6 zariadeniami
- keď sa globálne prejde na IPv6, IPv4 stack môže byť jednoducho odstránený
- dva samostatné protokolové stacky musia bežať súčasne(berie výkon)
- každá aplikácia musí byť schopná identifikovať verziu IP protokolu, ktorú používa uzol, s ktorým komunikuje
- DNS musí byť schopné pracovať s IPv4 aj IPv6 adresami

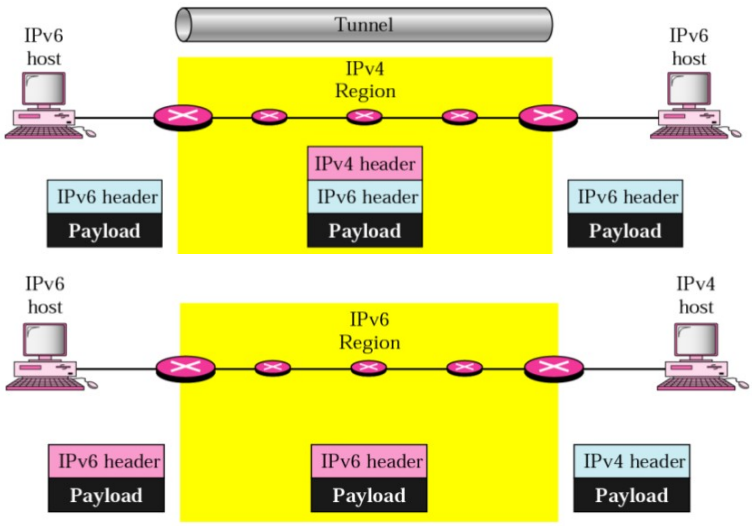
2. Tunneling

- keď sa globálne prejde na IPv6, niesú potrebné žiadne dodatočné zmeny
- niesú potrebné žiadne upgrady pre použitie tejto techniky
- dodatočná záťaž je pridaná na smerovače
- tunelovacie end-pointy môžu byť miestom zlyhávania alebo problémov ako MTU problémy alebo problémy s fragmentáciou
- tunely predstavujú potenciálnu hrozbu útokov
- zložitejšie z dôvodu enkapsulácie

3. Translators(NAT-PT):

- dočasné riešenie, keď nieje možné v danom čase implementovať Dual Stack ani Tunneling technológiu
- IPv6 zariadenie môže priamo komunikovať aj s IPv4 aj IPv6 zariadeniami
- nepodporuje pokročilé IPv6 funkcie ako end-to-end bezpečnosť
- limituje návrh topológie, odpoveď musí prísť cez ten istý NAT router, cez ktorý bola odoslaná požiadavka
- všetky aplikácie majúce IP adresu v payload sekcii packetu, sú zahodené
- NAT router predstavuje bod zlyhania





Interior Gateway Routing Protocol(IGRP):

- DV smerovací protokol vyvinutý Ciscom s cieľom vylepšiť nedostatky RIPv1
- nepodporuje VLSM
- updaty sú posielané na adresu multicast 224.0.0.10
- na výpočet ceny cesty používa bandwidth(B), delay(D), reliability(R), load(L) spolu s kladnými reálnymi koeficientami routru K1, K2, K3, K4, K5
- updaty obsahujú komponenty metriky a každý router sám počíta cenu cesty
- každý router musí mať totožne nastavené koeficienty K1..K5
- hľadanie najkratších ciest prebieha algoritmom Bellman-Ford

Extended Interior Gateway Routing Protocol(EIGRP):

- DV smerovací protokol vyvinutý Ciscom s cieľom vylepšiť IGRP
- podporuje VLSM
- výpočet ceny cesty zostáva rovnaký ako v prípade IGRP
- pôvodne podporoval len IPv4, IPv6 verzia pridaná neskôr
- hľadanie najkratších ciest prebieha na Diffusing computation

Intermediate System To Intermediate System(IS-IS):

- Link State smerovací protokol
- štandarzdizovaný organizáciou ISO pre komunikáciu sieťových zariadeni
- vyvinutý v rovnakom čase ako OSPF
- používa Hello Packety rovnako ako OSPF na vytvorenie susedských vzťahov a ich udržovanie
- podporuje VLSM
- udržuje rovnako ako OSPF link state databázu
- na hľadanie najkratších ciest používa Djikstrov algoritmus
- IS-IS packety sú zabalené priamo do L2 rámcov
- beží na vrchu L2 vrstvy, nezávislý na použití sieťovej adresy (IPv4, IPv6)
- poskytuje **overload declaration**, čo znamená, že pri hľadaní najvhodnejšej cesty sa preťažené routre neberú do úvahy
- cena cesty v rozsahu 0 – 63 (narrow metric), neskôr rozšírené na 0 – 16 777 215(wide metric)

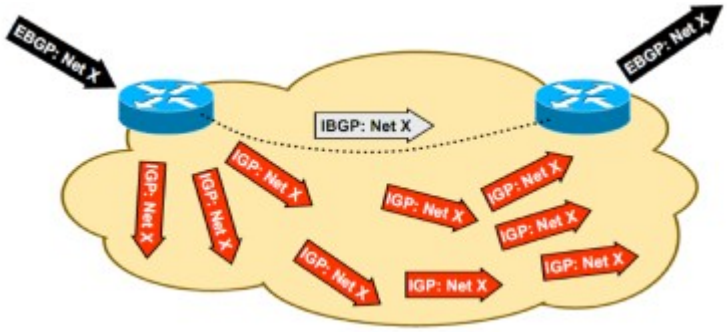
Border Gateway Protocol(BGP):

- Path Vector smerovací protokol aktuálne vo verzi 4 (BGP-4)
- navrhnutý pre potreby rastúceho Internetu
- používaný na komunikáciu medzi autonómnymi systémami
- vytvára spojenie pomocou TCP protokolu medzi hraničnými routrami AS
- poskytuje zavedenie smerovacích pravidiel
- ako metriku používa hop count
- používa CIDR notáciu pre zoskupovanie ciest
- hraničné smerovače si posielajú BGP správy cez TCP protokol a port 179

- správa obsahuje:
    - destination network address(CIDR notácia)
    - path attributes(AS systémy na ceste do cieľa)
- pri vyberaní najkratšej cesty sa pozerá na smerovaciu politiku a jej pravidlá, ktoré definujú, ktorý AS môže/nemôže smerovať packety cez ktorý AS (bizňis)
- BGP správy:
    - OPEN: zostavenie spojenia medzi BGP smerovačmi (predstavenie routrov)
    - UPDATE: update informácií o AS poslaný hraničným BGP routrom druhému
    - KEEPALIVE: keď neprebieha žiadny prenos, odlišuje od zlyhaného spojenia
    - NOTIFICATION: ukončenie spojenia alebo nahlásenie chyby
    - ROUTE-REFRESH: požiadavka updatu všetkých ciest v smerovacej tabuľke

Internal BGP(IBGP):

- slúži pre poskytnutie informácií o iných AS medzi vnútornými routrami jedného AS
- hraničný router pomocou IBGP pošle informácie získané pomocou eBGP(external BGP) svojim susedom, tí šíria tieto informácie ďalej už ale pomocou IGP



Router functions:

- router je zodpovedný za dva procesy:

1. Routing proces:
- na základe vymieňaných informácií medzi routrami pomocou smerovacích protokolov router rozhoduje o najlepších cestách, ktoré si ukladá vo **forwarding table**

2. Packet forwarding proces:
- presunutie packetu zo vstupného interfaceu(“ingress”), na ktorom packet prijal, na výstupný interface(“egress”), podľa informácií vo forwarding table
- výkon forwarding procesu udáva celkový výkon routra

Základné funkcie forwarding procesu routra:

- IP Header Validation: kontrola verzie protokolu, veľkosti hlavičky, checksum...
- Packet Lifetime Control: kontrola a dekrementácia TTL, generovanie ICMP pri chybe
- Checksum recalculation: prepočítanie header checksum po znížení TTL

- Route lookup: vyhľadávanie výstupného interfaceu pre daný packet na základe jeho destination adresy vo forwarding table routra
- Fragmentation: router musí paket “rozkrájať” ak veľkosť paketu je väčšia ako MTU výstupnej linky
- Handling IP Options: špecíalne zaobchádzanie s paketom definované v hlavičke

Zložitejšie funkcie forwarding procesu routra:

- Packet Classification: skúmanie zdrojovej adresy, cieľového/zdrojového portu ...
- Packet Translation: preklad adries ak router slúži ako brána pre NAT sieť
- Traffic Prioritization: prioritné spracúvanie prioritných paketov

Funkcie routing procesu routra:

- Routing protocols: implementácia smerovacích protokolov na výmenu informácií
- System configuration: implementácia funkcií, ktoré umožňujú konfiguráciu
- Router management: monitorovanie operácií smerovača

Router elements:

- Network Interfaces: poskytujú pripojenie k fyzickej linke(Ethernet, Sonet, ...)
- Forwarding Engines: rozhoduje, na ktorý výstupný interface má byť poslaný spracovávaný packet tým, že sa pozre do forwarding table
- Queue Manager: poskytuje buffer pre dočasné uloženie paketov, keď východzí interface, na ktorý majú byť poslané je momentálne obsadený, keď buffer pretečie, queue manager zahadzuje pakety
- Traffic Manager: má na starosti prioritizáciu a reguláciu východzích prenosov
- Backplane: prepája vstupný interface s výstupným
- Route Control Processor: zodpovedný za smerovacie protokoly, udržiava routing table, na základe ktorej je udržovaná forwarding table, riadi software na konfiguráciu a spravovanie routra, rieši errory, posiela ICMP správy

Address Lookup with CIDR:

1. Native Algorithms: hľadanie najväčšej zhody lineárne => O(n), kde n je počet prefixov vo forwarding table

2. Trie Based Algorithms: stromové vyhľadávanie
    - Binary Tries
    - Multibit Tries
    - Compressed Multibit Tries

3. Iné prístupy:
    - Search by Length algoritmy
    - Search by Value algoritmy
    - Hardwarové algoritmy: RAM-based lookup, Ternary CAM-Based lookup ...

Traffic Engineering:

- proces objavovania iných ciest v sieti ako sú najkratšie, aké cesty sa využívajú v danom momente a smerovanie prenosu cez cesty iné ako najkratšie z dôvodu optimálneho vyťaženia dostupných zdrojov siete

Simple Network Management Protocol(SNMP):

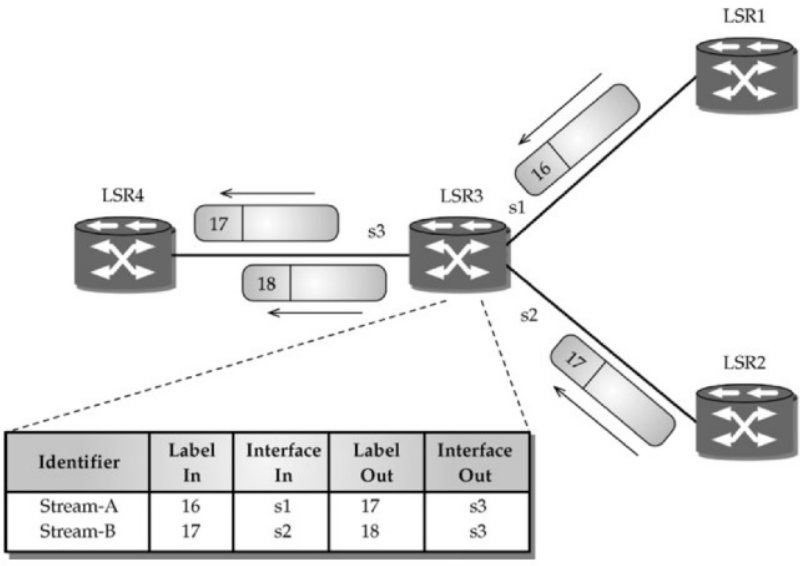
- získava informácie od všetkých routrov, tieto informácie následne využije pre prehľad využívania zdrojov v sieti
- nerozhoduje, ktorý tok by mal byť prerozdelený inými cestami pre odľahčenie zaťažených miest, len akýsi náhľad prenosovej záťaže je získaný

NetFlow:

- nástroj firmy Cisco, pre zber informácií o sieťovom prenose na kľúčových uzloch
- kľúčové body zhromažďovania informácií sú kolektory(collectors)

MultiProtocol Label Switching(MPLS):

- smerovací mechanizmus, v ktorom sú packety smerované na základe popisu, ktorý je pridaný pred packet ako nová hlavička
- popis je pridaný packetu pri vstupe do MPLS-schopnej siete
- router nemusí vyhladávať destination IP v tabuľke ale smeruje packet na základe MPLS popisu, čo je rýchlejšie
- pre šírenie MPLS informácií sú potrebné nové protokoly alebo rozšírenia starých
- MPLS používa conection oriented spojenie, packety sú prenášané cez ustavené **Label Switched Paths(LSP)** spojenia, ktoré sú jednosmerné => pre obojsmernú komunikáciu sú potrebné 2 LSP spojenia
- umožňuje sieťovým správcom určiť cestu packetov
- router obsahuje tabuľky Label Information Base(LIB) alebo Label Forwarding Information Base(LFIB) v ktorých má namapovaný vstupný interface + vstupný popisok na výstupný interface + výstupný popisok, ktorý pred odoslaním packetu nastaví packetu namiesto jeho vstupného popisku
- medzi dvoma routrami môže byť zostavených viac paralelných LSP spojení a prenos dát môže byť medzi ne rozdelený



Identifier	Label In	Interface In	Label Out	Interface Out
Stream-A	16	s1	17	s3
Stream-B	17	s2	18	s3

Edge Label-Switched Routers(Edge-LSR):

- hraničné MPLS routre, Ingress = vstupný, Egress = výstupný

Ingress-LSR

- analyzuje hlavičku IP paketu
- na základe analýzy hlavičky je paketu priradená Forwarding Equivalence Class(FEC) trieda
- na základe FEC triedy je paketu priradený popisok(label) do MPLS hlavičky

Egress-LSR

- odstráni paketu MPLS hlavičku, ktorú už nebude potrebovať, keďže paket opúšťa MPLS-schopnú sieť a smeruje paket na príslušný východzí interface ďalej
- dekrementuje TTL

Core Label-Switched Routers(Core-LSR):

- smerujú pakety na základe MPLS popisov
- na týchto routroch IP hlavičky niesú čítané ani modifikované

Distribúcia MPLS popisov:

- ešte pred zostavením LSP spojení, LFIB musí existovať na každom LSR na ceste paketu pomocou label distribution protocol
- ako label distribution protocol sa môže použiť: BGP, RSVP-TE, LDP, TDP, LDP/CR

Label Distribution Protocol(LDP):

- protokol na šírenie popiskov v MPLS prostredí
- môže pracovať v režime Downstream-on-demand alebo Downstream-unsolicited
- používa správy typu:
    - LDP Discovery message
    - LDP Adjacency message
    - LDP Label Advertisement message
    - LDP Notification message

Generalized MPLS(GMPLS):

- rozširuje klasické MPLS o funkcionality ako:
    - Time-Division Multiplexing
    - Lambda-Switching
    - Fiber-Switching

QoS-Based Routing:

- proces smerovania, pri ktorom sú cesty pre dátové toky vyberané aj na základe dodatočných informácií o sieti ako dostupnosti zdrojov a QoS požiadavok

QoS smerovacie protokoly:

1. Source-based routing algorithms
- každý router má informácie o stavoch v celej sieti
- cesta paketu je rozhodnutá lokálne na jednom routri
- problémy so škálovatelnosťou, routre musia držať vela informácií
- preťaženie routra, ktorý rozhoduje o celej ceste

2. Hop-by-hop routing algorithms
- každý router rozhoduje len o next-hop destinácii
- záťaž je rozdelená na všetky routre rovnomerne
- problémy s loop
- tiež problémy so škálovatelnosťou

3. Hierarchical routing algorithms
- používa viac levelov
- najnižší level tvoria skutočné routre
- skutočné routre sú zoskupené do logických skupín, ktoré tvoria ďalší level
- tieto skupiny môžu byť organizované do skupín vyšších vrstiev
- ucelené smerovacie informácie sú na hraničných routroch každej skupiny
- každý router obsahuje informácie o svojej skupine a o iných skupinách
- nemá problém so škálovatelnosťou, je vhodný aj do veľkých sietí
- zoskupovanie znižuje presnosť informácií o sieti

Private Network-Network Interface(PNNI):

- hierarchický dynamický smerovací protokol pre ATM siete
- založený na link-state
- je hierarchický => používa koncept levelov a logických zariadení
- podporuje zoskupovanie topológie a informácií o dostupnosti

QOSPF:

- QoS rozšírenie protokolu OSPF
- pre každý dosiahnuteľný cieľ vyráta widest-shortest path, čo je cesta, s najmenším počtom hopov a najvyššou bandwidth

Rozšírenia TCP protokolu:

- GridDT: skupina ad-hoc modifikácií
- Scalable TCP
- High-Speed TCP(HSTCP)
- H-TCP
- BIC-TCP
- CUBIC-TCP
- QuickStart(QS)

- E-TCP
- FAST

QuickStart(QS):

- 4 bytová hodnota v IP hlavičke, ktorá pozostáva z QS TTL a Initial Rate(IR) polí
- odosielateľ, ktorý chce použiť QS, nastaví QS TTL na dostatočne vysokú hodnotu, a Initial Rate na požadovanú rýchlosť, na akej chce začať posielanie
- každý router na ceste, ktorý podporuje QS, zmenší QS TTL a zníži IR ak musí
- príjemca odošle QS TTL a IR v SYN/ACK pakete odosielateľovi
- odosielateľ na základe porovnania QS TTL a TTL zistí či všetky routre na ceste podporujú QS
- odosielateľ nastaví cwnd a začne používať congestion control mechanizmus
- vyžaduje si zmeny v IP vrstve

E-TCP:

Early Congestion Notification(ECN)
- bit, ktorý je nastavený keď nastáva zahltenie linky/fronty/bufferu
- TCP musí reagovať na ECN ako na paket loss

FAST:

- používa end-to-end delay, ECN a paket lossy pre detekciu congestion(zahltenia)
- ak RTT monitoringom zistí málo paketov vo fronte routra =>zvýši sending rate

Client-Server architektúra:

- pozostáva z 2 typov softvérových modulov
1. Server module
2. Client module

1. Server module

- pasívna časť architektúry
- pasívne počúva požiadavky klientov
- viac klientskych požiadaviek môže byť vybavených spôsobmi:
    1. postupne
    2. súčasne pri viac vláknových serveroch
    3. viacerými serverami na rôznych miestach
- pozastavené požiadavky klientov môžu byť ukladané

2. Client module

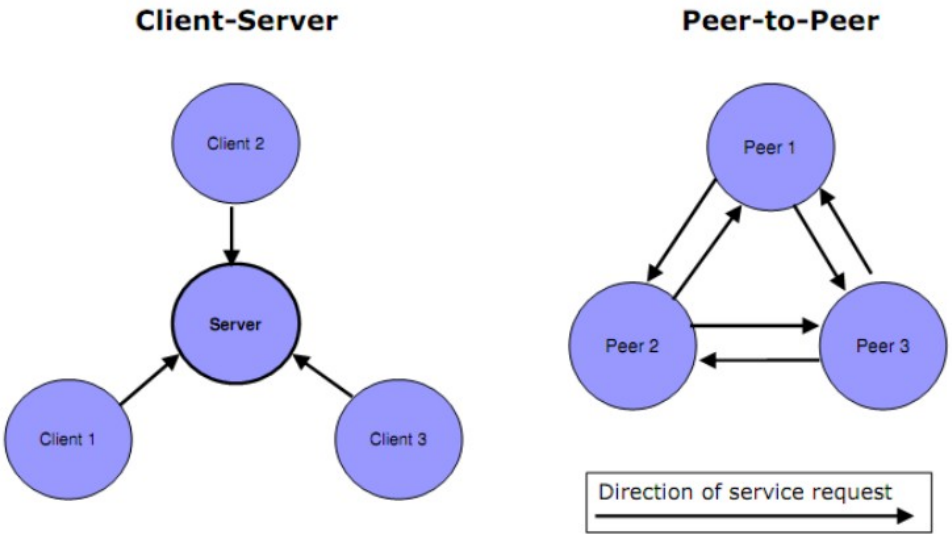
- aktívna časť architektúry
- aktívne posiela požiadavky na server
- klienti nekomunikujú priamo medzi sebou
- musí poznať sieťovú adresu a port serveru

Príklady client-server systémov:

- web server/web browser
- SSH/Telnet/FTP server/klienti

Peer-to-peer architektúra:

- pozostáva z identických softvérových modulov(peers)
- peers môžu medzi sebou komunikovať priamo
- každý peer je serverom a zároveň aj klientom
    - v stave server je, keď poskytuje služby iným peerom
    - v stave klient je, keď požaduje služby od iných peerov



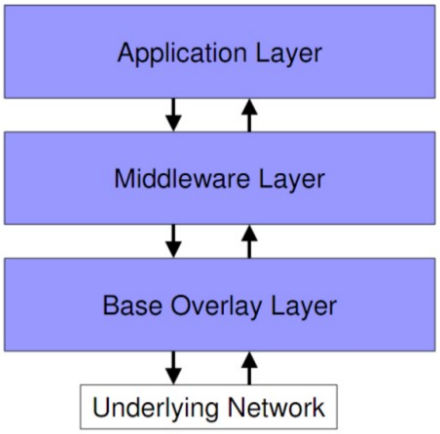
Client-Server vs Peer-to-peer:

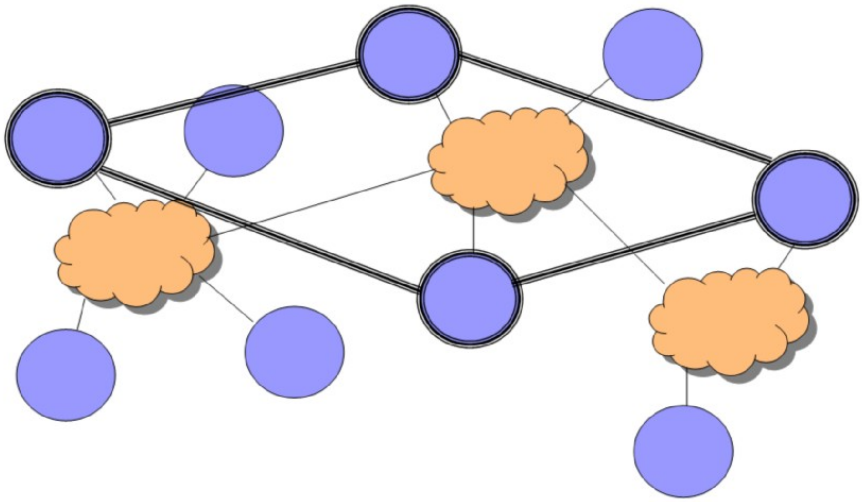
- C-S architektúra je známejšia a rozšírenejšia
- C-S má jednoduchšiu interakciu medzi modulmi
- C-S je jednoduchší na udržovanie a správu
- C-S škálovatelnosť je limitovaná hardvérom serveru, čo môže byť ale riešené load balancingom medzi viaceré servery za istú cenu
- P2P škálovatelnosť je zaručená sama o sebe, ako stúpa počet peerov, stúpa aj kapacita hardvérová
- C-S bezpečnosť je zaručovaná na strane serveru, P2P je rozdelená medzi peerov
- C-S spolahlivosť je založená na použití viacerých záložných serverov s automatickou náhradou pri výpadku
- P2P spolahlivosť je “automatická”, viac peerov väčšinou ponúka tú istú službu, keď teda jeden vypadne, môže odber pokračovať od ďalšieho

Overlay network:

- sieť(virtuálna), ktorá je postavená na vrchu existujúcej(fyzickej) siete
- umožňuje pridanie funkcionalít bez nutnosti redisegnu siete
- na prepájanie zariadení používa overlay network funkcie underlying siete, teda siete, na ktorej “stojí”, funkcie ako (TCP, UDP)
- typické overlay siete: P2P, Cloud

P2P architektúra siete:





Base Overlay Layer:

- zodpovedá za hľadanie nových peerov
- zodpovedá za udržovanie spojení medzi peermi
- zodpovedá za komunikáciu medzi peermi

Middleware Layer:

- security: riadi prístup k službám a zdrojom poskytovaným peermi
- service/resource discovery: vyhľadáva a indexuje služby a zdroje medzi peermi
- peer groups: zoskupuje peerov poskytujúcich alebo odoberajúcich tie isté služby alebo zdroje

Application Layer:

- využíva služby middleware vrstvy pre vytvorenie konkrétnych aplikácií
- napr: zdieľanie súborov, smerovacie protokoly, instant messaging, ...

Overlays and Peer discovery:

- zariadenie, ktoré sa chce pridať do existujúcej P2P siete, musí lokalizovať aspoň jedného peera, ktorý do tejto siete patrí

1. Static Configuration
2. Centralized directory
3. Member Propagation Techniques with Initial Member Discovery

1. Static Configuration:

- každému peeru je staticky zadaný list peerov v podobe IP adresy a portu
- peer sa pokúša skontaktovať s niektorými peermi z tohto listu
- kvôli pracnej manuálnej konfigurácii je tento prístup vhodný pre menšie P2P siete

2. Centralized directory:

- každý peer pozná IP adresu centralizovaného serveru
- každý peer na začiatku pripojenia kontaktuje server, ktorý ho označí za aktívneho peera a dostane od neho list aktívnych peerov
- prevažná väčšina komunikácie potom už prebieha mimo servera
- server výnimočne poskytuje aj iné služby ako napr: ktorý peer obsahuje ktoré súbory atď
- server je single point of failure

3. Member Propagation Techniques with Initial Member Discovery:

- po objavení prvého peera patriaceho do P2P siete sa o ostatných dozvedá od neho a od ďalších získaných prostredníctvom prvého objaveného

Overlay Network Topology:

- topológia overlay siete ovplyvňuje výkon P2P siete
- hlavné faktory ovplyvňujúce výkon P2P siete:
    1. **Diameter**: najdlhšia vzdialenosť medzi dvoma peermi
    2. **Average Degree**: priemerný počet spojení jedného peera
- topológie overlay siete:
    1. Random Mesh
    2. Tiered
    3. Ordered Lattice

1. Random Mesh:

- peer objaví niekoľko iných peerov a pripojí sa k nim
- vhodné pre prepájanie veľkého počtu peerov
- pre hľadanie zdrojov a služieb u peerov v sieti sa môže použit search message flood, čo je jednoduché ale produkuje to veľkú záťaž

2. Tiered:

- peery sú usporiadané do úrovní na základe ich zdrojov a konektivity
- tier0 je základná úroveň obsahujúca spoľahlivých peerov s overenými službami a zdrojmi
- na každej úrovni je každý peer prepojený s niekoľkými peermi na nižšej úrovni, a komunikuje s peermi vo vyššej aj nižšej vrstve
- listové peery (leaf peers), peer na najnižšej úrovni, sú pripojené len k svojim super-peerom, nezúčastňujú sa na komunikácii iných peerov a sú tiež vynechané z procesu peer discovery

3. Ordered Lattice:

- peeri sú usporiadané do do obdĺžnikovej mriežky
- každý peer je priamo pripojený k 4 susedným peerom
- peeri na opačných koncoch mriežky sa môžu prepojiť, aby vytvorili uzavretie
- koordinácie v mriežke môžu byť použité ako identifikátor v **C**ontent **A**ddresable **N**etworks (CAN), tiež označované ako **D**istributed **H**ash **T**able (DHT)

Service/Resource discovery:

- peer musí oznamovať svoje poskytované služby aby mohli byť lokalizované ostatnými peermi, požadujúcimi dané služby
- prístupy service/resource discovery:
    - 1. **Centralized**: požiadavka na server o lokalizovanie služby, priestorová zložitosť: O(n) vzhľadom k počtu peerov
    - 2. **čisté P2P**: požiadavka je poslaná všetkým (flood), v najhoršom prípade sa posiela až O(n) discovery správ

Základné rozdelenie P2P systémov:

1. Centralized
2. Decentralized
3. Hybrid

1. Centralized:

- kombinuje funkcie centralizovaných (klient-server) systémov a decentralizovaných
- obsahuje 1 alebo viac centrálnych serverov, ktoré slúžia na lokalizáciu požadovaných služieb alebo na plánovanie úloh peerov
- keď už peer vie, od koho môže požadovanú službu získať, komunikuje už priamo s daným peerom a nekomunikuje cez server
- servery predstavujú úzke hrdlo (bottleneck) pre P2P siete s veľa peermi
- náchylné na útoky a single point of failure

2. Decentralized:

- každý peer má rovnaké práva a zodpovednosti
- každý peer má len čiastočný náhľad P2P siete
- problém rýchlelo lokalizovania peera s požadovanými službami
- imúnny voči single point of failure
- žiadna limitácia alebo úzke hrdlo, vysoký výkon
- rozdelenie logických topológií:
    - 1. Structured
    - 2. Unstructured

1. Structured:

- funguje tu akýsi vzťah medzi dátami a peermi
- peer poskytujúci dáta je na základe dát vyhľadateľný (DHT)
- tieto systémy vedia garantovať vyhľadania za cenu držania dodatočných informácií

2. Unstructured:

- nieje žiadny vzťah medzi dátami a peermi
- každý peer sa stará len o svoje dáta a max. o úzke okolie susedov, ktorým bude pravdepodobne v budúcnosti posielať požiadavky
- nieje garantovaná odpoveď na požiadavku/nájdenie peera s požadovanými dátami

3. Hybrid:

- berie hlavnú výhodu centralizovaných P2P systémov: rýchle lokalizovanie peera s požadovanými dátami/službami (nevýhoda: limitácia v podobe škálovatelnosti)
- berie hlavnú výhodu decentralizovaných P2P systémov: škálovatelnosť (nevýhoda: potrebný dlhší čas na lokalizovanie peerov na základe služieb a dát)
- pre zachovanie škálovatelnosti: nepoužíva centrálne servery
- používa tzv super-peers, peery, ktoré slúžia ako servery pre ostatné peery
- lokalizovanie peera na základe služieb a dát môže byť robené oboma spôsobmi: pomocou decentralizovaného vyhľadávania aj centralizovaného vyhľadávania
- Fungovanie:
    - 1. peer svoju požiadavku pošle superpeerovi, pod ktorý spadá
    - 2. príslušný superpeer v spolupráci s ostatnými superpeermi lokalizujú superpeer, pod ktorý spadá peer s požadovanými službami/dátami

Routing in P2P networks:

- efektivita smerovania v P2P sieti môže byť meraná viacerými metrikami:
1. **Storage**: úložný priestor použitý pre metadata (hľadanie peerov)
2. **Efficiency**: rýchlosť nájdenia peera s požadovanými službami/dátami, metrika efektivity je doba odozvy (respons time)
3. **Usability**: jednoduchosť a typy použiteľných požiadaviek (query)
4. **Coverage**: či prehľadávaný priestor obsahuje odpovede
5. **Scalability**: použiteľnosť s veľkým počtom zariadení

Smerovanie v unstructured P2P sieťach:

- flood techniky posielania požiadaviek sa používajú
- TTL hodnoty sú priraďované každej požiadavke aby sa predišlo zahlteniu systému
- používané smerovacie stratégie:
    1. Breadth-First Search(BFS): napr. Gnutella
    2. Depth-First Search(DFS): napr. FreeNet
    3. Heuristic-Based Routing Strategies

1. Breadth-First Search(BFS):

- posielanie požiadaviek zahltením (flood)

2. Depth-First Search(DFS):

- požiadavku pošle len jednému najpotenciálnejšiemu susedovi, ak do stanoveného času od neho nepríde odpoveď, pošle ďalšiemu najvhodnejšiemu kandidátovi

3. Heuristic-Based Routing Strategies:

1. Iterative deepening:
- požiadavka je zasielaná klasickou BFS metódou s postupným rozširovaním radiusu dosahu požiadavky
- proces hľadania končí úspechom alebo dosiahnutím najväčšej možnej hĺbky

2. Directed BFS and Intelligent search:
- oproti klasickému BFS, pri ktorom zariadenie preposiela požiadavku všetkým svojím susedom, v tomto prípade zariadenie posiela požiadavku len podmožine svojich susedov, ktorých inteligentne vyberá na základe štatistík
- výhoda: redukcia počtu správ posielaných oproti klasickému BFS
- nevýhoda: štatistiky o susedoch sú väčšinou stručné a nedostatočné

3. Local Indices search:
- každé zariadenie si drží zoznamy svojich lokálnych dát a dát svojich susedov, nachádzajúcich sa v rámci k hopov od neho
- výhoda: spracovanie požiadaviek na menšom počte zariadení
- nevýhoda: väčšia pamäťová náročnosť pre zoznamy
- nevýhoda: väčšia cena updatov zoznamov

4. Random Walk:
- zariadenie preposiela požiadavku náhodne vybranému susedovi
- proces končí úspechom alebo dosiahnutím maximálneho TTL
    - k-Walker algorithm: tvorca požiadavky posiela požiadavku náhodne vybranej podmnožine svojich susedov, každý zo susedov už potom posiela len jednému náhodne vybranému susedovi ako v pôvodnom algoritme, počet odoslaných správ stúpa lineárne vzhľadom k pôvodnému algoritmu
    - Random Breadth-First Search(RBFS): každý z peerov posiela požiadavku náhodne vybranej podmnožine svojich susedov, počet správ stúpa exponenciálne vzhľadom k pôvodnému algoritmu

5. Adaptive Probabilistic Search(APS):
- každý peer obsahuje pravdepodobnosti úspechu pre každého svojho suseda, ktoré si aktualizuje a udržiava na základe minulých prenosov
- peer posiela požiadavku susedovi s najlepšou pravdepodobnosťou
- prístupy aktualizácie pravdepodobností susedov:
    - 1. Optimistic approach: peer stále zvyšuje pravdepodobnosť úspechu vybraného suseda a znižuje ju len vtedy, keď požiadavka poslaná tomuto susedovi sa skončí failom
    - 2. Pesimistic approach: peer stále znižuje pravdepodobnosť úspechu vybraného suseda a zvyšuje len vtedy, keď požiadavka poslaná tomuto susedovi skončí úspechom

6. Interest-Based Shortcuts:
- každý peer má dodatočné linky k peerom, ktoré majú podobné “záujmy”
- tieto dodatočné linky sú tzv shortcuts (skratky)
- keď peer vytvorí požiadavku, najprv ju pošle na svoje evidované skratky, ak hľadanie nieje úspešné, normálny algoritmus je použitý
- po každom úspešnom zodpovedaní požiadavky peera, si peer uloží shortcuts na peery, ktoré mu poskytli odpovede na jeho požiadavku

Smerovanie v structured P2P sieťach:

- garancia, že ak odpoveď na dotaz v sieti existuje, bude nájdená, často veľmi efektívne v čase O(log N)
- nevýhoda: veľká réžia v podobe udržovania väčšieho počtu informácií na uzloch
- v štrukturovaných P2P sieťach sa peeri organizujú do rôznych topológií:
    - 1. Chord: kruhová
    - 2. CAN: multidimenzionálna mriežka
    - 3. mesh : Pastry a Tapestry
    - 4. Skip Graph: viacnásobný zoznam
- na základe overlay štruktúry sa štrukturované P2P systémy rozdeľujú na:
    - 1. Distributed Hash Table(DHT) based systems (Chord, CAN, Tapestry...)

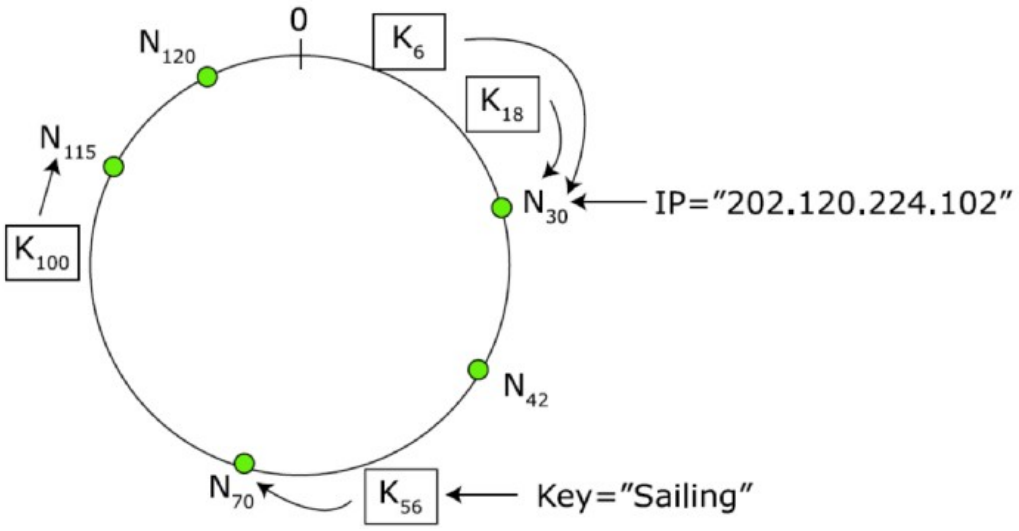
- 2. Skip List based systems (Skip Graph, SkipNet)
- 3. Tree based systems (P-Grid, P-Tree, BATON)

Distributed Hast Table(DHT) based systems:

- každé zariadenie si spravuje istú časť globálnej hašovacej tabuľky
- získanie položky A znamená dotazovanie zariadenia, ktoré spravuje tú časť hašovacej tabuľky, ktorá obsahuje hash(A)

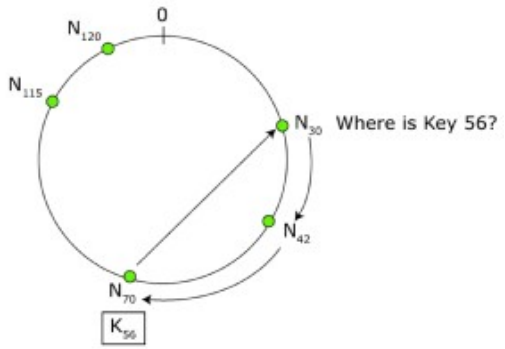
1. Chord:

- používa hašovaciu funkciu pre mapovanie IP adresy zariadenia na m-bitový identifikátor
- používa hašovaciu funkciu pre mapovanie dátovej položky alebo kľúča dátovej položky na m-bitový identifikátor
- identifikátory 0 – 2^{m-1} sú usporiadané v kruhu
- systém priradí identifikátor m zariadeniu alebo dátam, ktorých identifikátor je rovný m alebo je v smere hodinových ručičiek najbližšie

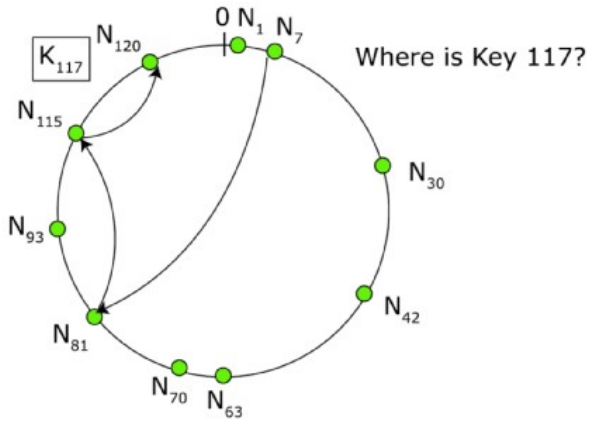


Simple lookup algorithm:

- každý peer pozná len svojho nasledovníka/peer v smere ručičkových hodiniek
- keď obdrží požiadavku, prezrie svoje lokálne dáta, či nemá odpoveď
- ak nemá, posiela to nasledovníkovi
- končí, keď identifikátor prehľadávaného uzla presiahne identifikátor hľadaných dát
- časová zložitosť O(N)



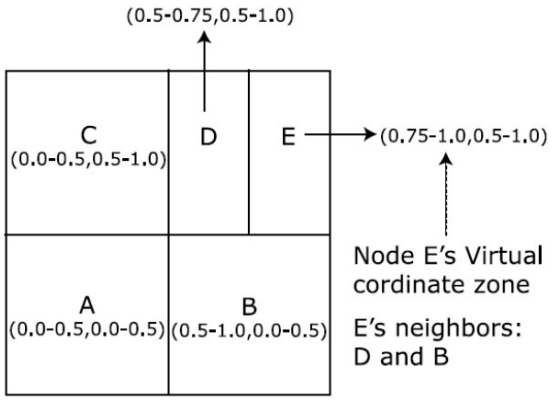
Scalable lookup algorithm:
- každý peer má tzv finger table, v ktorej má uložených viacero nasledovníkov
- keď dostane požiadavku, prezie svoje lokálne dáta, či nemá odpoveď
- ak nemá, hľadá vo finger table vhodného nasledovníka takého, ktorý ma najvyšší identifikátor spĺňajúci vzťah: n(identifikátor uzla) < k(identifikátor hľadaných dát), ak taký uzol v tabuľke nemá, prepošle požiadavku svojmu najbližšiemu nasl.
- končí, keď identifikátor uzla presiahne identifikátor dát
- časová zložitosť O(log N)

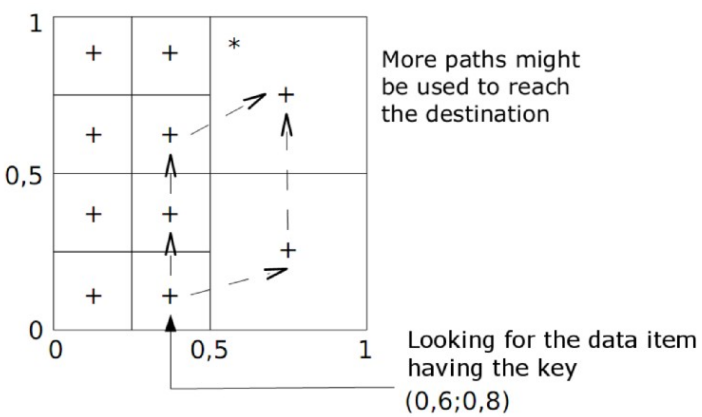


Keď do siete vstúpi nový peer, musí:
1. Nájsť svoje miesto v kruhu
2. Obdržať dáta na základe identifikátoru, za ktoré bude zodpovedný
3. Získať svoju finger table
4. Aktualizovať finger table u susedov, aby o ňom vedeli

Keď zo siete odíde zariadenie, nieje potrebné robiť nič

2. Content Addresable Network (CAN):
- rozdelenie úložného priestoru do viacerých zón, kde každá zóna je priradená pod správu iného zariadenia
- zariadenie spravuje všetky dáta spadajúce do jeho spravovanej zóny
- používa sa virtuálny n-dimenzionálny priestor pre identifikáciu zón
- používa sa hašovacia funkcia pre zahašovanie dát do identifikátoru v podobe koordinácií v n-dimenzionálnom priestore => teda do n-tice
- každý uzol si drží informácie o svojich susedoch, tzn o susediacich zónach





3. Pastry:
- smerovací systém, založený na PRR stromoch
- dáta sú uložené na uzle, ktorého identifikátor zdieľa najdlhší prefix s identifikátorom dát
- v každom kroku smerovacieho procesu je vybraný sused, ktorý má dlhší spoločný prefix s cieľovým uzlom

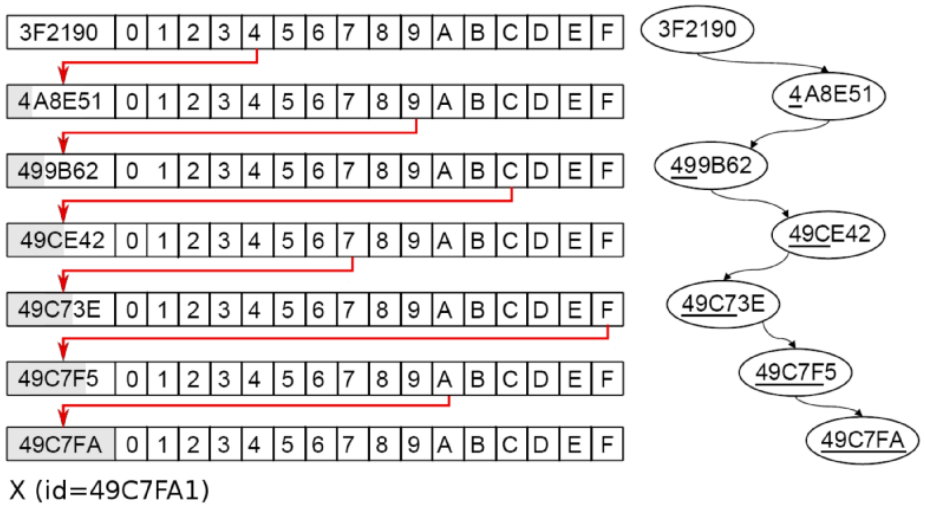


Figure: An example of inserting/searching an object X with an identifier $49C7FA1$ into a Pastry network (starting at the node $3F2190$).

4. Tapestry:
- veľmi podobný systému Pastry
- Pastry v každom smerovacom kroku rozširuje zhodu prefixu, Tapestry v každom smerovacom kroku rozširuje zhodu suffixu

Skip List based systems:
- skip list je dátová štruktúra pre ukladanie zoradených listov dát, ktorá používa hierarchiu zreťazených zoznamov
- vrstva 0 je klasický zoradený reťazený zoznam
- element vrstvy i sa objaví vo vrstve i+1 s fixnou pravdepodobnosťou
- očakávaná zložitosť nájdenia dátovej položky je O(log N)

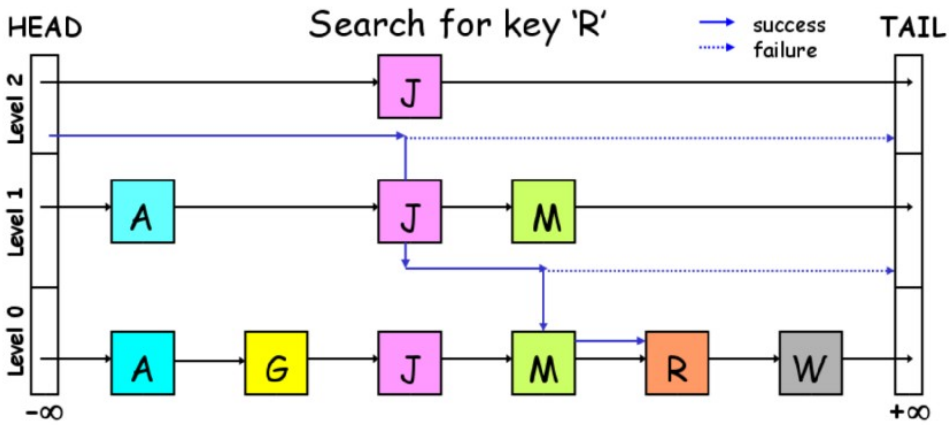


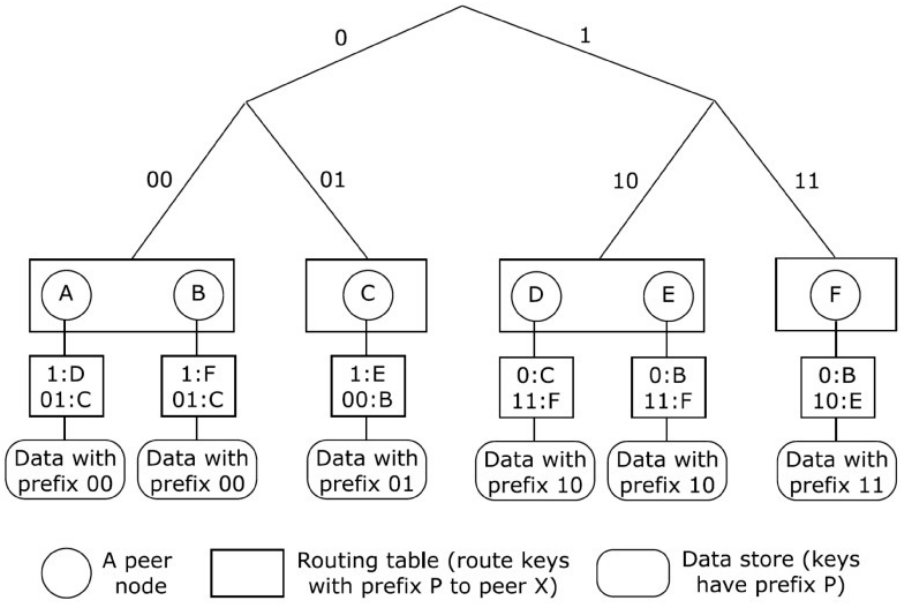
Figure: The searching process in a Skip List structure.

1. Skip-Graph:
- smerovanie na základe skip listov
- modifikuje obyčajný skip list, pretože top-level uzly by boli preťažené
- na každom leveli používa skip-graph viacero reťazených zoznamov
- každý uzol sa vyskytuje v zozname na každej úrovni
- počet levelov je O(log N)
- časová zložitosť odpovede na požiadavku je O(log N)

2. SkipNet:
- podobný princíp ako Skip-Graph
- miesto skip-listov používa na organizovanie uzlov kruhy
- kruhy sú organizované do levelov ako v skip-graph skip listy
- uzly sú na každom leveli zoradené podľa dát
- na každej úrovni si uzol udržiava ukazateľe na susedov v pravo a vlavo
- všetky uzly sú prepojené cez kruh na úrovni 0

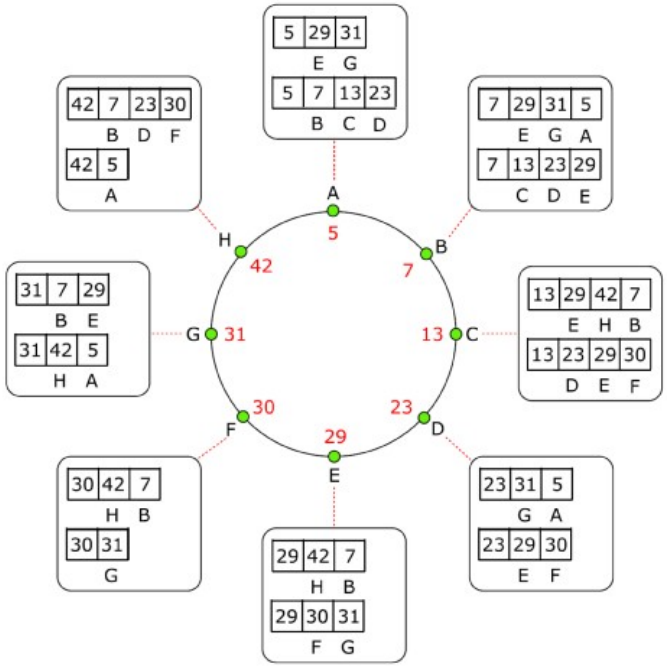
Tree based systems:

1. P-Grid:
- ako štruktúru využíva virtuálny binárny strom
- každý peer ma identifikátor, ktorý pozostáva z reťazca bitov pozdĺž cesty z koreňa stromu do daného peera
- viacerým peerom môže byť priradený rovnaký identifikátor
- každý peer si udržiava smerovaciu tabuľku
- proces vyhľadávania má zložitosť: $O(\log_2 N)$



2. P-Tree:

- ako štruktúru používa virtuálny vyvážený B+ strom postavený na vrchu Chord



3. BATON:

- oproti predošlým dvom má BATON 2 hlavné vylepšenia:
    - 1. dáta sú uložené aj v listoch aj na vnútorných uzloch
    - 2. okrem rodič-potomok liniek, používa BATON aj susedné linky
- susedné linky sa používajú pre prepojenie uzlu s uzlom, ktorý obsahuje dáta so susedným rozsahom dát, ktoré ukrýva daný uzol
- susedné linky sa používajú pre prepojenie uzlu s jeho susedom na rovnakej úrovni
- susedné linky pomáhajú pred úzkym hrdlom v mieste koreňa

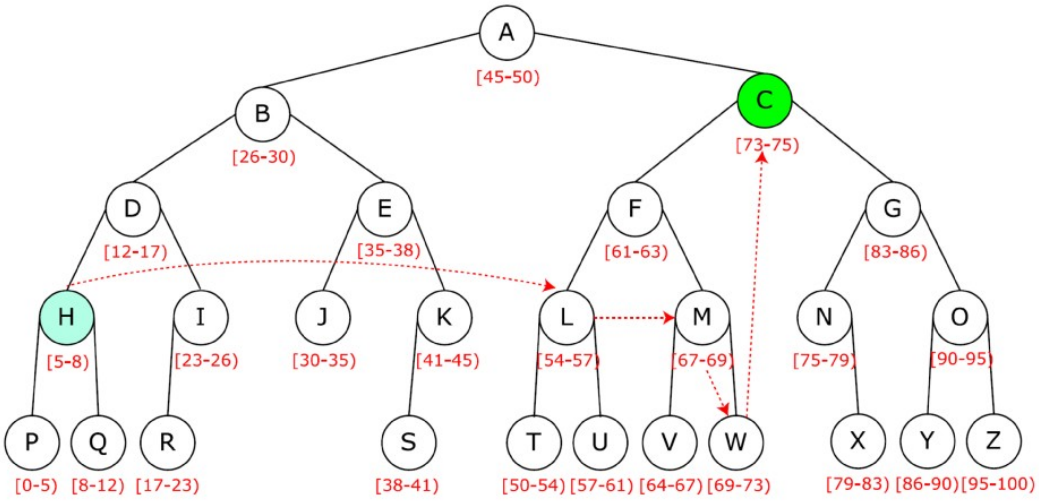
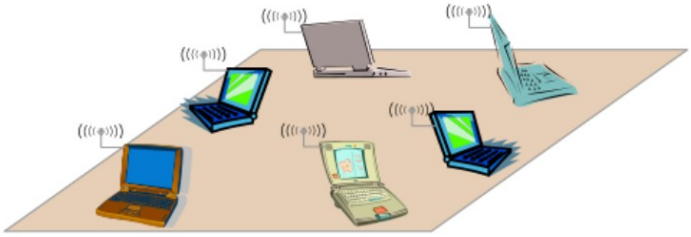


Figure: A lookup example in BATON: the node *H* wants to search for a data item (having the key 74) stored in the node *C*.

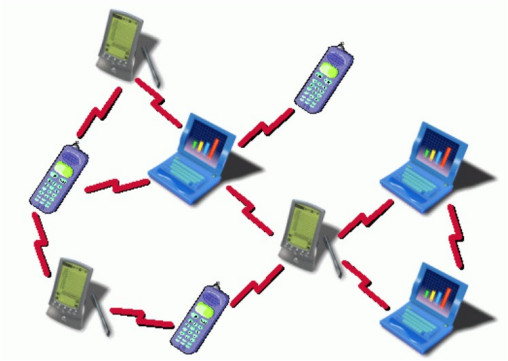
	Structured P2P	**Unstructured P2P**
Smerovanie	Na základe smerovacích tabuliek	Flooding, random walk
Vyhľadávanie	Na základe kľúčov	Možnosť komplexnejších požiadaviek
Nájdenie existujúcej položky	vždy	Nieje garantované
Kritická časť	Pripojenie/odpojenie uzla	Vyhľadávanie/smerovanie

Ad-hoc: latinský výraz ktorý znamená “pre tento účel”

Wireless Ad-hoc network:
- sieť bez stanovenej infraštruktúry, sieť tvoria len bezdrôtové zariadenia
- využívajú sa sieťové vlastnosti zúčastnených zariadení
- každé zariadenie figuruje ako host aj ako router
- správa siete je rozdelená medzi uzly
- topológia je dynamická



single-hop Ad-hoc sieť
(všetky zariadenia sú vo vzájomnom dosahu)



multi-hop Ad-hoc sieť

Výhody/Nevýhody Ad-hoc:
- rýchle zostavenie, niesú potrebné žiadne káble
- nemá single point of failure
- zariadenia sa musia zorganizovať samé do topológie
- obmedzený rozsah bezdrôtovej komunikácie
- pohyb mobilných zariadení v priestore a nutnosť rýchlej adaptácie na dynamické zmeny
- medium access control distribuovane realizované
- nájdenie cesty od jedného zariadenia k druhému (proces smerovania)

	Infrastructure-based sieť	**Ad-hoc sieť**
Prerekvizity	Rozmiestnenie routrov, switchov, staníc, serverov	žiadne
Vlastnosti uzlov	Len end(koncové) funkcie	End funkcie aj sieťové funkcie
Spojenia	Drôtové/bezdrôtové	bezdrôtové
Topológia	Daná rozmiestnením zariadení	Dynamicky vytváraná samotnými uzlami
Sieťové funkcie	Poskytnuté infraštruktúrou	Zaistené všetkými uzlami v sieti

Mobile Ad-hoc NETwork(MANET): ad-hoc sieť s pohybujúcimi sa zariadeniami

Wireless Sensor Networks(WSN):
- zariadenia interagujú s prostredím, v ktorom sú umiestnené
- uzly spracovávajú dáta z prostredia a posielajú si ich medzi sebou bezdrôtovo

Medium Access Control v Ad-hoc a WSN:
- všeobecne je veľmi ťažké riadiť prístup k prenosovému médiu v bezdrôtových sieťach
- je veľmi zložité vysielať aj prijímať data súčasne
- z pohľadu odosielateľa je ťažké odhadnúť ruch na strane príjemcu

MAC protokoly v bezdrôtových rádiových sieťach sa rozdelujú:
- 1. Contention-based
- 2. Contention-based with reservation mechanism
- 3. Contention-based with scheduling mechanism

1. Contention-based protocols:
- keď chce uzol vysielať, dohaduje sa o prístupe k médiu so susedmi
- ak viaceré uzly chcú vysielať v tom istom čase nedá sa zabrániť kolízii => tzv content – resolution musí zabezpečovať protokol
- používa 2 prístupy:
    - 1. sender-initiated protocols: prenos začína príjemca
    - 2. receiver-initiated protocols: príjemca spúšťa content – resolution protokol

- pre rezerváciu prenosového média/kanálu pre prenos, zariadenie môže použiť 2 prístupy:
- 1. príjemca informuje potenciálnych narušiteľov počas prebiehajúceho prenosu,

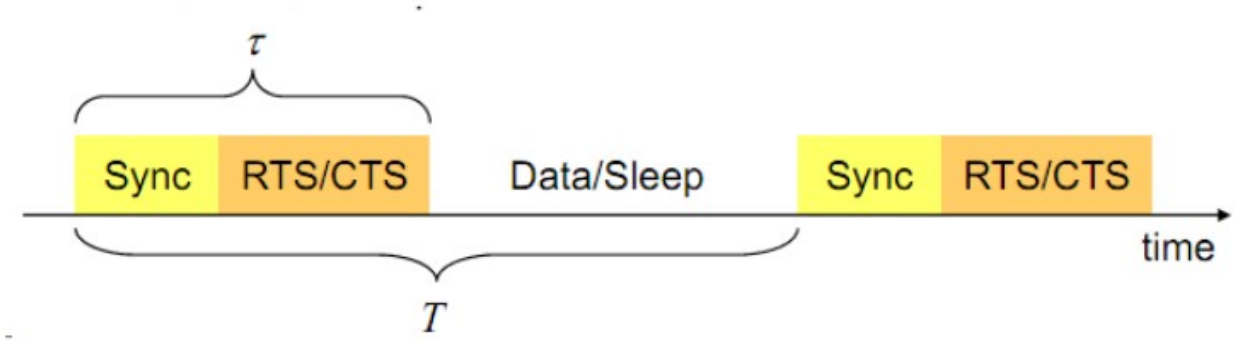
načo ale potrebuje použiť iný špeciálny kanál
2. príjemca informuje potenciálnych narušiteľov pred samotným prenosom

Busy Tone Multiple Access(BTMA):
- prenosový kanál je rozdelený na dátový a kontrolný kanál
- keď chce uzol vysielať, pozre na kontrolný kanál, či je nastavený tzv busy tone, ak nie, nastaví ho sám a začne vysielať

Multiple Access Collision Avoidance(MACA):
- odosielateľ pošle dotaz príjemcovi, či je schopný prijímať, Request to Send(RTS)
- ak príjemca môže prijímať, pošle Clear to Send(CTS)
- RTS/CTS pakety obsahujú informácie o odhadovanej dobe trvania prenosu

Power-Control MAC(PCM):
- redukuje spotrebu energie
- pomocou RTS/CTS správ zisťuje maximálnu prenosovú kapacitu a požadovanú

Sensor-MAC(S-MAC):
- vypína uzly a snaží sa aby boli zapnuté súčasne
- prenos nastáva len keď sú zapnuté
- susedia si musia vymeniť rozvrh bdenia/spánku
- pri bdení si vymieňajú RTS/CTS
- každý uzol pravidelne vysiela svoj rozvrh v SYN pakete, v čase bdenia



Ad-hoc routing:
- smerovací protokol musí spolupracovať s MAC protokolom
- smerovací protokol vyberie cestu od zdroja do cieľa
- MAC protokol povie kedy sa prenos uskutoční

1. Address-based routing:
- každý uzol má priradený unikátny identifikátor v celej sieti
- smerovanie na základe týchto adries/identifikátorov
- podporuje unicast, multicast a broadcast správy

2. Data-centric forwarding:
- smerovanie na základe obsahu správy
- je nutné mať preddefinované typy správ

Rozdelenie smerovacích protokolov pre ad-hoc bezdrôtové siete:

Proactive protocols:
- protokol nájde cesty ešte predtým ako sú potrebné
- uzly si periodicky vymieňajú informácie o topológii

Reactive protocols:
- protokol nájde cesty až keď sú potrebné pre prenos

Table-driven protocols:
- každý uzol vie len next-hop adresu do cieľa

Source-routing protocols:
- každý uzol vie kompletnú cestu do cieľa

Flat protocols:
- každý uzol používa rovnaké algoritmus

Hierarchical protocols:
- niektoré uzly používajú zložitejšie algoritmy, majú väčšiu zodpovednosť

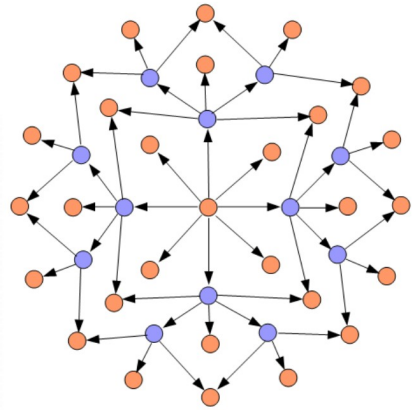
Location-based protocols:
- využívajú fyzické umiestnenie uzlov, ktoré musia mať GPS alebo niečo podobné

Non-location-based protocols:
- nepozerájú na fyzické umiestnenie uzlov

Proactive ad-hoc routing:

Destination Sequence Distance Vector(DSDV):
- distance vector smerovací protokol inšpirovaný protokolom RIP
- smerovacia tabuľka má priestorovú zložitosť O(N)
- starnutie informácií používa pre vyhnutí sa slučkám

Optimized Link State Routing(OLSR):
- link-state smerovací protokol
- šírenie smerovacích informácií optimalizuje pomocou tzv MultiPoint Relays(MPR)
- platné pravidlo: pre každý 2-hop uzol musí existovať MPR cez ktorý ho je schopný kontaktovať

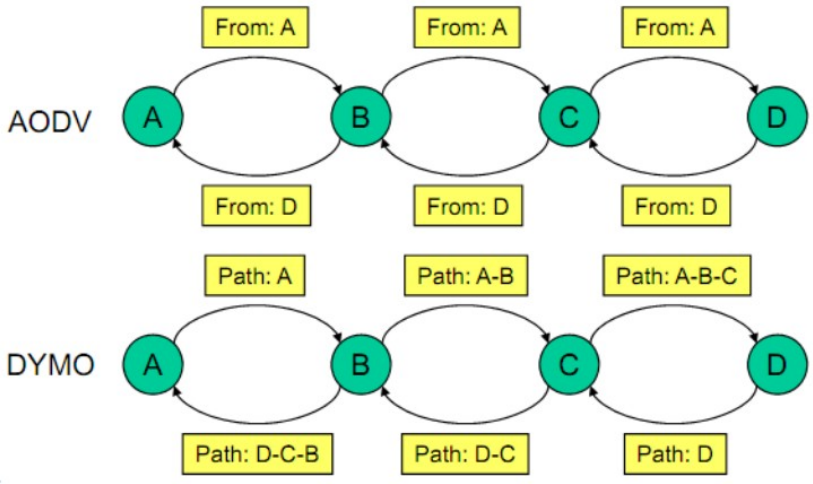


Reactive ad-hoc routing:

Dynamic Source Routing(DSR):
- source-based routing protokol
- odosielateľ posiela Route Request(RREQ) záplavou(flood)
- príjemca posiela odosielateľovi odpoveď Route Reply(RREP)
- niesú potrebné smerovacie tabuľky na každom uzle

Ad-hoc on Demand Distance Vector(AODV):
- používa rovnaký princíp objavovania cesty ako DSR
- uzly si ukladajú informácie odkiaľ paket prišiel v smerovacej tabuľke
- keď je potom sieťou šírená RREQ a uzol na ceste pozná cestu do cieľa, môže odpovedať s RREP namiesto cieľového uzla

Dynamic Manet On Demant(DYMO):
- nasledovník AODV



media = text/obrázky/zvuk/video
multimedia = dáta, ktoré sa skladajú z viacerých typov medií a v istom zmysle sú navzájom spojené, ucelené

Príklady multimediálnych aplikácií:
- e-mail: text, obrázky, zvuk ...
- TV: zvuk, video
- atď...

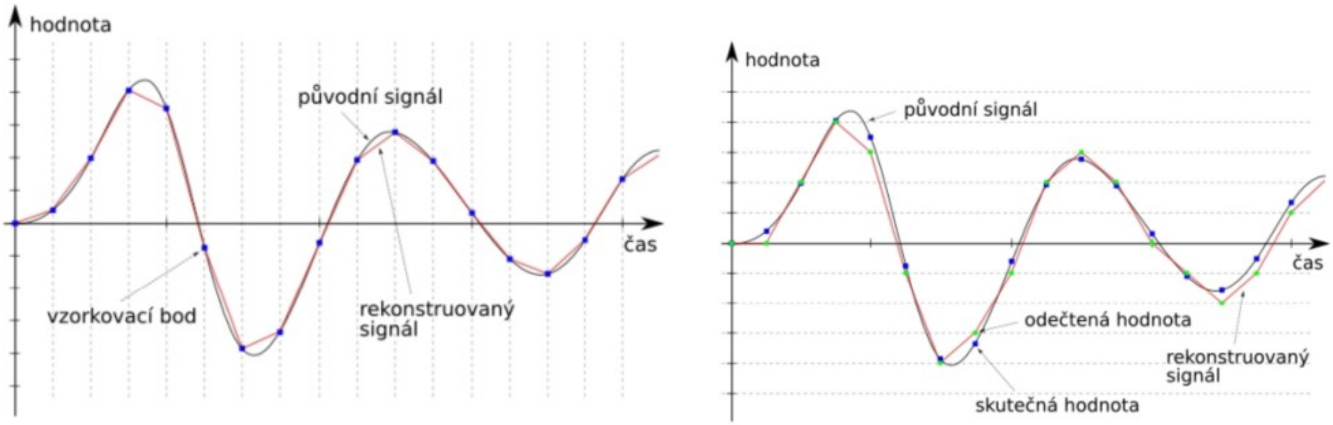
Kompresia:
- zmenšovanie objemu dát

Kompresné algoritmy:
1. Stratové: vo výsledných dátach sa vyskytujú chyby oproti originálnym dátam
2. Bezstratové: vo výsledných dátach sa nevyskytujú žiadne chyby oproti originálnym

Prevod analógového audio signálu na digitálny pomocou procesou sampling(vzorkovanie) a quantization(dávkovanie).

Sampling = redukcia spojitého signálu na diskrétny, vzorky(hodnoty) pôvodného signálu sú odoberané v stanovenom časovom intervale.

Quantization = aproximovanie hodnoty pôvodného signálu diskrétnymi hodnotami alebo celočíselnými(integer) hodnotami, (niečo ako zaokrúhlovanie na celé čísla)



Audio:
- je schopné zniesť 1 – 2% straty dát bez výrazného porušenia
- Real-Time Intolerant(RTI) aplikácie: sú aplikácie, ktoré netolerujú vysokú odozvu v prenose audio dát, napr: Internet conferencing/telephony, max. 100 – 200 msec
- Real-Time Tolerant(RTT) aplikácie: sú aplikácie, ktorým oneskorenie prenosu audio dát až tak nevadí

Graphics and Animation:
- dáta pozostávajú zo statických medií ako obrázky a dynamických medií ako flash animácie
- pôvodná forma obrázku pred kompresiou je vo forme poľa pixelov, kde každý pixel je reprezentovaný postupnosťou bitov, udávajúcich jeho jas a farbu
- pre kompresiu obrázkov sa často používajú progresívne kompresie

Progresívna kompresia:
- obrázky sú rozdelené akoby do vrstiev
- keď príjemca získa prvú vrstvu obrázku, môže zobraziť obrázok v slabom rozlíšení
- každé ďalšie prijaté dáta obrázku predstavujú dodatočné informácie a vrstvy, ktoré umožňujú zobrazenie obrázku vo vyššom rozlíšení

Video:
- sekvencia obrázkov zobrazovaných s určitou frekvenciou (napr: 30 FPS(Frames Per Second))
- požiadavky na priepustnosť (bandwidth) väčšinou vysoké

- HD: 1,5 Gbps, 2K: 3Gbps, 4K: 6Gbps
- kompresia zvyšuje latency, takže nevhodné pre real-time aplikácie

Všetky typy medii sa rozdeľujú na:
1. **Real-Time(RT)**: môžu mať tvrdé (hard) alebo volnejšie (soft) požiadavky na delay
2. **Non Real-Time(NRT)**: nemajú žiadne požiadavky na oneskorenie (delay)

Real-Time(RT) media:
1. Discrete Media(DM): dáta sú posielané naraz, napr: v jednom súbore
2. Continuous Media(CM), dáta sú posielané ako prúd navzájom závislých správ

RT Continuous Media:
1. Delay tolerant: tolerujú väčšie meškania bez degradácie úrovne
2. Delay intolerant: netoleruju väčšie meškania

Delay = čas potrebný na cestu paketu od zdroja k cieľu
Jitter = rozdielnosť v delay na strane príjemcu

End-to-end delay je ovplyvnené:
1. Packet Processing Delay
2. Packet Transmission Delay
3. Propagation Delay
4. Routing and Queuing Delay

1. Packet Processing Delay:
- oneskorenie na strane odosielateľa ako aj na strane príjemcu
- spôsobené enkapsuláciou a dekapsuláciou dát na každej vrstve

2. Packet Transmission Delay:
- čas, ktorý potrebuje fyzická vrstva odosielateľa na odoslanie dát cez médium
- ovplyvňuje ho naplnenie fronty fyzickej vrstvy
- ovplyvňuje ho MAC delay, čas na prístup k prenosovému médiu

3. Propagation Delay:
- čas putovania dát cez fyzické prenosové médium
- limitované prinajmenšom rýchlosťou svetla
- 200m propagation delay = 1ms, 20000km propagation delay = 100ms

4. Routing and Queuing Delay:
- best-effort internet zaobchádza s každým paketom rovnako
- keď paket príde do fronty routru, musí čakať kým môže byť obslúžený
- toto meškanie je náhodné a najviac prispieva k jitteru

Bandwidth:
- best-effort internet nepodporuje žiadne rezervácie sieťových zdrojov
- keď aplikácia vyžaduje istú prenosovú rýchlosť (bandwidth), musí si ho zabezpečiť sama

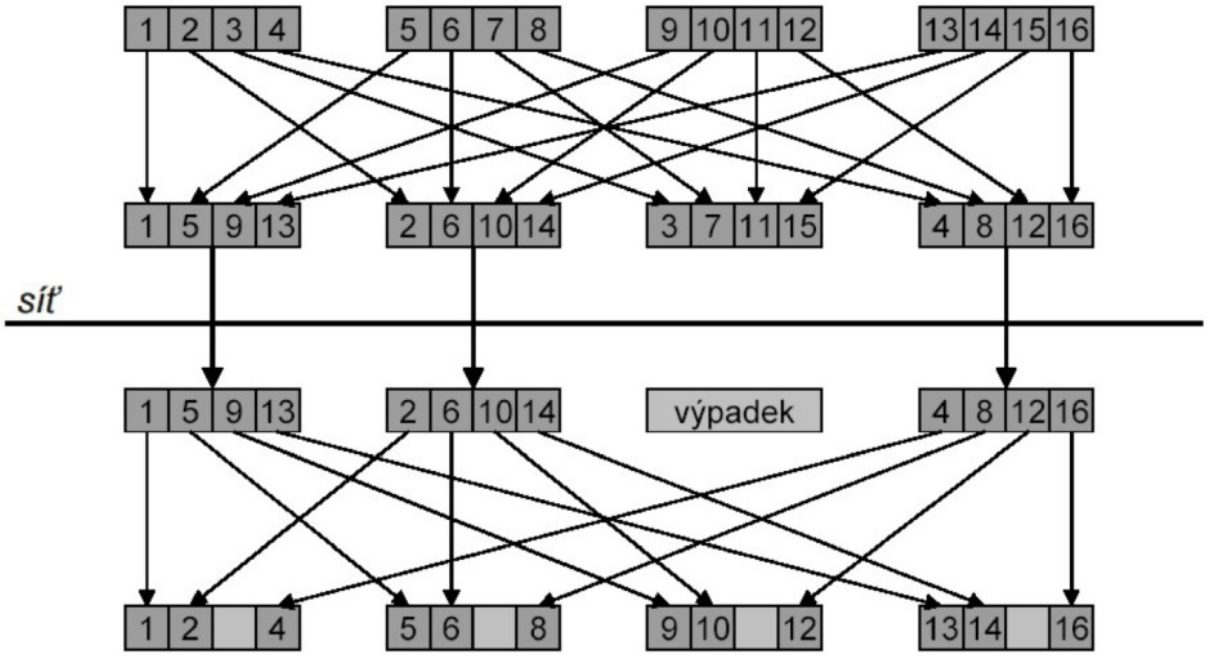
- aplikácie využívajúce TCP môžu využiť jeho vstavanú funkciu congestion control
- multimedia aplikácie ale väčšinou využívajú UDP protokol, ktorý takú funkciu nemá

Rýchlosť packet processingu závisí viac od počtu paketov ako od ich veľkosti.

Error Correction:
1. Sender-based Repair: active repair: ARQ, passive repair: FEC, Interleaving
2. Receiver-based Repair: Error Concealment

Forward Error Correction (FEC):
- požaduje dodatočné informácie k prúdu paketov, tzv repair data
- tieto informácie sú použité na rekonštrukciu stratených paketov

Interleaving:
- použiteľný len v prípade, že dátová jednotka daného media je menšia ako veľkosť paketu
- je založený na preusporiadaní dátových jednotiek media pred prenosom
- strata jedného paketu spôsobí viacero menších strát rôznych dátových jednotiek
- u príjemcu sú dátové jednotky usporiadané do pôvodnej postupnosti



Error Concealment:
- vygenerovanie strateného paketu na základe úspešne prijatých paketov
1. **Insertion-based**: paket, ktorý sa vyplní tichom, hlukom alebo opakovaním obsahu jeho susedných paketov
2. **Interpolation-based**: odvodenie strateného paketu interpoláciou
3. **Regeneration-based**: odvodí stav dekódera zo susedných paketov a z toho vygeneruje stratený paket