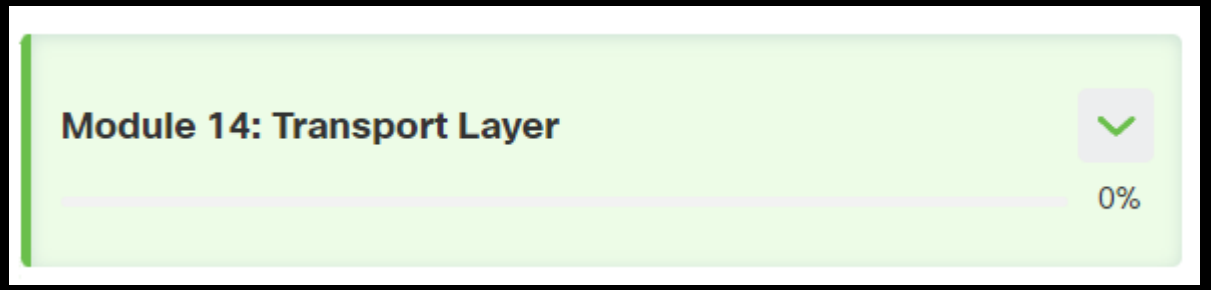
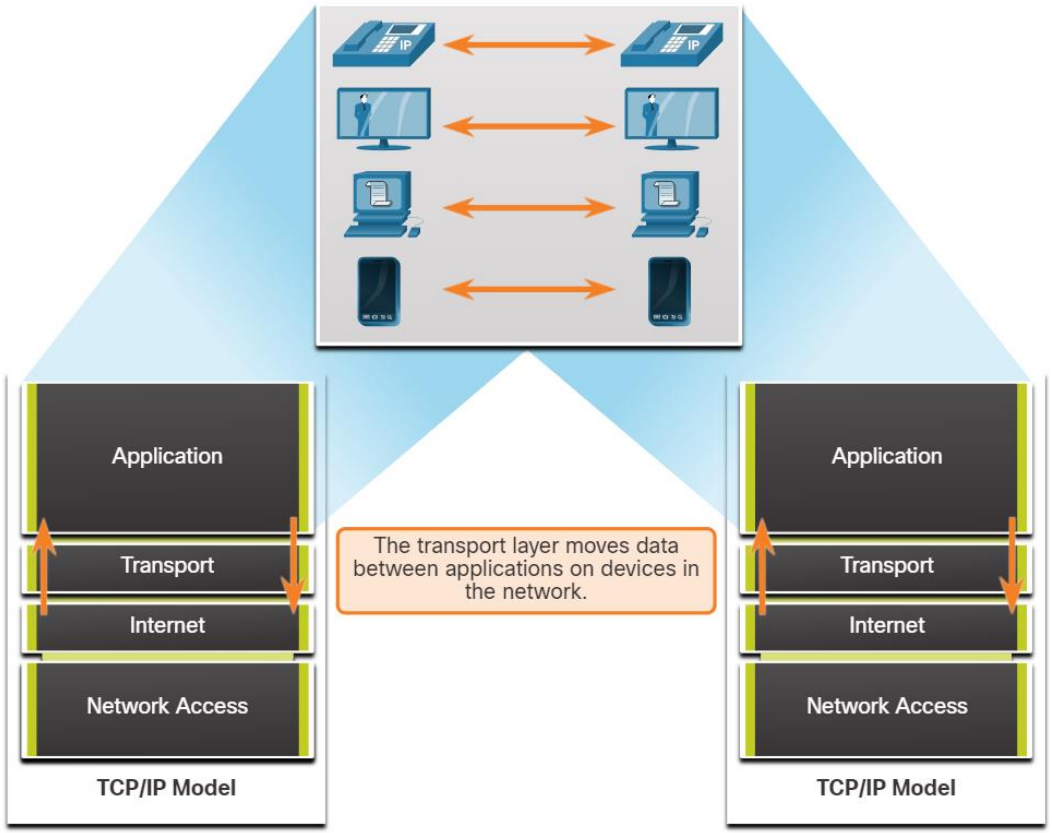
Transportni sloj

- Transportni sloj
- TCP
- UDP
- TCP/UDP port numbers

NetAcademy chapter 14.



Uloga transpornog sloja



- Programi aplikacijskog sloja generiraju podatke koji se m
  razmjeniti između izvorišnog i odredišnog računala.
- Transportni sloj odgovoran je za logičku komunikaciju izr
  aplikacija koje se izvode na različitim računalima. To mož
  uključivati usluge kao što je uspostavljanje privremene v
  između dva računala i pouzdan prijenos podataka za apli
- Transportni sloj nema saznanja (ne mari) o tipu odredišr
  računala, vrsti medija preko kojeg podaci moraju putova
  putanji kojom prolaze podaci, zagušenju veze ili veličini r

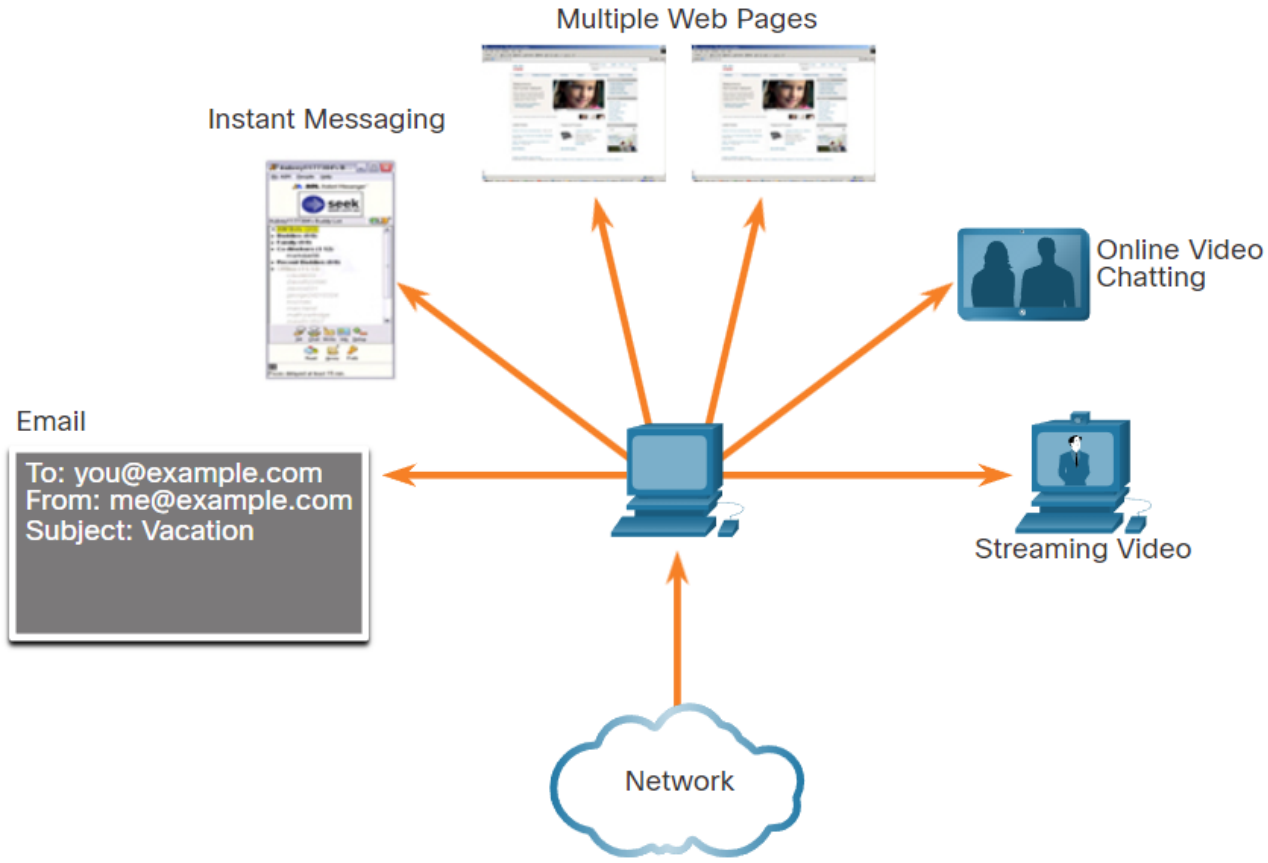
Transportni sloj uključuje dva protokola:

- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

Uloga transpornog sloja

- Na transportnom sloju, svaki skup podataka koji se razmjenjuje između izvorne aplikacije i odredišne aplikacije poznat je kao *razgovor* i zasebno se prati. Odgovornost je transportnog sloja da održava i prati te višestruke *razgovore*.

- Većina mreža ima ograničenje količine podataka koji se mogu uključiti u jedan paket. Stoga se podaci moraju podijeliti na dijelove kojima se može upravljati.

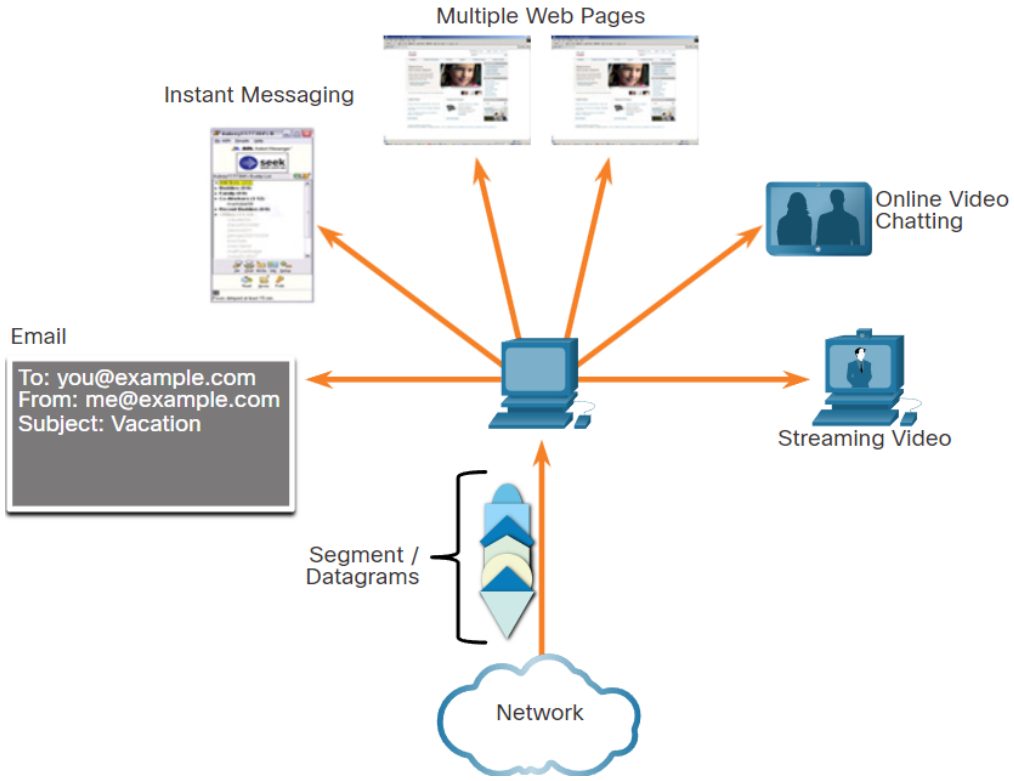


Uloga transpornog sloja

Exam question

- Segmentiranje podataka i ponovno sastavljanje segmenata
- Transportni sloj odgovoran je za dijeljenje podataka aplikacije u blokove odgovarajuće veličine. Ovisno o korištenom protokolu transportnog sloja, blokovi transportnog sloja (PDU) nazivaju se ili segmenti ili datagrami.
- Slika ilustrira transportni sloj koji koristi različite PDU za svaki razgovor.

- Transportni sloj dijeli podatke u manje blokove (tj. segmente ili datagrame) kojima je lakše upravljati i prenositi ih.



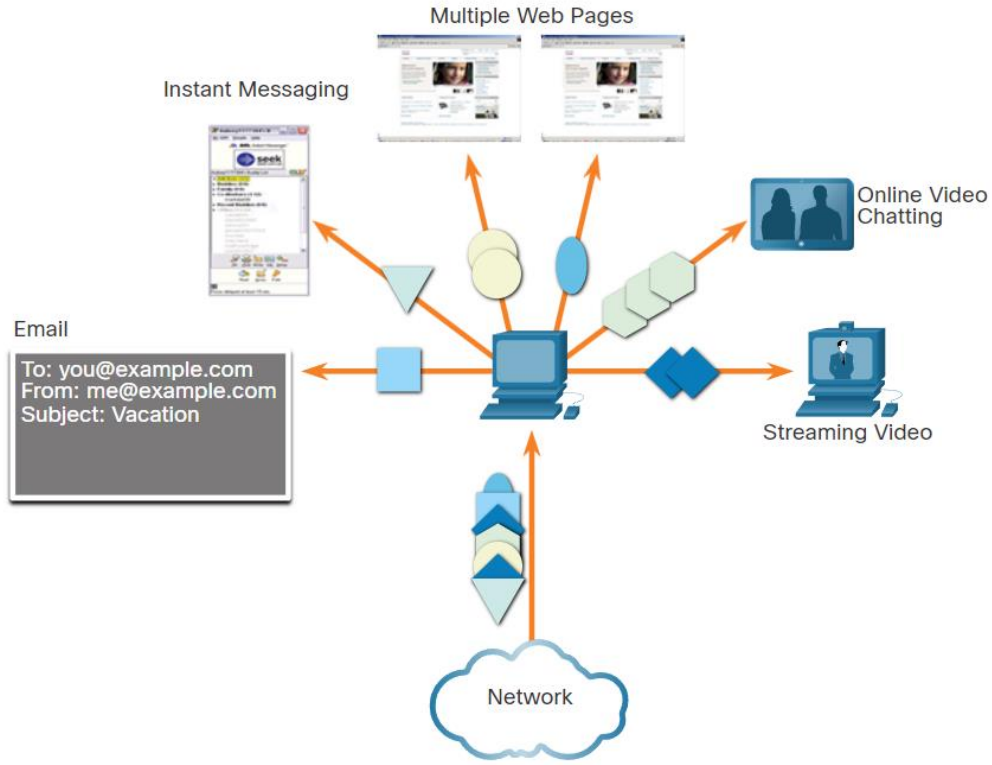
Uloga transpornog sloja

Exam question

- Dodavanje opisnih podataka u zaglavlje
- Protokol transportnog sloja također dodaje podatke zaglavlja koje sadrže binarne nizove organizirane u polja svakog bloka podataka. Vrijednosti u tim poljima omogućuju različitim protokolima transportnog sloja obavljanje različitih funkcija u upravljanju komunikacijom podataka.

- Na primjer, host primatelj koristi podatke zaglavlja za ponovno sastavljanje blokova podataka u potpuni tok podataka za aplikaciju primatelja.

- Transportni sloj osigurava da čak i kada se na uređaju izvodi više aplikacija, sve aplikacije primaju točne podatke.

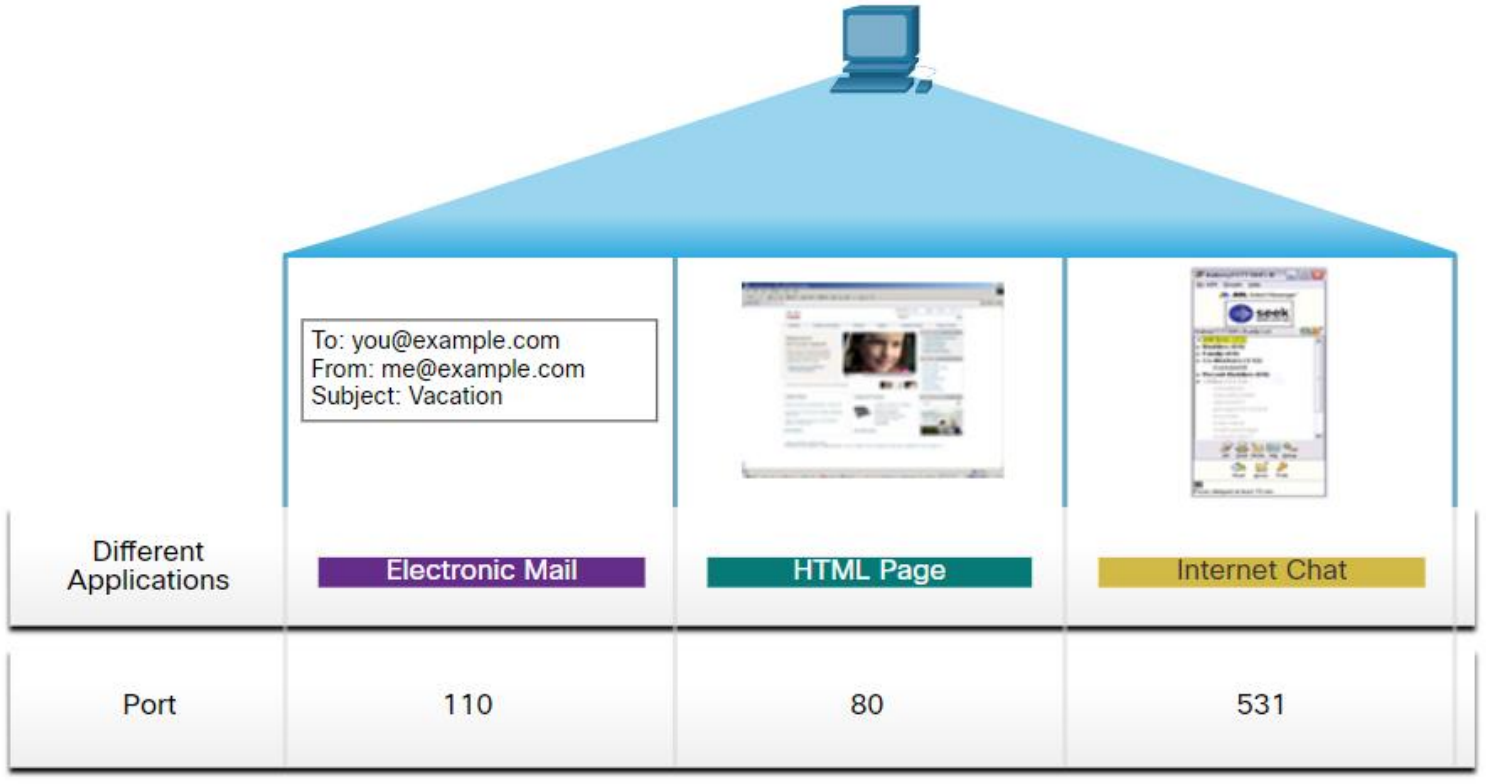


Uloga transpornog sloja

Exam question

- Identificiranje aplikacija

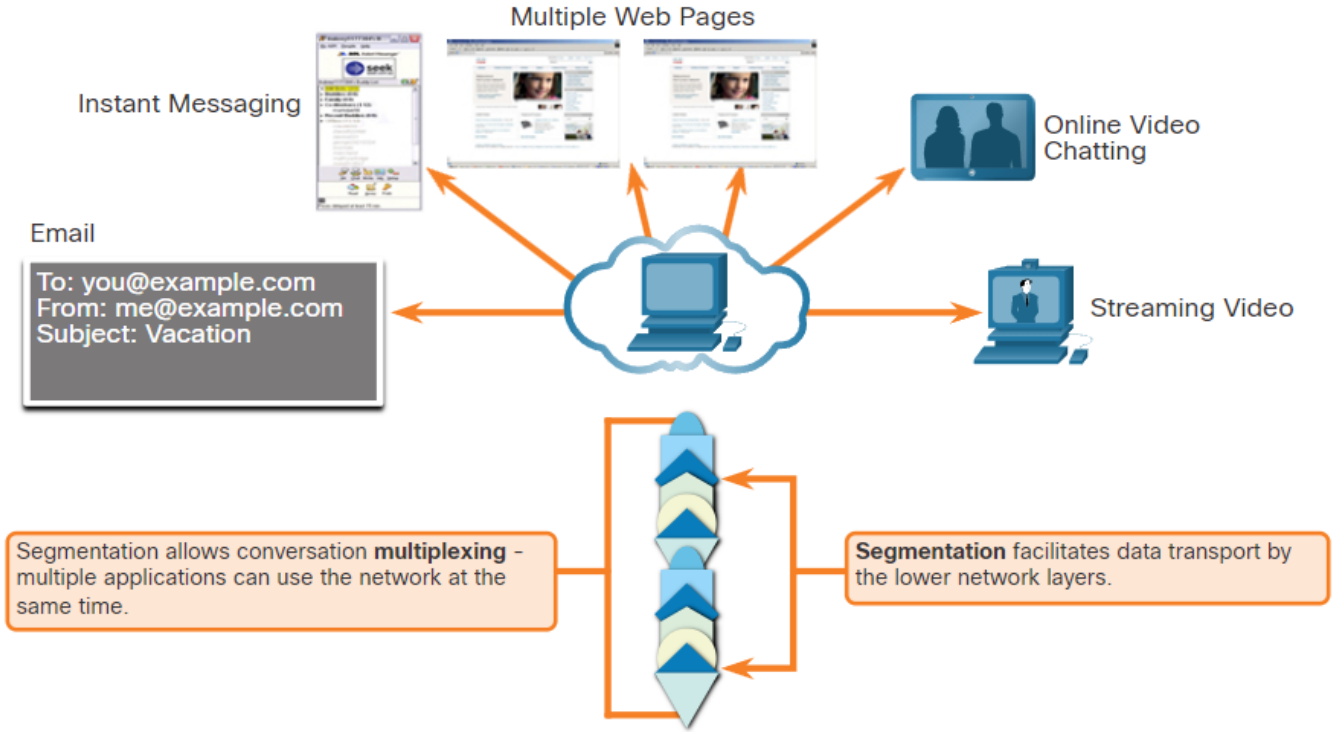
- Transportni sloj mora biti u stanju odvojiti i upravljati višestrukim komunikacijama s različitim zahtjevima prijenosa. Za prosljeđivanje tokova podataka odgovarajućim aplikacijama, transportni sloj identificira ciljnu aplikaciju pomoću identifikatora koji se naziva broj priključka (port number).
- Kao što je prikazano na slici, svakom softverskom procesu koji treba pristupiti mreži dodijeljen je broj porta koji je jedinstven za to glavno računalo.



Uloga transpornog sloja

Exam question

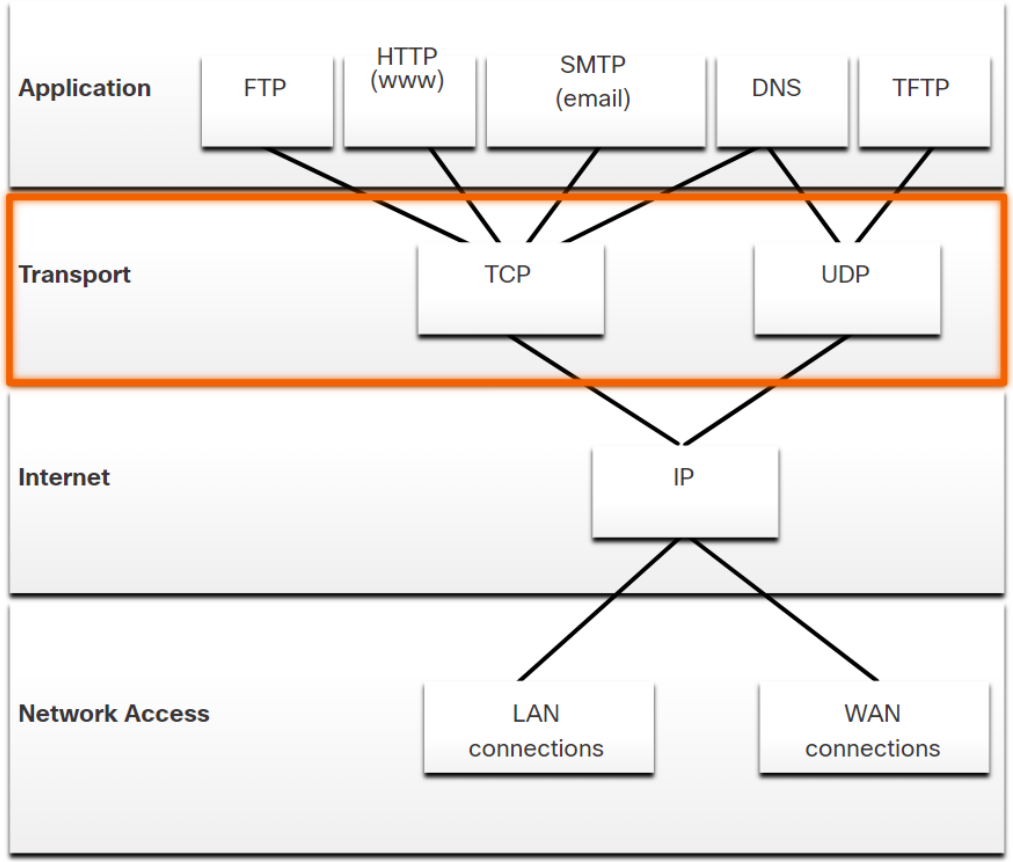
- Multipleksiranje razgovora
- Slanje nekih vrsta podataka (npr. strujanje videa-video stream) preko mreže, kao jedan potpuni komunikacijski tok, može zauzeti svu dostupnu mrežnu propusnost. To bi spriječilo druge komunikacijske razgovore da se odvijaju u isto vrijeme. To bi također otežalo oporavak od pogreške i ponovni prijenos oštećenih podataka.
- Transportni sloj koristi segmentaciju i multipleksiranje kako bi omogućio *„istovremeno"* odvijanje različitih komunikacijskih razgovora na istoj mreži.
- Provjera pogrešaka može se izvršiti na podacima u segmentu kako bi se utvrdilo je li segment promijenjen tijekom prijenosa.



Uloga transpornog sloja

Exam question

- IP se bavi samo strukturom, adresiranjem i usmjeravanjem paketa. IP ne specificira kako se odvija isporuka ili transport paketa.
- Protokoli transportnog sloja određuju kako prenijeti poruke između hostova i odgovorni su za upravljanje zahtjevima pouzdanosti razgovora. Transportni sloj uključuje TCP i UDP protokole.
- Različite primjene imaju različite zahtjeve u pogledu pouzdanosti transporta.
- Stoga TCP/IP pruža dva protokola transportnog sloja, kao što je prikazano na slici.



?

Transmission Control Protocol (TCP)

Exam question

- IP nije odgovoran za jamčenje isporuke ili određivanje treba li se uspostaviti veza između pošiljatelja i primatelja.
- TCP se smatra pouzdanim protokolom transportnog sloja sa svim mogućim funkcionalnostima potrebnima da podatci pouzdano stignu na odredište. TCP za to koristi polja koja omougćavaju pouzdanu isporuku podataka aplikacije. Ova polja zahtijevaju dodatnu obradu od strane računala pošiljatelja i primatelja, pa se to može smatrati dodatnim opterećenjem za računalo
- TCP dijeli podatke u segmente.
  - TCP prijenos analogan je slanju paketa koji se prate od izvora do odredišta. Ako je narudžba za dostavu podijeljena u nekoliko paketa, kupac može provjeriti redoslijed isporuke na internetu.

Transmission Control Protocol (TCP)

Exam question

TCP pruža pouzdanost i kontrolu protoka pomoću ovih osnovnih operacija:

- Označava i prati segmente podataka koji se prenose određenom hostu iz određene aplikacije
- Potvrđuje primljene podatke
- Ponovno šalje sve nepotvrđene podatke nakon određenog vremena
- Brine o redoslijedu podataka prilikom dostave (na odredištu)
- Šalje podatke „učinkovitom brzinom” koja je prihvatljiva primatelju
- Kako bi održao stanje razgovora i pratio informacije, TCP prvo mora uspostaviti vezu između pošiljatelja i primatelja. Zbog toga je TCP poznat kao protokol usmjeren na povezivanje (connection oriented).

User Datagram Protocol (UDP)

Exam question

- UDP je jednostavniji protokol transportnog sloja od TCP-a. Ne pruža pouzdanost i kontrolu protoka, što znači da zahtijeva manje polja zaglavlja. Budući da UDP procesi pošiljatelja i primatelja ne moraju upravljati pouzdanošću i kontrolom protoka, to znači da se UDP datagrami mogu obraditi brže od TCP segmenata. UDP pruža osnovne funkcije za isporuku datagrama između odgovarajućih aplikacija, s vrlo malo „troška" i provjere podataka.
- UDP dijeli podatke u datagrame.
- UDP je protokol bez uspostave veze (connectionless). Budući da UDP ne pruža pouzdanost ili kontrolu protoka, ne zahtijeva uspostavljenu vezu.
- UDP je također poznat kao protokol za best-effort delivery jer ne postoji potvrda da su podaci primljeni na odredištu. S UDP-om nema procesa na transportnom sloju koji obavještava pošiljatelja o uspješnoj isporuci.

- UDP je poput slanja običnog, neregistriranog pisma poštom. Pošiljatelj pisma nije svjestan dostupnosti primatelja da primi pismo. Niti poštanski ured nije odgovoran za praćenje pisma ili obavještavanje pošiljatelja ako pismo ne stigne na konačno odredište.

TCP karakteristike

- Osim podrške osnovnim funkcijama segmentacije i ponovnog sastavljanja podataka, TCP također pruža sljedeće usluge:

- Uspostavlja veze - TCP je protokol orijentiran na vezu koji pregovara i uspostavlja vezu (ili sesiju) između izvorišnog i odredišnog uređaja prije prosljeđivanja bilo kakvog prometa. Kroz uspostavu veze, uređaji pregovaraju o količini prometa koji se može proslijediti u određenom trenutku i prate što se događa s podatcima koji se šalju između uređaja kako bi upravljao tim tokom podataka.

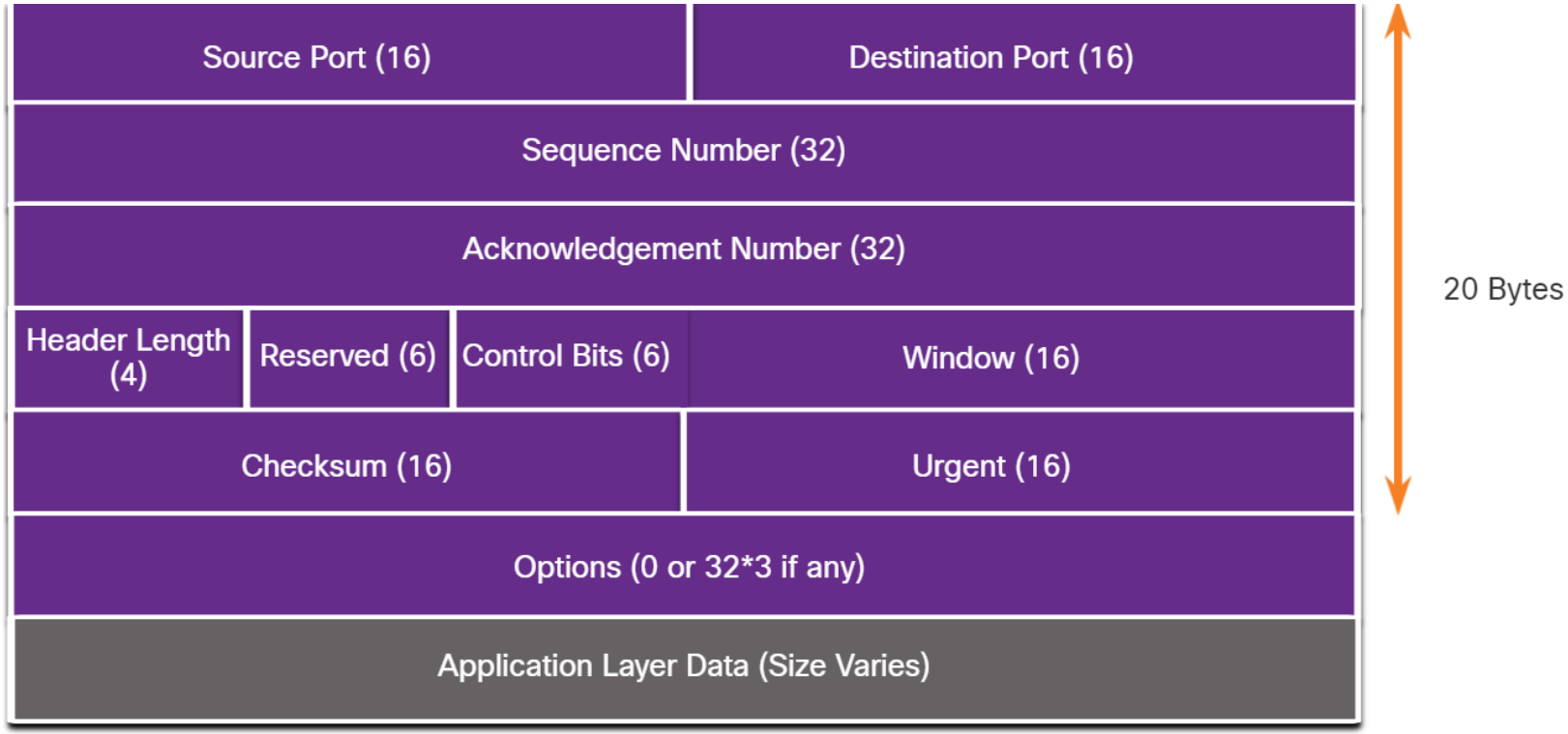
- Osigurava pouzdanu isporuku - Iz mnogo razloga, moguće je da se segment ošteti ili potpuno izgubi, dok se prenosi preko mreže. TCP osigurava da svaki segment koji šalje izvor stigne na odredište.

- Omogućuje isporuku istim redoslijedom - Budući da podatci (segmenti) mogu doći do odredišta kroz različite dijelove mreže, podaci mogu stići pogrešnim redoslijedom. Numeriranjem i preslagivanjem segmenata, TCP osigurava ponovno sastavljanje segmenata u pravilan redoslijed.

- Podržava kontrolu protoka - Mrežni hostovi imaju ograničene resurse (tj. memoriju i procesorsku snagu). Kada je TCP „svjestan" da su ti resursi preopterećeni, može zatražiti da aplikacija koja šalje smanji brzinu protoka podataka. To znači da TCP regulira količinu podataka koju izvor šalje. **Kontrola protoka može spriječiti potrebu za ponovnim slanjem podataka** kada su resursi glavnog računala primatelja preopterećeni.

TCP zaglavlje - Header

- TCP je „stateful“ protokol što znači da prati stanje komunikacijske veze. Kako bi pratio stanje veze, TCP bilježi koje je informacije poslao i koje je informacije potvrdio. Stateful veza počinje uspostavljanjem veze i završava prekidom veze.
-
- TCP segment dodaje 20 bajtova (tj. 160 bitova) opterećenja (opisnih podataka) kada enkapsulira podatke sloja aplikacije.

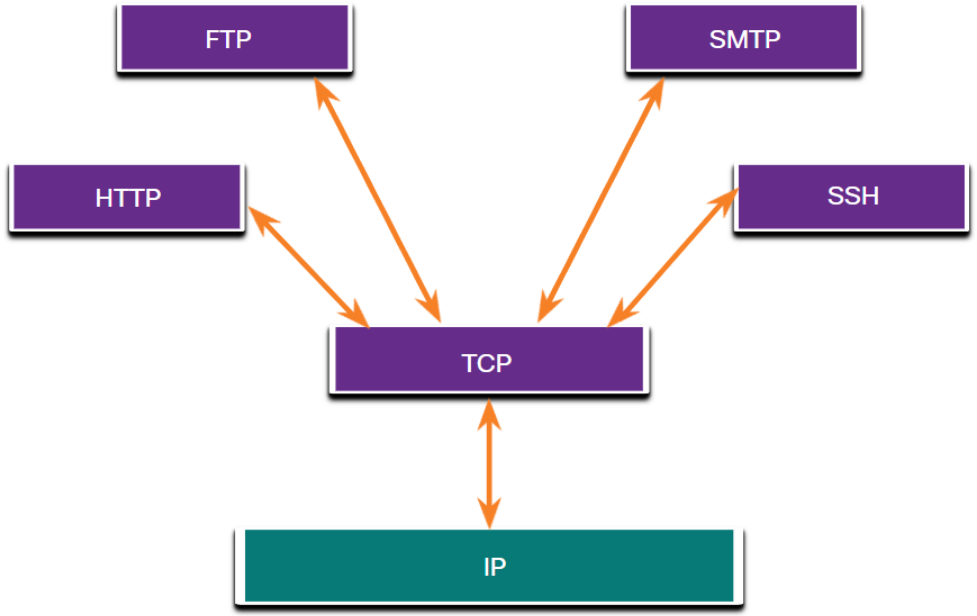


TCP zaglavlje - Header

TCP Header Field	Description
Source Port	16-bitno polje koje se koristi za identifikaciju izvorne aplikacije prema broju priključka (porta).
Destination Port	16-bitno polje koje se koristi za identifikaciju odredišne aplikacije prema broju priključka (porta).
Sequence Number	32-bitno polje koje se koristi za potrebe ponovnog sastavljanja podataka.
Acknowledgment Number	32-bitno polje koje se koristi za označavanje da su podaci primljeni i koji je sljedeći bajt (tzv. Expectational acknowledgement) koji se očekuje od izvora.
Header Length	4-bitno polje poznato kao "pomak podataka„ (data offset) koje označava duljinu zaglavlja TCP segmenta.
Reserved	6-bitno polje koje je rezervirano za buduću upotrebu.
Control bits	6-bitno polje koje uključuje bitne(važne) kodove ili zastavice koje označavaju svrhu i funkciju TCP segmenta.
Window size	16-bitno polje koje se koristi za označavanje broja bajtova koji se mogu prihvatiti odjednom.
Checksum	16-bitno polje koje se koristi za provjeru pogrešaka zaglavlja segmenta i podataka.
Urgent	16-bitno polje koje se koristi za označavanje jesu li sadržani podaci hitni.

Aplikacije koje koriste TCP

- TCP je dobar primjer kako različiti slojevi TCP/IP stoga protokola (protocol stack) imaju specifične uloge. TCP odrađuje sve zadatke povezane s dijeljenjem toka podataka u segmente, pružajući pouzdanost, kontrolirajući protok podataka i mijenjajući redoslijed segmenata.
- TCP oslobađa aplikaciju potrebe za upravljanjem bilo kojim od ovih zadataka. Aplikacije, poput onih prikazanih na slici, mogu jednostavno poslati tok podataka transportnom sloju i koristiti usluge TCP-a.

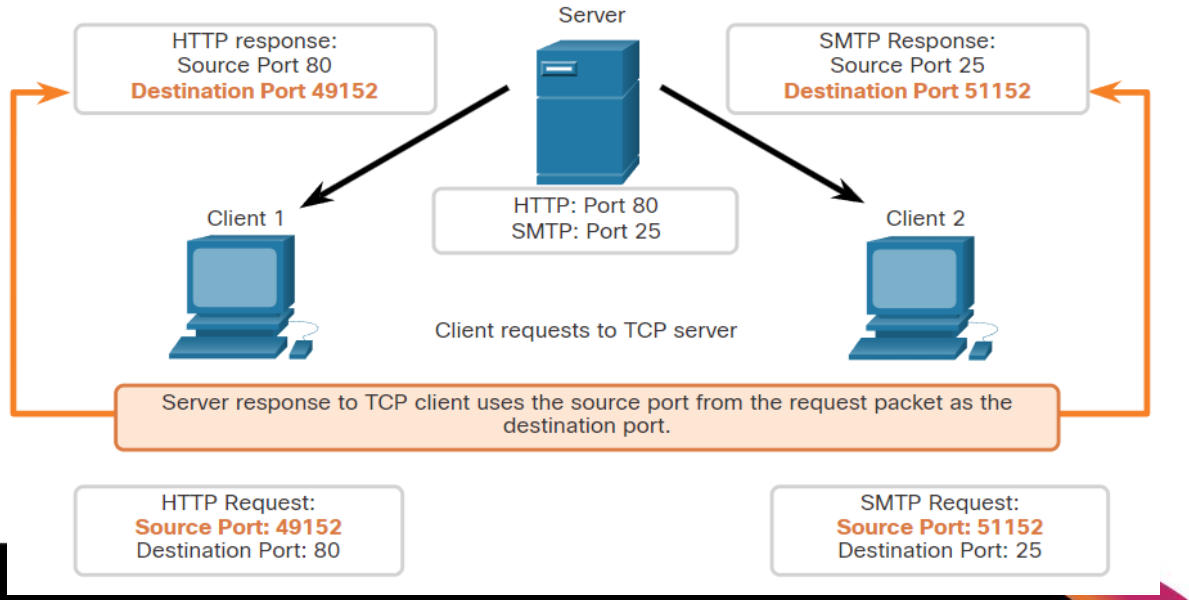


TCP komunikacijski proces

Svaki aplikacijski proces (aplikacija/servis) koji se izvodi na poslužitelju konfiguriran je za korištenje broja priključka. Broj priključka se ili automatski dodjeljuje ili ručno konfigurira administrator sustava.

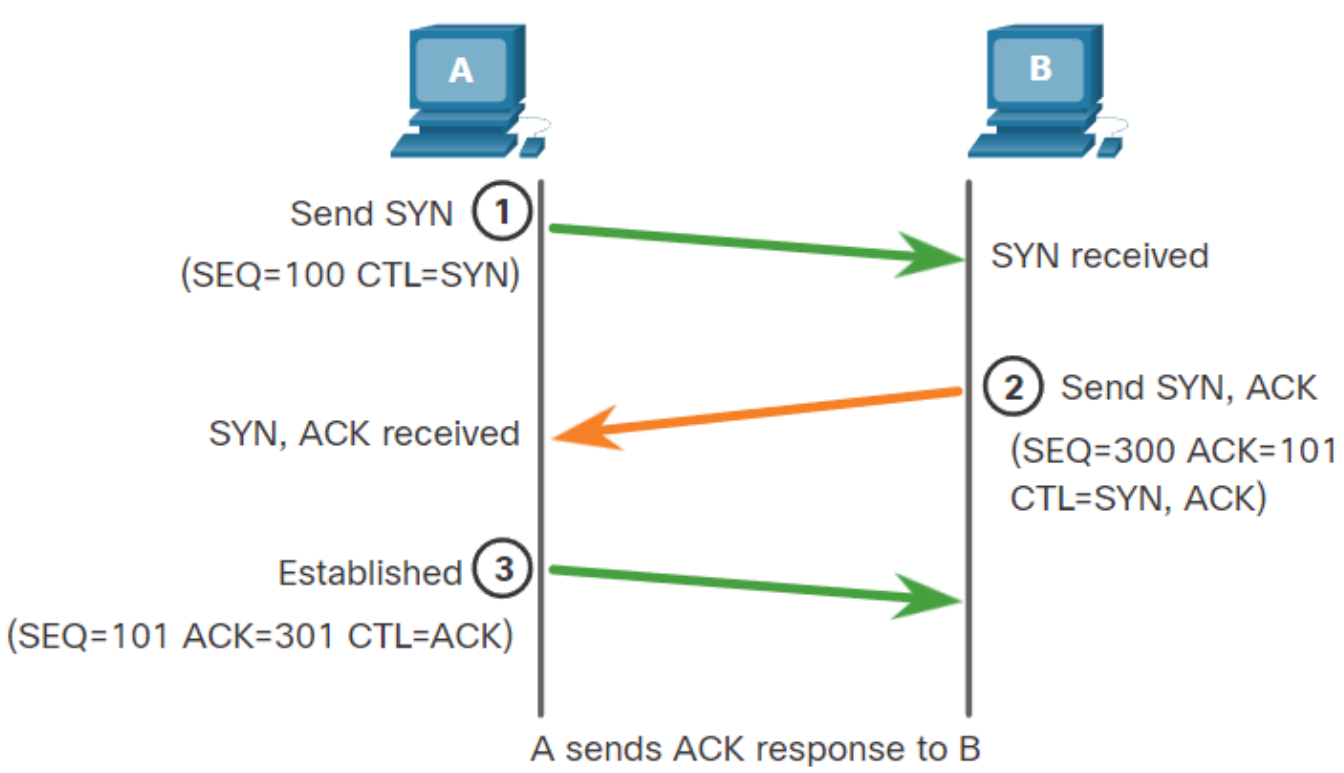
Pojedinačni poslužitelj ne može imati dvije usluge dodijeljene istom broju priključka unutar istih usluga transportnog sloja. Na primjer, host koji pokreće aplikaciju web poslužitelja i aplikaciju za prijenos datoteka ne mogu oboje imati konfiguriran isti port npr. TCP port 80.

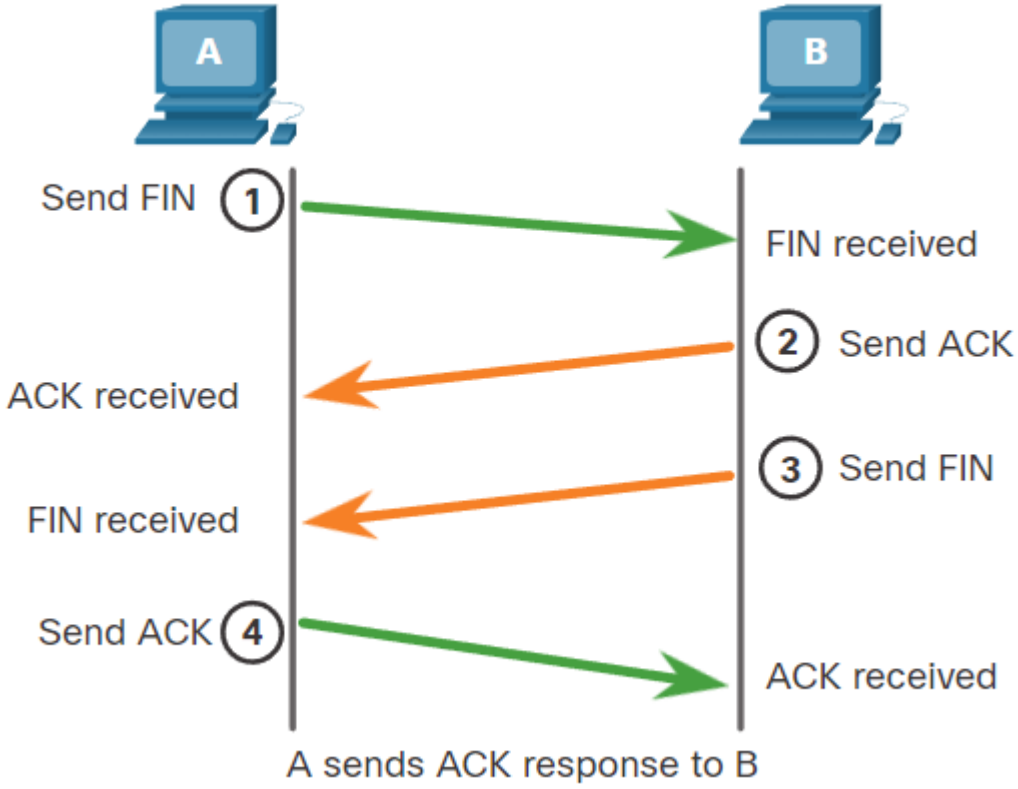
Aktivna poslužiteljska aplikacija povezana s određenim portom smatra se otvorenom, što znači da transportni sloj prihvaća i obrađuje segmente upućene tom portu. Svaki dolazni zahtjev klijenta upućen ispravnom socketu se prihvaća, a podaci se prosljeđuju aplikaciji poslužitelja. Na poslužitelju može biti istovremeno otvoreno mnogo portova, jedan za svaku aktivnu poslužiteljsku aplikaciju.



Uspostava i zatvaranje TCP veze

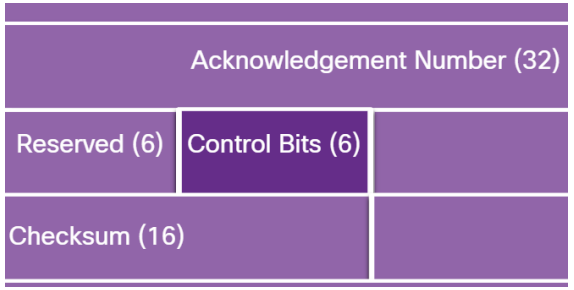
three-way handshake process





Uspostava i zatvaranje TCP veze

Računala održavaju stanje veze, prate svaki segment unutar sesije i razmjenjuju informacije o tome koji su podaci primljeni pomoću informacija u TCP zaglavlju. TCP je full-duplex protokol, gdje svaka veza predstavlja dvije jednosmjerne komunikacijske sesije. Da bi uspostavili vezu, domaćini izvode trostruko rukovanje (3-way handshake). Kontrolni bitovi u TCP zaglavlju pokazuju status veze.

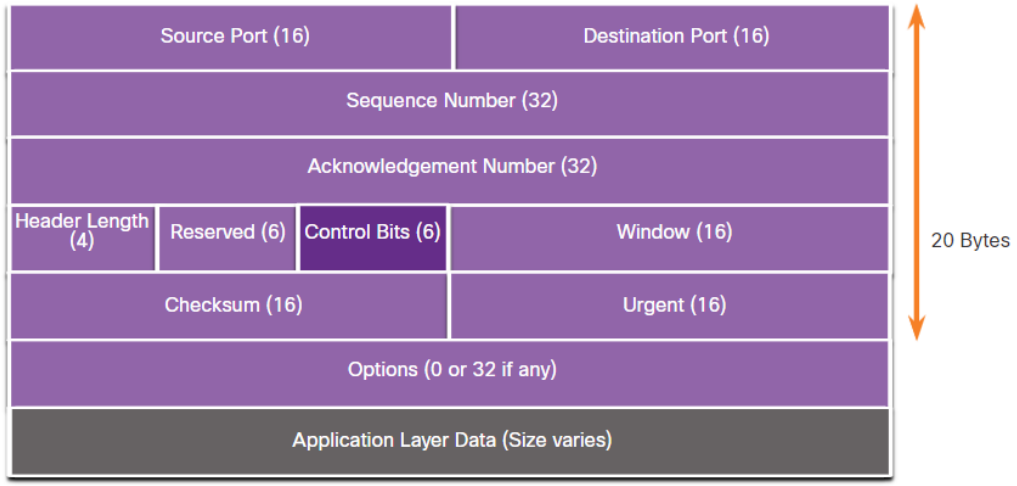


Uloga trostrukog rukovanja (3-way handshake) :

1. Utvrđuje da je odredišni uređaj prisutan na mreži.
2. Provjerava ima li odredišni uređaj aktivnu uslugu i prihvaća li zahtjeve na broju odredišnog priključka koji inicirajući klijent namjerava koristiti.
3. Obavještava odredišni uređaj da izvorni klijent namjerava uspostaviti komunikacijsku sesiju na tom broju priključka.

Nakon završetka komunikacije između aplikacija na računalima veza na transportnom sloju se također prekida.

Uspostava i zatvaranje TCP veze



URG - Značajno polje hitnog pokazivača, označava da postoje hitni podaci u segmentu koji zahtijevaju hitnu pozornost od primajuće aplikacije..npr. telnet veza kada pošaljemo kombinaciju tipki za prekid

ACK - Oznaka potvrde koja se koristi za uspostavljanje veze i prekid sesije

PSH - Push funkcija-signalizira primatelju da se podaci u trenutnom segmentu trebaju odmah obraditi, umjesto da se spremaju u međuspremnik dok ne stigne više podataka. Kada je PSH bit postavljen na 1, on ukazuje prijemnom TCP stogu da bi trebao "gurnuti" ove podatke do aplikacije primatelja bez čekanja da dodatni podaci popune međuspremnik. Primjer aplikacije koja koristi oznaku PSH (Push) je telnet (ali i Secure Shell-SSH), koja omogućuje daljinski pristup naredbenom retku uređaja. Telnet korisnici očekuju da se njihove naredbe i odgovori odmah pojave zato se koristi PSH zastavica.

RST - Resetirajte vezu kada dođe do pogreške ili isteka vremena

SYN - Sinkronizirajte redne brojeve koji se koriste za uspostavljanje veze

FIN - Nema više podataka od pošiljatelja i koristi se za prekid sesije

Uspostava i zatvaranje TCP veze

three-way handshake process

192.168.43.223	104.19.141.57	TCP	66 49441→80	[SYN] Seq=0 Win=8192 Len
104.19.141.57	192.168.43.223	TCP	66 80→49441	[SYN, ACK] Seq=0 Ack=1 W
192.168.43.223	104.19.141.57	TCP	54 49441→80	[ACK] Seq=1 Ack=1 Win=66

Session termination

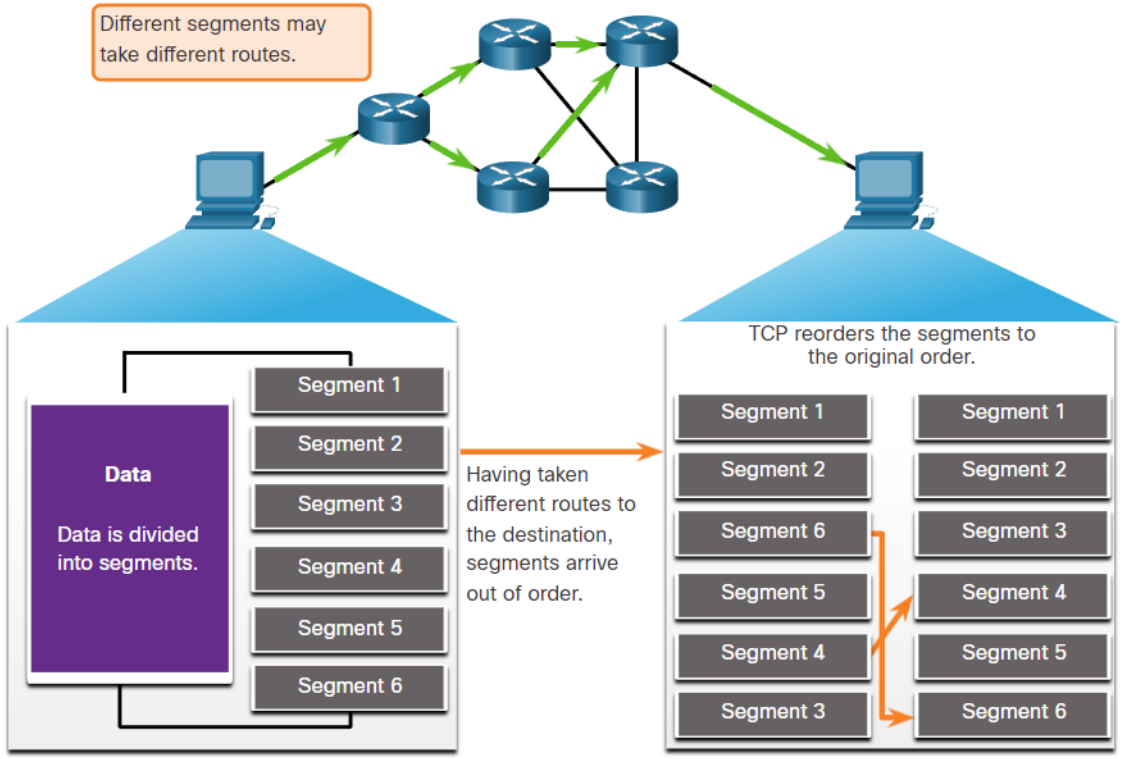
This ACK is for some of the previous segments

192.168.43.223	104.19.141.57	TCP	54 49607→80	[FIN, ACK] Se
104.19.141.57	192.168.43.223	TCP	54 80→49607	[FIN, ACK] Se
192.168.43.223	104.19.141.57	TCP	54 49607→80	[ACK] Seq=2 A

TCP pouzdanost - zajamčena isporuka „po redu”

- Tijekom postavljanja sesije nasumično se odabire početni redni broj (ISN-Initial sequence number). Ovaj ISN predstavlja početnu vrijednost bajtova koji se prenose prijemnoj aplikaciji.
- Kako se podaci prenose tijekom sesije, redni broj se povećava za broj bajtova koji su preneseni.
- Ovo praćenje bajtova podataka omogućuje da svaki segment bude jedinstveno identificiran i potvrđen (sequence numbers). Tada se mogu identificirati segmenti koji nedostaju.
- Zbog jednostavnosti i čitljivosti koriste se i relativni sequence numberss koje možemo vidjeti u wiresharku.

```
› [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 0]
  Sequence Number: 0    (relative sequence number)
  Sequence Number (raw): 4203655995
  [Next Sequence Number: 1    (relative sequence number)]
  Acknowledgment Number: 0
  Acknowledgment number (raw): 0
  1000 .... = Header Length: 32 bytes (8)
› Flags: 0x002 (SYN)
```



TCP pouzdanost - Gubitak podataka i ponovni prijenos

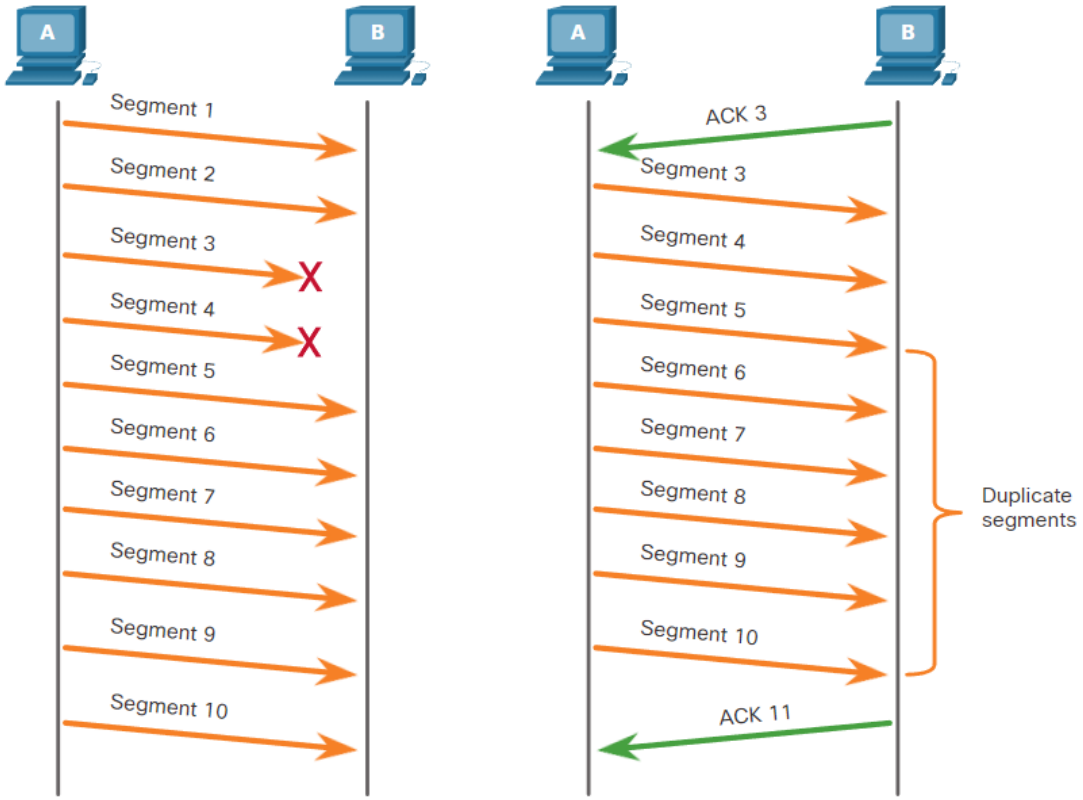
- Bez obzira koliko je dobro projektirana mreža, povremeno dolazi do gubitka podataka. TCP pruža metode upravljanja ovim gubicima segmenata. Među njima je mehanizam za ponovni prijenos segmenata za nepotvrđene podatke.

- Broj sekvence (SEQ) i broj potvrde (ACK) koriste se zajedno za potvrdu prijema bajtova podataka sadržanih u odaslanim segmentima.

- SEQ broj identificira prvi bajt podataka u segmentu koji se prenosi.
- TCP koristi ACK broj poslan natrag izvoru da označi sljedeći bajt koji primatelj očekuje primiti. To se zove očekivana potvrda (expectational acknowledgement).

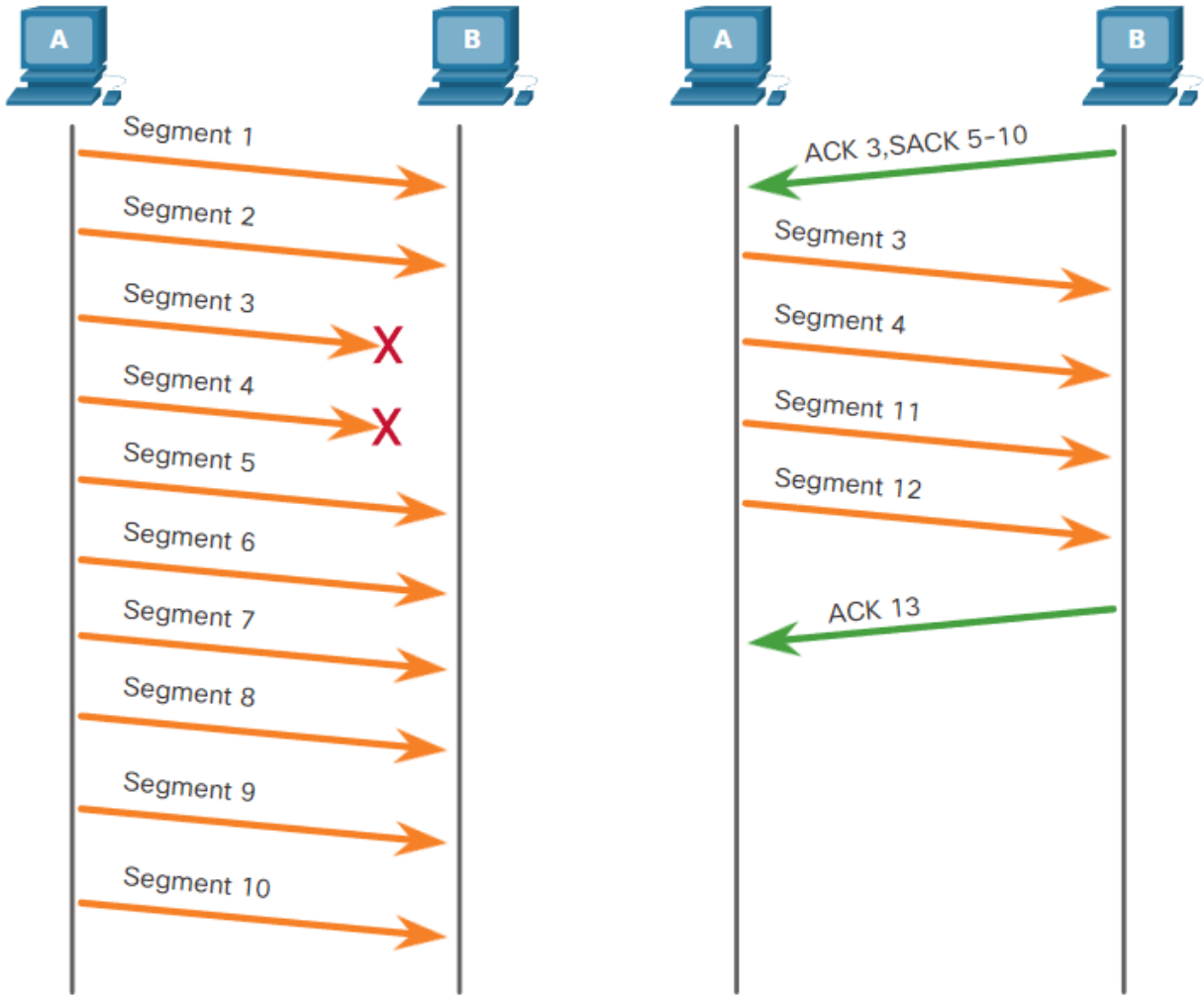
TCP pouzdanost - Gubitak podataka i ponovni prijenos

- Prije nego su implementirana poboljšanja, TCP je mogao potvrditi samo sljedeći očekivani bajt.
- Na primjer, host A šalje segmente od 1 do 10 hostu B. Ako stignu svi segmenti osim segmenata 3 i 4, host B će odgovoriti potvrdom navodeći da je sljedeći segment koji se očekuje segment 3.
- Domaćin A nema pojma jesu li neki drugi segmenti stigli ili ne. Host A bi stoga ponovno poslao segmente od 3 do 10.
- Ako su svi ponovno poslani segmenti uspješno stigli, segmenti od 5 do 10 bili bi duplikati.
- To može dovesti do kašnjenja, zagušenja i neučinkovitosti.



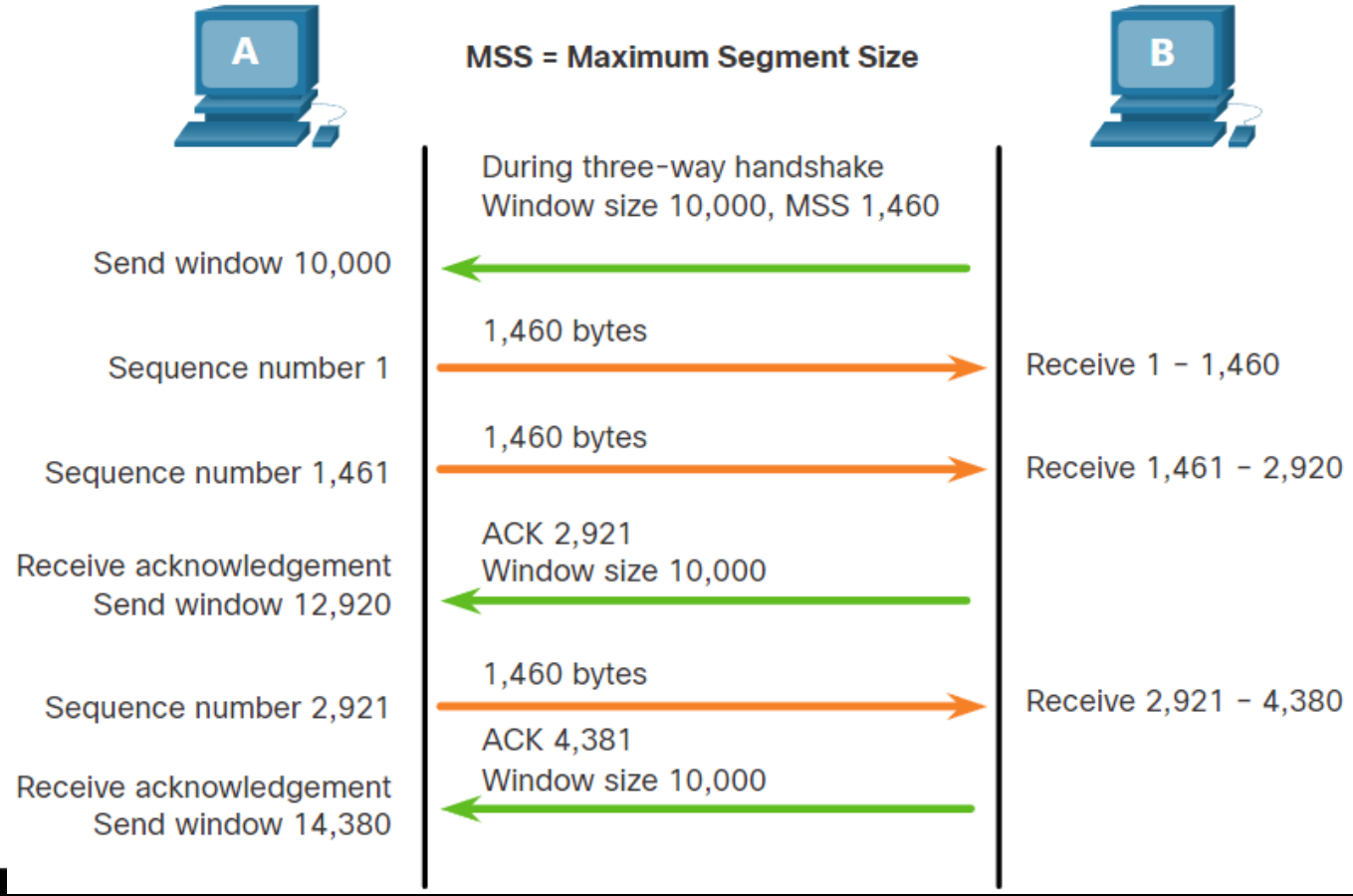
TCP pouzdanost - Gubitak podataka i ponovni prijenos

- Operacijski sustavi danas obično koriste neobaveznu (optional) TCP značajku zvanu selektivna potvrda (SACK), koja se dogovara tijekom trostrukog rukovanja.
- Ako oba hosta podržavaju SACK, primatelj može eksplicitno potvrditi koji su segmenti (bajtovi) primljeni uključujući sve diskontinuirane segmente.
- Host pošiljatelj bi stoga trebao samo ponovno poslati podatke koji nedostaju.
- Na primjer, na sljedećoj slici, ponovno koristeći jednostavne brojeve segmenata radi jasnoće, host A šalje segmente od 1 do 10 hostu B. Ako stignu svi segmenti osim segmenata 3 i 4, host B može potvrditi da je primio segmente 1 i 2. (ACK 3) i selektivno potvrdite segmente od 5 do 10 (SACK 5-10). Domaćin A trebao bi ponovno poslati samo segmente 3 i 4.
- TCP koristi mjerače vremena kako bi znao koliko dugo čekati prije ponovnog slanja segmenta



Kontrola TCP protoka - veličina prozora i potvrde

- TCP također nudi mehanizme za kontrolu protoka. Kontrola protoka pomaže u održavanju pouzdanosti TCP prijenosa podešavanjem brzine protoka podataka između izvora i odredišta za određenu sesiju.
  - Da bi se to postiglo, TCP zaglavlje uključuje 16-bitno polje koje se naziva veličina prozora (Windows size).
- Veličina prozora određuje broj bajtova koji se mogu poslati prije očekivanja potvrde.

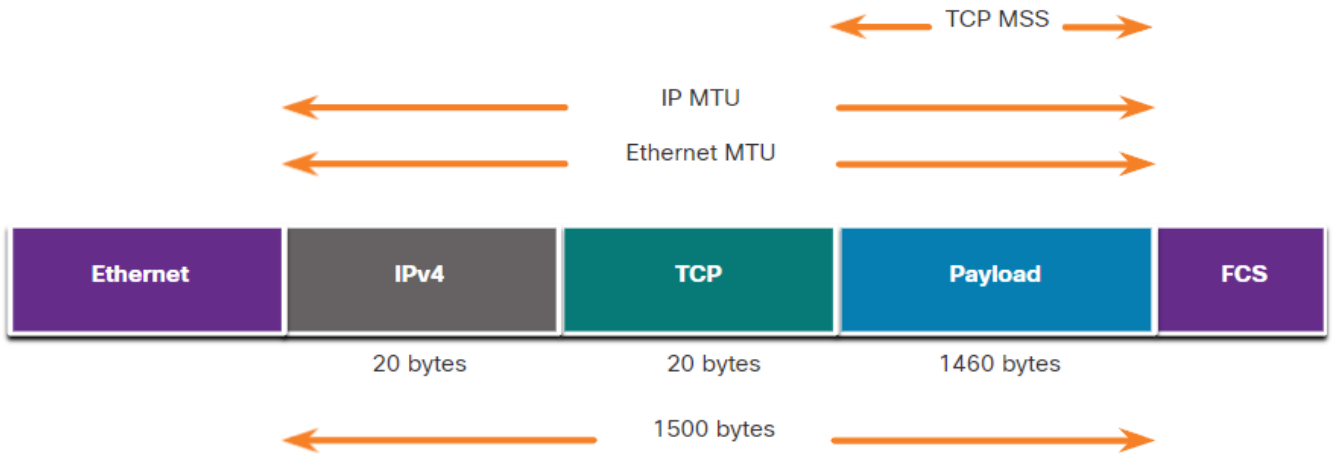


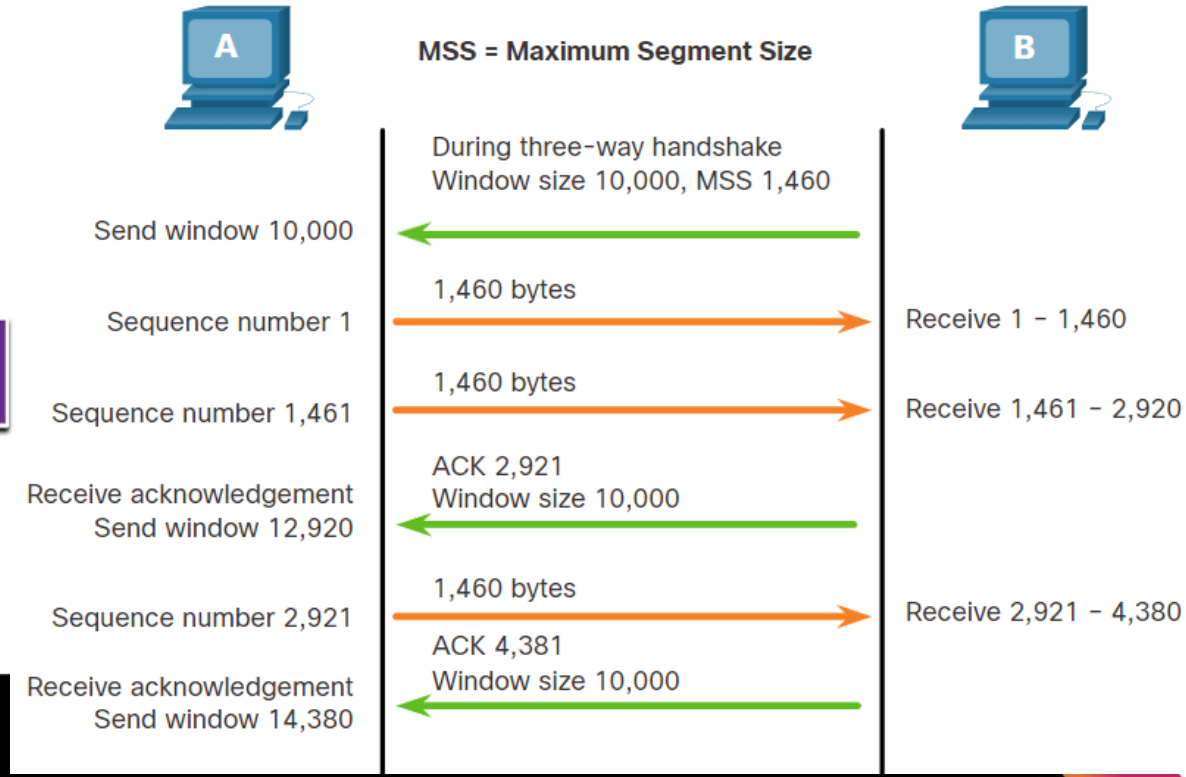
Kontrola TCP protoka - veličina prozora i potvrde

- Veličina prozora je broj bajtova koje odredišni uređaj TCP sesije može prihvatiti i obraditi odjednom.
- U ovom primjeru, početna veličina PC B prozora za TCP sesiju je 10.000 bajtova. Počevši od prvog bajta, bajt broj 1, posljednji bajt koji PC A može poslati bez primanja potvrde je 10.000 bajt. Ovo je poznato kao prozor za slanje PC A. Veličina prozora uključena je u svaki TCP segment tako da odredište može promijeniti veličinu prozora u bilo kojem trenutku ovisno o dostupnosti međuspremnika.
- Početna veličina prozora je dogovorena kada se TCP sesija uspostavi tijekom trostrukog rukovanja. Izvorni uređaj mora ograničiti broj bajtova koje šalje odredišnom uređaju na temelju veličine prozora odredišta.
- Tek nakon što izvorni uređaj primi potvrdu da su bajtovi primljeni, može nastaviti slati više podataka za sesiju.
- Odredišno slanje potvrda dok obrađuje primljene bajtove i kontinuirano prilagođavanje izvornog prozora slanja poznati su kao klizni prozori (sliding window).

TCP kontrola protoka - maksimalna veličina segmenta (MSS)

- Na slici izvor odašilje 1460 bajtova podataka unutar svakog TCP segmenta. To je obično maksimalna veličina segmenta (MSS) koju odredišni uređaj može primiti. MSS je dio polja opcija u TCP zaglavlju koje specificira najveću količinu podataka, u bajtovima, koje uređaj može primiti u jednom TCP segmentu. MSS veličina ne uključuje TCP zaglavlje. MSS se obično uključuje tijekom trosmjernog rukovanja.
- Uobičajeni MSS je 1460 bajtova kada se koristi IPv4. Računalo određuje vrijednost svog MSS polja oduzimanjem IP i TCP zaglavlja od maksimalne prijenosne jedinice na Ethernetu (MTU). Na Ethernet sučelju, zadani MTU je 1500 bajtova. Oduzimajući IPv4 zaglavlje od 20 bajtova i TCP zaglavlje od 20 bajtova, zadana veličina MSS-a bit će 1460 bajtova, kao što je prikazano na slici.



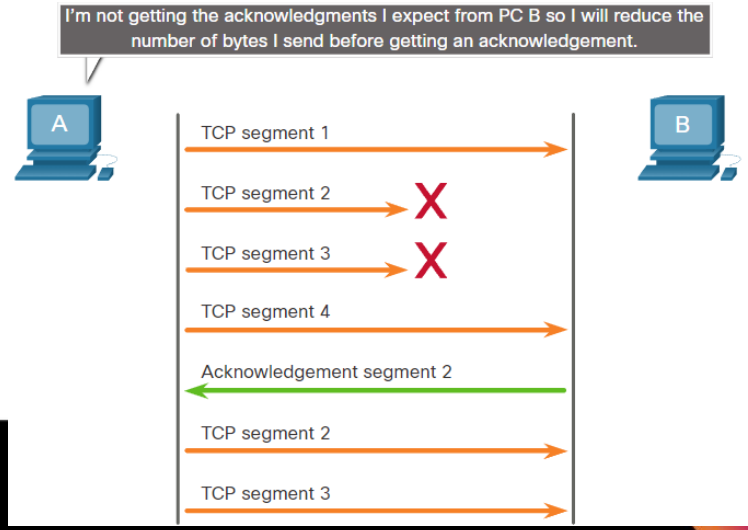


Kontrola TCP protoka - Izbjegavanje zagušenja

Kada dođe do zagušenja mreže, to rezultira odbacivanjem paketa od strane preopterećenog usmjernika. Kada paketi koji sadrže TCP segmente ne dosegnu svoje odredište, ostaju nepotvrđeni. Određivanjem brzine kojom se TCP segmenti šalju ali ne potvrđuju, izvor može pretpostaviti određenu razinu zagušenja mreže.

Kad god postoji zagušenje, dogodit će se ponovno slanje izgubljenih TCP segmenata iz izvora. Ako ponovni prijenos nije pravilno kontroliran, dodatni ponovni prijenos TCP segmenata može dodatno pogoršati zagušenje. Ne samo da se novi paketi s TCP segmentima uvode u mrežu, već će i učinak povratne sprege ponovno poslanih TCP segmenata koji su izgubljeni također povećati zagušenje. Kako bi izbjegao i kontrolirao zagušenje, TCP koristi nekoliko mehanizama za rukovanje zagušenjem, mjerača vremena i algoritama.

Ako izvor utvrdi da se TCP segmenti ili ne potvrđuju ili da nisu pravovremeno potvrđeni, tada može smanjiti broj bajtova koje šalje prije primanja potvrde. Kao što je prikazano na slici, PC A „osjeća” da postoji zagušenje i stoga smanjuje broj bajtova koje šalje prije nego što primi potvrdu od PC B.

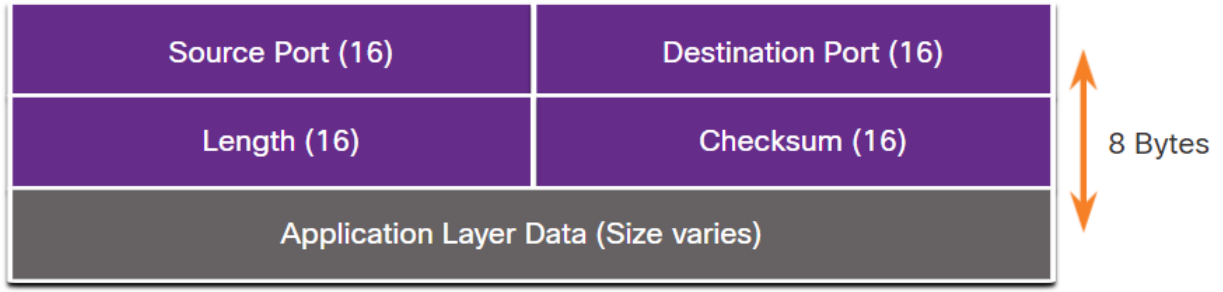


?

UDP karakteristike

UDP karakteristike uključuju sljedeće:

- Podaci se rekonstruiraju redoslijedom kojim su primljeni.
- Izgubljeni segmenti ne šalju se ponovno.
- Ne postoji uspostava sesije.
- Pošiljatelj nije obaviješten o dostupnosti resursa.



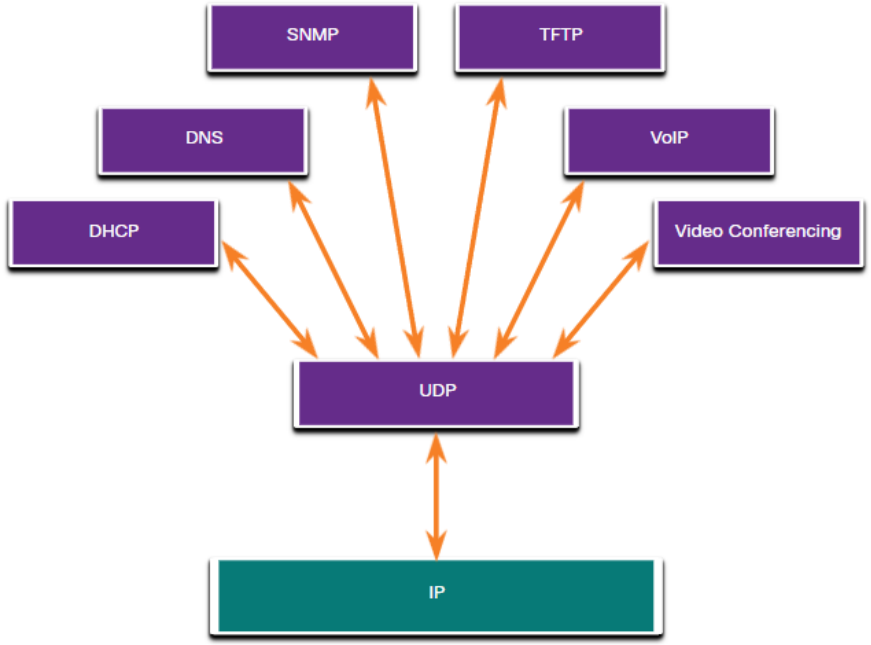
UDP je „stateless“ protokol, što znači da ni klijent ni poslužitelj ne prate stanje komunikacijske veze. Ako je potrebna pouzdanost kada se koristi UDP kao transportni protokol, to je zadatak aplikacije.

Jedan od najvažnijih zahtjeva za isporuku videa uživo i glasa putem mreže je da podaci nastave teći brzo. Video i glasovne aplikacije uživo mogu tolerirati određeni gubitak podataka s minimalnim ili nikakvim vidljivim učinkom i savršeno su prilagođene UDP-u.

UDP Header Field	Description
Source Port	16-bitno polje koje se koristi za identifikaciju izvorne aplikacije prema broju priključka (porta).
Destination Port	16-bitno polje koje se koristi za identifikaciju odredišne aplikacije prema broju priključka (porta).
Length	16-bitno polje koje označava duljinu zaglavlja UDP datagrama.
Checksum	16-bitno polje koje se koristi za provjeru grešaka u zaglavlju datagrama i podacima.

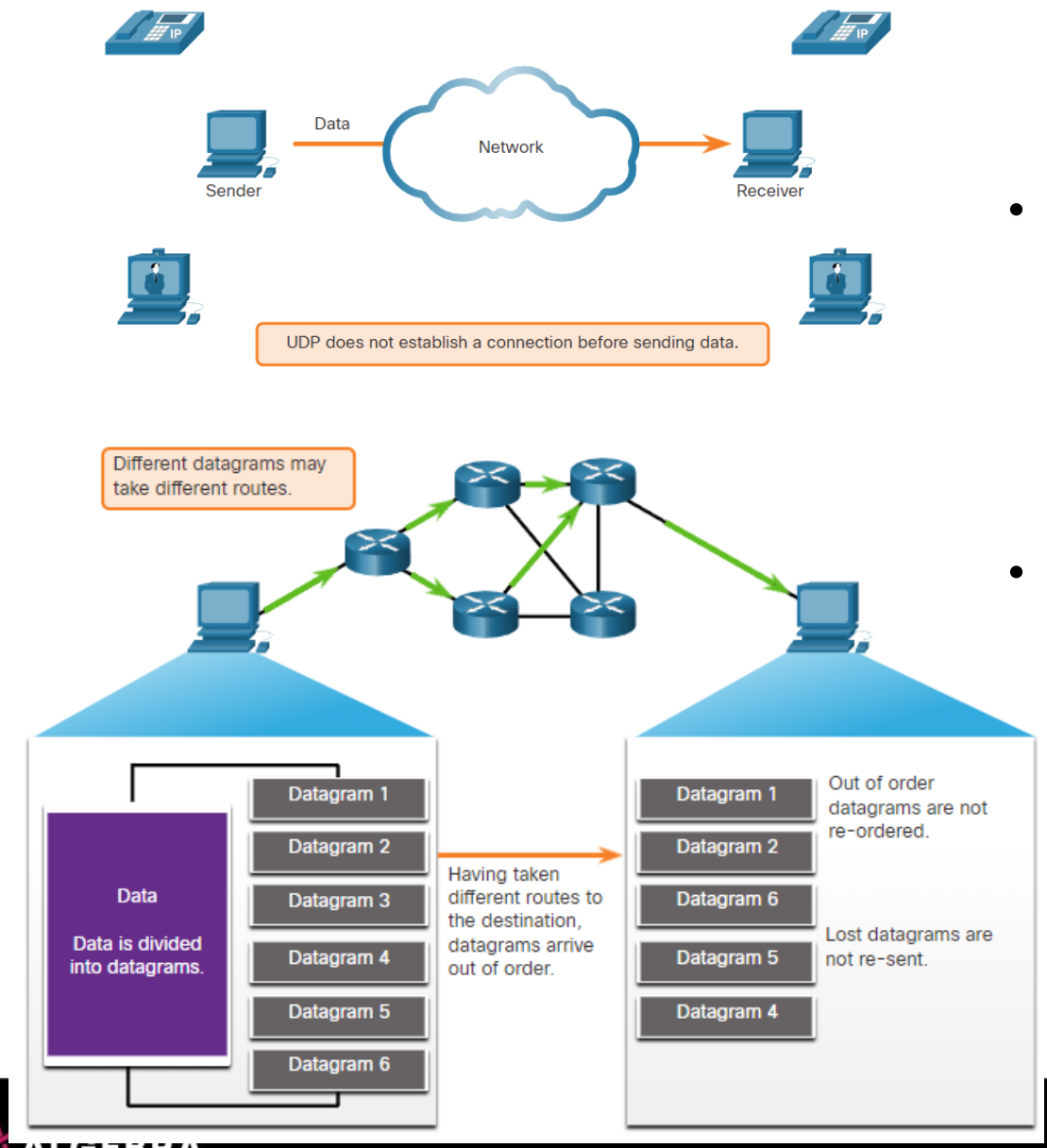
Aplikacije koje koriste UDP

- Postoje tri vrste aplikacija kojima odgovara UDP:
  - Razne multimedijalne aplikacije i aplikacije za „Live video” – Ovakve aplikacije mogu tolerirati određeni gubitak podataka, ali su osjetljive na delay. Primjeri su VoIP i live streaming video.
  - Jednostavne apliakcije koje rade po principu „request and reply” – Aplikacije koje koriste jednostavne upite gdje računalo šalje zahtjev na koji možda neće odmah dobiti odgovor, ali će ponovno poslati upit koliko puta treba. Primjer ovakvih aplikacija su DNS (Domain Name System/Service) i DHCP (Dynamic Host configuration Protocol).
  - Aplikacije koje mogu same upravljati komunikacijom na pouzdan način – Jednosmjerne komunikacije gdje kontrolu protoka, detekciju grešaka, potvrde i oporavak od greške nije potrebno ili to sve može obavljati aplikacija. Primjer:SNMP (Simple Network Management Protocol) i TFTP (Trivial File Transfer Protocol).



Iako DNS i SNMP prema zadanim postavkama koriste UDP, oba također mogu koristiti TCP. DNS će koristiti TCP ako je DNS zahtjev ili DNS odgovor veći od 512 bajtova, na primjer kada DNS odgovor uključuje mnoga razrješenja naziva. Slično, u nekim situacijama mrežni administrator možda želi konfigurirati SNMP za korištenje TCP-a.

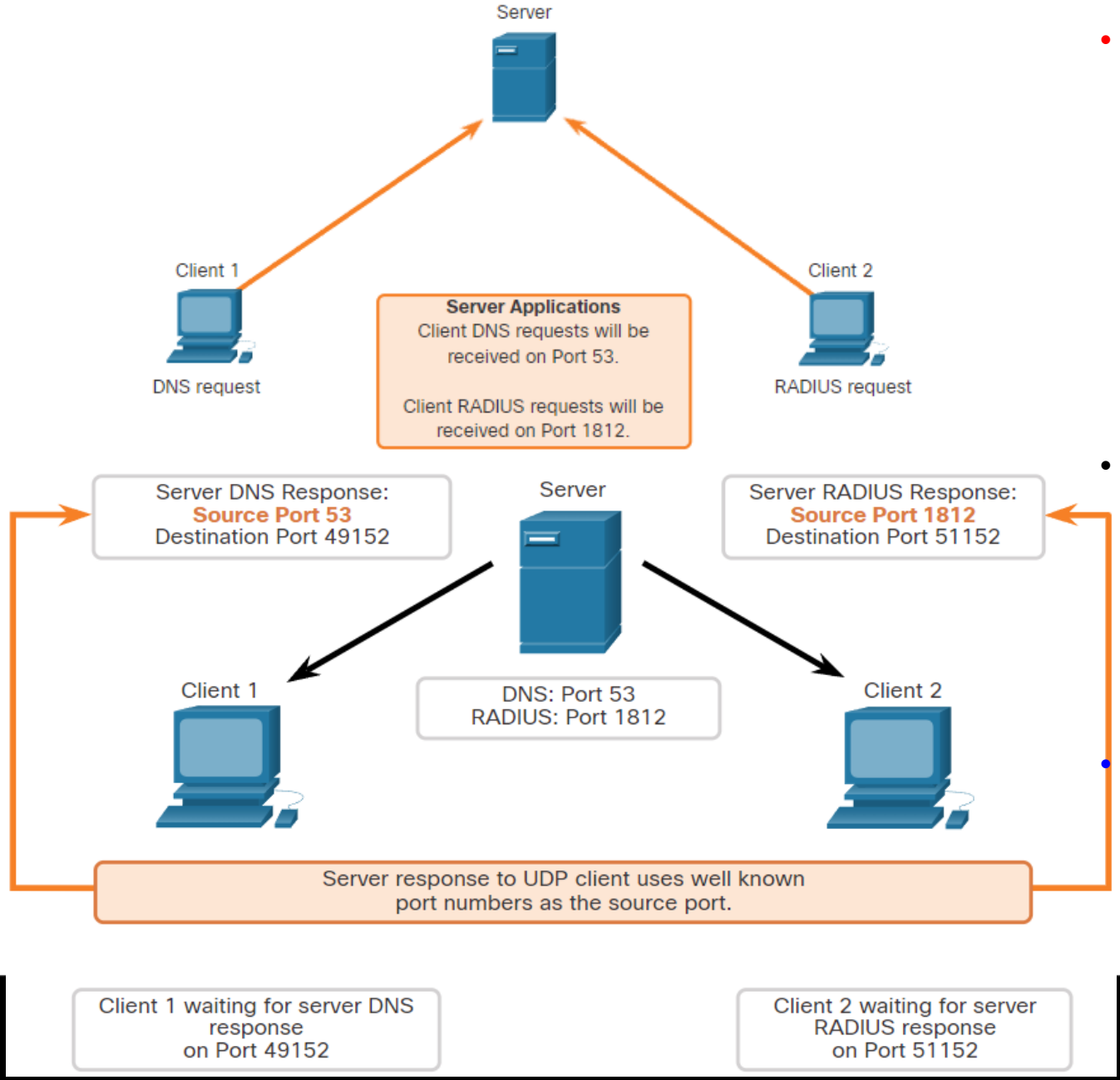
UDP komunikacija



- UDP ne uspostavlja vezu. UDP ima nizak trošak (overhead) prijenosa podataka jer ima malo zaglavlje datagrama i nema mehanizme upravljanja prometom

- Poput segmenata s TCP-om, kada se UDP datagrami šalju na odredište, često idu različitim putovima i stižu pogrešnim redoslijedom. UDP ne prati redne brojeve na način na koji to radi TCP. UDP nema načina da promijeni redoslijed datagrama u njihov originalni redoslijed prije slanja.
- Stoga UDP jednostavno sastavlja podatke redoslijedom kojim su primljeni i prosljeđuje ih aplikaciji. Ako je slijed podataka važan za aplikaciju, aplikacija mora identificirati ispravan slijed i odrediti kako se podaci trebaju obrađivati.

Procesi i zahtjevi UDP poslužitelja i klijenta



- Poput aplikacija temeljenih na TCP-u, aplikacijama poslužitelja temeljenim na UDP-u dodjeljuju se dobro poznati ili registrirani brojevi portova, kao što je prikazano na slici. Kada se ove aplikacije ili procesi izvode na poslužitelju,one prihvaćaju podatke koji se podudaraju s dodijeljenim brojem priključka. Kada UDP primi datagram namijenjen jednom od ovih portova (broj priključka), prosljeđuje odgovarajućoj aplikaciji na temelju broja porta.

- Kao i kod TCP-a, komunikaciju klijent-poslužitelj pokreće aplikacija klijenta koja zahtijeva podatke od procesa poslužitelja. Proces koji koristi UDP na klijentu dinamički odabire broj priključka iz niza brojeva priključaka i koristi ga kao izvorni priključak za razgovor. Odredišni priključak obično je dobro poznati ili registrirani broj priključka.

- Nakon što je klijent odabrao izvorni i odredišni port, isti par portova koristi se u zaglavlju svih datagrama u transakciji. Za podatke koji se vraćaju klijentu s poslužitelja, izvorni i odredišni brojevi porta u zaglavlju datagrama su obrnuti.

?

UDP vs TCP

- Neke aplikacije mogu tolerirati određeni gubitak podataka tijekom prijenosa preko mreže, ali kašnjenja u prijenosu su neprihvatljiva. Za ove je aplikacije UDP bolji izbor jer ima manji „overhead“ (manje opisnih podataka od TCP-a).
- UDP se koristi za aplikacije kao što je Voice over IP (VoIP). Potvrde i ponovno slanje usporili bi isporuku i glasovni razgovor učinili neprihvatljivim.

- UDP također koriste aplikacije koje rade po prinicpu „request-and-reply“ gdje je količina podataka minimalna, a ponovni prijenos se može obaviti brzo. Na primjer, DNS koristi UDP za komunikaciju. Klijent zahtijeva IPv4 i IPv6 adrese za poznato domensko ime od DNS poslužitelja. Ako klijent ne dobije odgovor u unaprijed određenom vremenu, jednostavno ponovno šalje zahtjev.

- Na primjer, ako jedan ili dva segmenta video streama uživo ne stignu, to stvara prekid u streamu. To se može pojaviti kao izobličenje slike ili zvuka, ali korisniku to možda neće biti vidljivo. Ako bi odredišni uređaj morao računati na izgubljene podatke, tok bi mogao biti odgođen dok se čekaju ponovni prijenosi, što bi uzrokovalo veliku degradaciju slike ili zvuka. U ovom slučaju, bolje je prikazati najbolju moguću sliku/zvuk s datagramima koji su primljeni nego imati potpuni prekid jer želimo primiti sve podatke...dakle priroda komunikacije je takva da je pretjerano inzistiranje na pouzdanosti nepotrebno, čak i štetno.

UDP vs TCP

- Za neke aplikacije važno je da svi podaci stignu i da se mogu obraditi u odgovarajućem redoslijedu. Za ove vrste aplikacija, TCP se koristi kao transportni protokol. Na primjer, aplikacije kao što su baze podataka, web preglednici i klijenti e-pošte zahtijevaju da svi podaci koji se šalju stignu na odredište u izvornom stanju. Podaci koji nedostaju mogu učiniti komunikaciju nepotpunom ili nečitljivom. Na primjer, važno je kada pristupate bankovnim podacima putem weba provjeriti jesu li sve informacije ispravno poslane i primljene.

- Programeri aplikacija moraju odabrati koji je tip transportnog protokola prikladan na temelju zahtjeva aplikacija. Video se može poslati preko TCP-a ili UDP-a. Aplikacije koje emitiraju pohranjeni audio i video obično koriste TCP. Aplikacija koristi TCP za izvođenje međuspremnika (buffer), ispitivanje propusnosti i kontrolu zagušenja, kako bi se bolje kontroliralo korisničko iskustvo.
- Na primjer, ako vaša mreža iznenada ne može podržati propusnost potrebnu za gledanje filma na zahtjev, aplikacija pauzira reprodukciju. Tijekom pauze možete vidjeti poruku "**buffering**..." dok TCP radi na ponovnom uspostavljanju toka. Kada su svi segmenti u redu i minimalna razina propusnosti je vraćena, vaša TCP sesija se nastavlja i film se nastavlja reproducirati.

- Video i glas u stvarnom vremenu obično koriste UDP, ali mogu koristiti i TCP ili oba protokola. Aplikacija za videokonferencije može koristiti UDP „po defaultu“, ali budući da mnogi vatrozidi blokiraju UDP, aplikacija se također može poslati preko TCP-a.

UDP vs TCP

UDP

VoIP
(IP telephony)

DNS
(Domain Name
Resolution)

Required protocol properties:

- Fast
- Low overhead
- Does not require acknowledgements
- Does not resend lost data
- Delivers data as it arrives

TCP

SMTP/IMAP
(Email)

HTTP/HTTPS
(World Wide Web

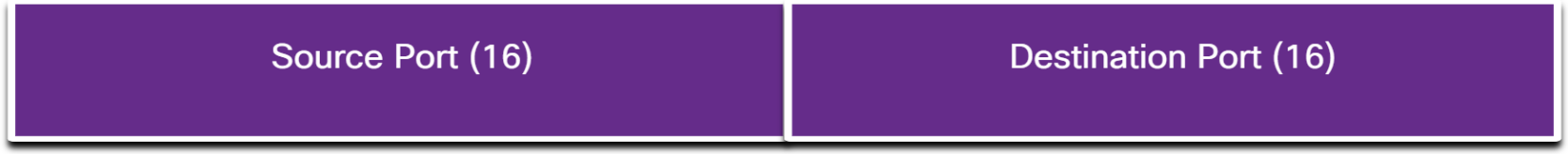
Required protocol properties:

- Reliable
- Acknowledges data
- Resends lost data
- Delivers data in sequenced order

?

Brojevi priključaka (port numbers) - više odvojenih komunikacija

- Bez obzira koja se vrsta podataka prenosi, i TCP i UDP koriste brojeve portova.
- TCP i UDP protokoli transportnog sloja koriste brojeve portova za upravljanje višestrukim istovremenim „razgovorima”.
- Polja zaglavlja izvornog priključka i odredišnog priključka imaju po 2 bajta (16 bitova)



- Izvorni broj priključka povezan je s izvornom aplikacijom na lokalnom glavnom računalu, dok je odredišni broj priključka povezan s odredišnom aplikacijom na udaljenom glavnom računalu.

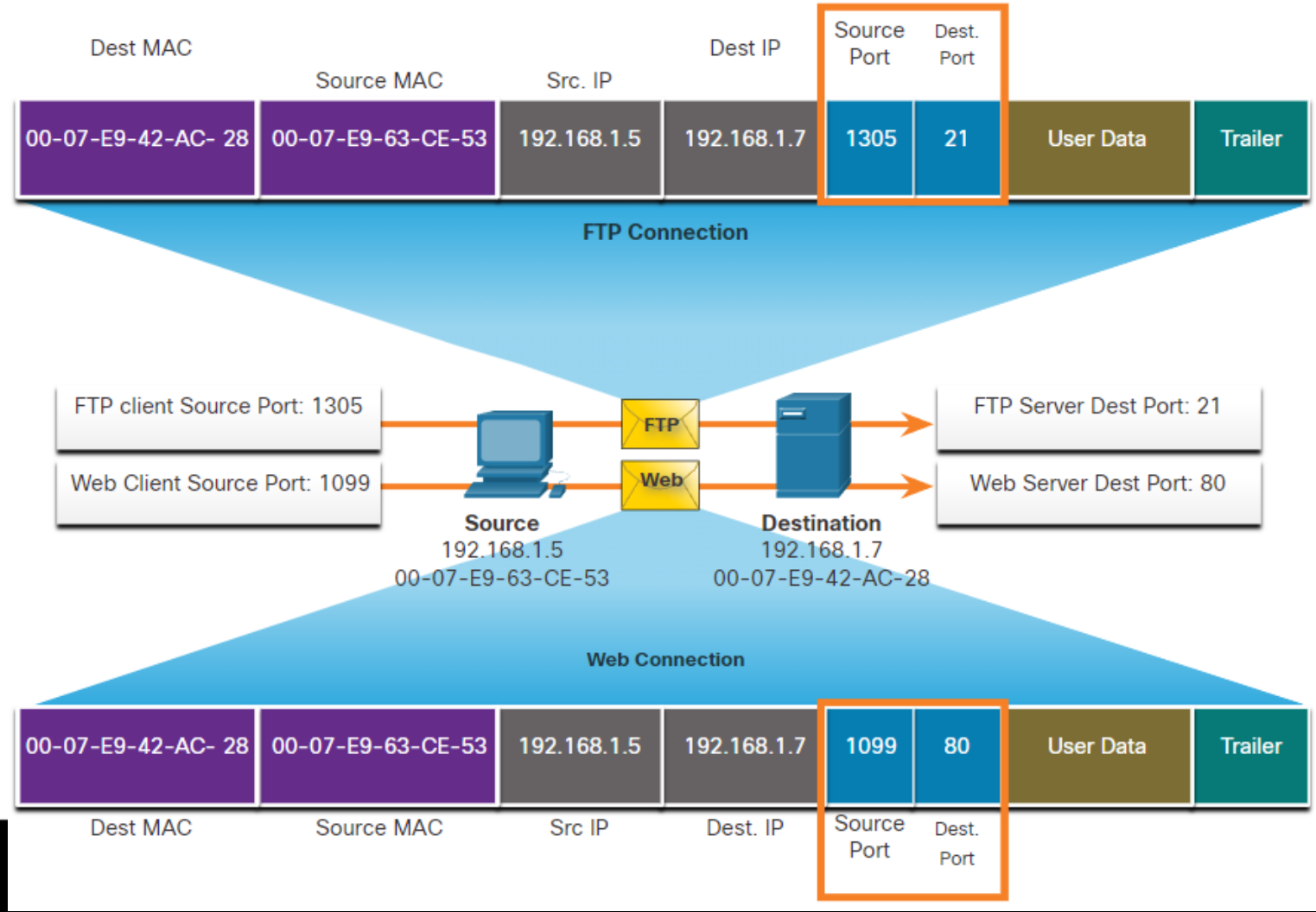
- Na primjer, pretpostavimo da host pokreće zahtjev prema web stranici na web poslužitelju. Kada računalo pokrene zahtjev za web-stranicu, dinamički generira izvorni broj porta kako bi jedinstveno identificirao razgovor. Svaki zahtjev koji generira host koristit će drugačiji dinamički kreirani izvorni broj porta. Ovaj proces omogućuje istovremeno odvijanje više razgovora.

- U zahtjevu, broj odredišnog porta je ono što identificira vrstu usluge koja se traži od odredišnog web poslužitelja. Na primjer, kada klijent navede port 80 kao odredišni port, poslužitelj koji prima poruku zna da se traži web usluga.

- Poslužitelj može ponuditi više od jedne usluge istovremeno, kao što su web usluge na portu 80, dok nudi uspostavljanje veze protokola za prijenos datoteka (FTP) na portu 21.

Socket Pairs

- Izvorni i odredišni portovi upisani su unutar segmenta/datagrama. Segmenti se zatim enkapsuliraju unutar IP paketa. IP paket sadrži IP adresu izvora i odredišta.
- Kombinacija izvorne IP adrese i broja izvornog porta ili odredišne IP adrese i odredišnog broja porta poznata je kao „utičnica“ ili „socket“.



Socket Pairs

- U primjeru na slici na prethodnom slajdu, FTP zahtjev koji generira računalo uključuje MAC adrese sloja 2 i IP adrese sloja 3. Zahtjev također identificira izvorni broj porta 1305 (tj. dinamički generiran od strane računala) i odredišni port 21 koji identificira FTP uslugu.
- Izvorišno računalo je također zatražilo web stranicu od poslužitelja koristeći iste adrese sloja 2 i sloja 3 . Međutim, koristi izvorni broj porta 1099 (tj. dinamički generiran od strane računala) i odredišni port 80 koji identificira web uslugu.

- Utičnica (socket) se koristi za identifikaciju poslužitelja i usluge koju klijent zahtijeva. Klijentska utičnica može izgledati ovako: 192.168.1.5:1099 (pri čemu 1099 predstavlja broj izvorišnog priključka), a utičnica (socket) na web poslužitelju može izgledati ovako: 192.168.1.7:80

- **Zajedno, ove dvije utičnice se kombiniraju u par utičnica: 192.168.1.5:1099, 192.168.1.7:80**

- **Utičnice omogućuju da se višestruki procesi, koji se izvode na klijentu, razlikuju jedni od drugih, te da se međusobno razlikuju višestruke veze s procesom poslužitelja.**

- Izvorni broj porta služi kao povratna adresa za aplikaciju na računalu kada server šalje odgovor. Transportni sloj prati komunikaciju i prosljeđuje podatke točno određenim procesima na računalu i to upravo na temelju ovih „socketa“.

Grupe brojeva portova

- Internet Assigned Numbers Authority (IANA) organizacija je za standarde odgovorna za dodjelu različitih standarda adresiranja, uključujući 16-bitne brojeve portova. 16 bitova koji se koriste za identifikaciju brojeva izvorišnog i odredišnog porta omogućavaju raspon portova od 0 do 65535.

Port Group	Number Range	Description
Well-known Ports	**0 to 1,023**	•Ovi brojevi priključaka rezervirani su za uobičajene ili popularne usluge i aplikacije kao što su web preglednici, klijenti e-pošte i klijenti za daljinski pristup. •Definirani dobro poznati priključci za uobičajene poslužiteljske aplikacije omogućuju klijentima da lako identificiraju potrebnu povezanu uslugu.
Registered Ports	**1,024 to 49,151**	•Ove brojeve priključaka IANA dodjeljuje subjektu koji je zatražio broj priključka za korištenje s određenim procesima ili aplikacijama. •Ti su procesi prvenstveno pojedinačne aplikacije koje je korisnik odabrao instalirati, a ne uobičajene aplikacije koje bi dobile dobro poznati broj priključka. •Na primjer, Cisco je registrirao port 1812 za svoj proces provjere autentičnosti RADIUS poslužitelja.
Private and/or **Dynamic Ports**	**49,152 to 65,535**	•Ti su priključci poznati i kao efemerni priključci (ephemeral ports)-"koji traju kratko". •Klijentov OS obično dinamički dodjeljuje brojeve portova kada se pokrene veza s uslugom. •Dinamički port se tada koristi za identifikaciju klijentske aplikacije tijekom komunikacije.

Well-Known Port Numbers

Port Number	Protocol	Application
20	TCP	File Transfer Protocol (FTP) - Data
21	TCP	File Transfer Protocol (FTP) - Control
22	TCP	Secure Shell (SSH)
23	TCP	Telnet
25	TCP	Simple Mail Transfer Protocol (SMTP)
53	UDP, TCP	Domain Name System (DNS)
67	UDP	Dynamic Host Configuration Protocol (DHCP) - Server
68	UDP	Dynamic Host Configuration Protocol - Client
69	UDP	Trivial File Transfer Protocol (TFTP)
80	TCP	Hypertext Transfer Protocol (HTTP)
110	TCP	Post Office Protocol version 3 (POP3)
143	TCP	Internet Message Access Protocol (IMAP)
161	UDP	Simple Network Management Protocol (SNMP)
443	TCP	Hypertext Transfer Protocol Secure (HTTPS)

Neke aplikacije mogu koristiti i TCP i UDP. Na primjer, DNS koristi UDP kada klijenti šalju zahtjeve DNS poslužitelju. Međutim, komunikacija između dva DNS poslužitelja koristi TCP.

Well-Known Port Numbers

- Neobjašnjive TCP veze mogu predstavljati veliku sigurnosnu prijetnju. Mogu značiti da je nešto ili netko povezan s lokalnim hostom. Ponekad je potrebno znati koje su aktivne TCP veze otvorene i izvode se na umreženom glavnom računalu. `Netstat` je važan mrežni uslužni program koji se može koristiti za provjeru tih veza.
- Prema zadanim postavkama, naredba *`netstat`* pokušat će razriješiti IP adrese u nazive domena i brojeve priključaka za dobro poznate aplikacije. Opcija *`-n`* može se koristiti za prikaz IP adresa i brojeva priključaka u njihovom numeričkom obliku.

```
C:\> netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.1.124:3126     192.168.0.2:netbios-ssn   ESTABLISHED
  TCP    192.168.1.124:3158     207.138.126.152:http      ESTABLISHED
  TCP    192.168.1.124:3159     207.138.126.169:http      ESTABLISHED
  TCP    192.168.1.124:3160     207.138.126.169:http      ESTABLISHED
  TCP    192.168.1.124:3161     sc.msn.com:http           ESTABLISHED
  TCP    192.168.1.124:3166     www.cisco.com:http        ESTABLISHED
(output omitted)
C:\>
```

netstat
netstat -n

?