Komunikim të dhënash dhe rrjetat kompjuterike

Kapitulli 4: Shtresa Transport

Prof. Asoc. Dr. Ezmolda Barolli

Përmbajtja

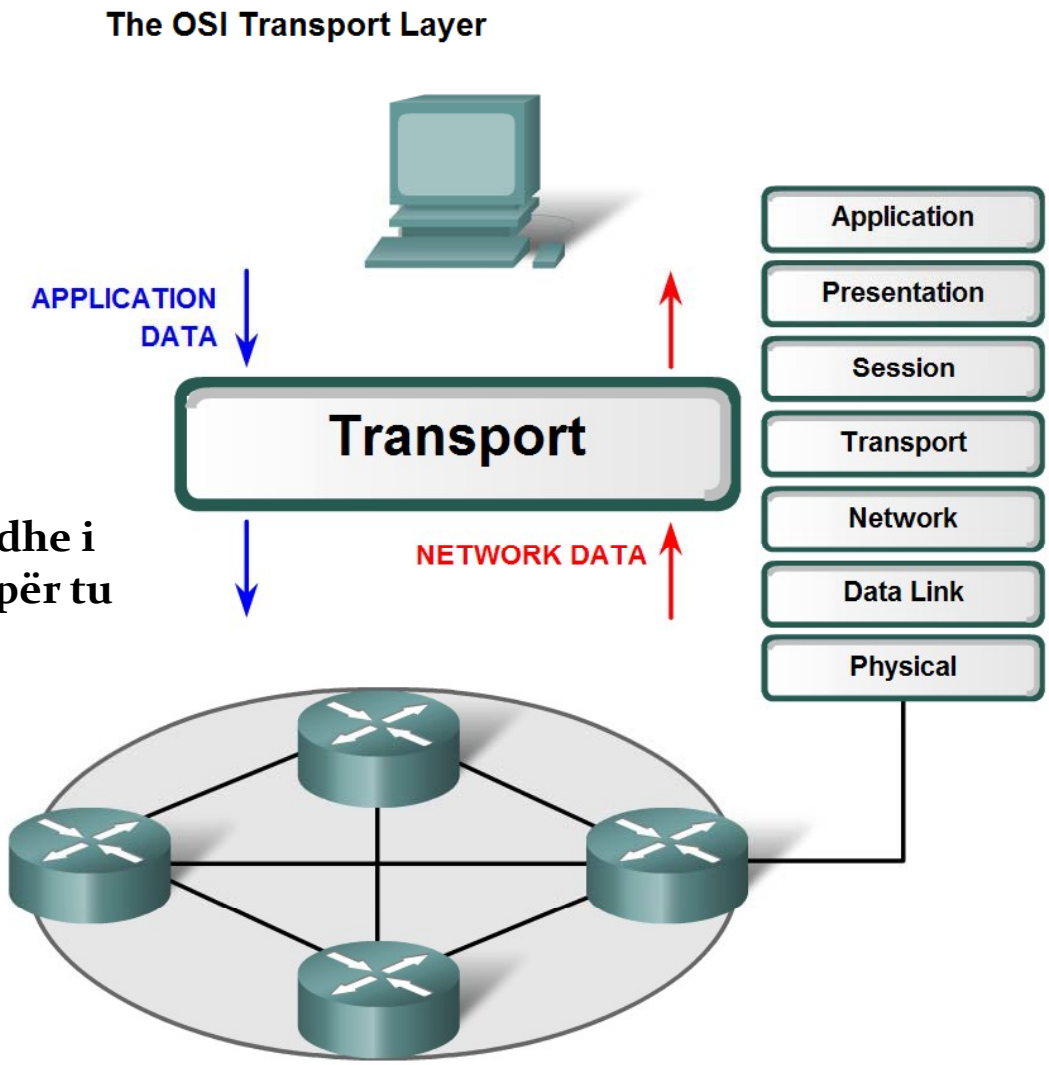
1. Hyrje
2. Roli i shtresës Transport
3. Protokolli TCP – komunikimi i besueshëm
4. Menaxhimi i sesioneve TCP
5. Protokolli UDP – komunikimi me ngarkesë të vogël

1.1. Hyrje

- **Komunikim i besueshëm** dhe **i plotë**.
- **Një pajisje e vetme - shumë shërbime dhe aplikacione**.
- Do të ekzaminojmë **rolin e shtresës Transport** - përgjegjëse për **transferimin nga skaji-në-skaj** të të dhënave të aplikacioneve.
  - **Shumë aplikacione** të komunikojnë në rrjet në **të njëjtën kohë** nga **e njëjta pajisje** .
  - Të dhënat **merren me saktësi ,** në **rradhën e duhur**, nga **aplikacioni i duhur**.
  - Mekanizma të **menaxhimit të gabimeve**

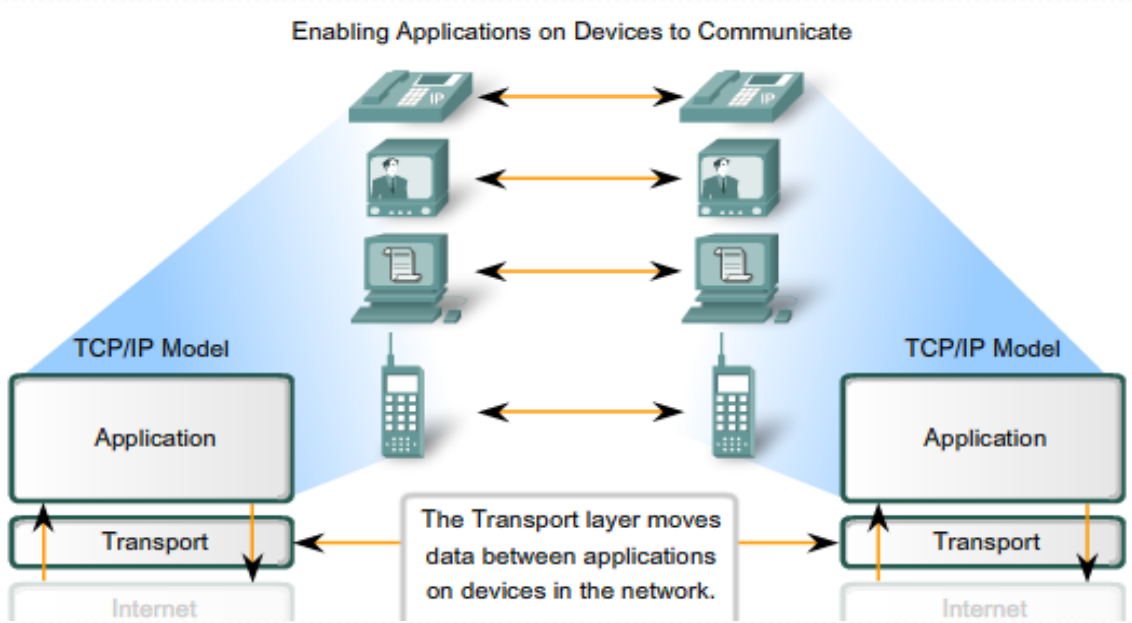
2.1. Qëllimi i Shtresës Transport

Shtresa Transport përgatit të dhënat e aplikacionit për tu transportuar përgjatë rrjetit dhe i përpunon të dhënat e rrjetit për tu përdorur nga aplikacioni.



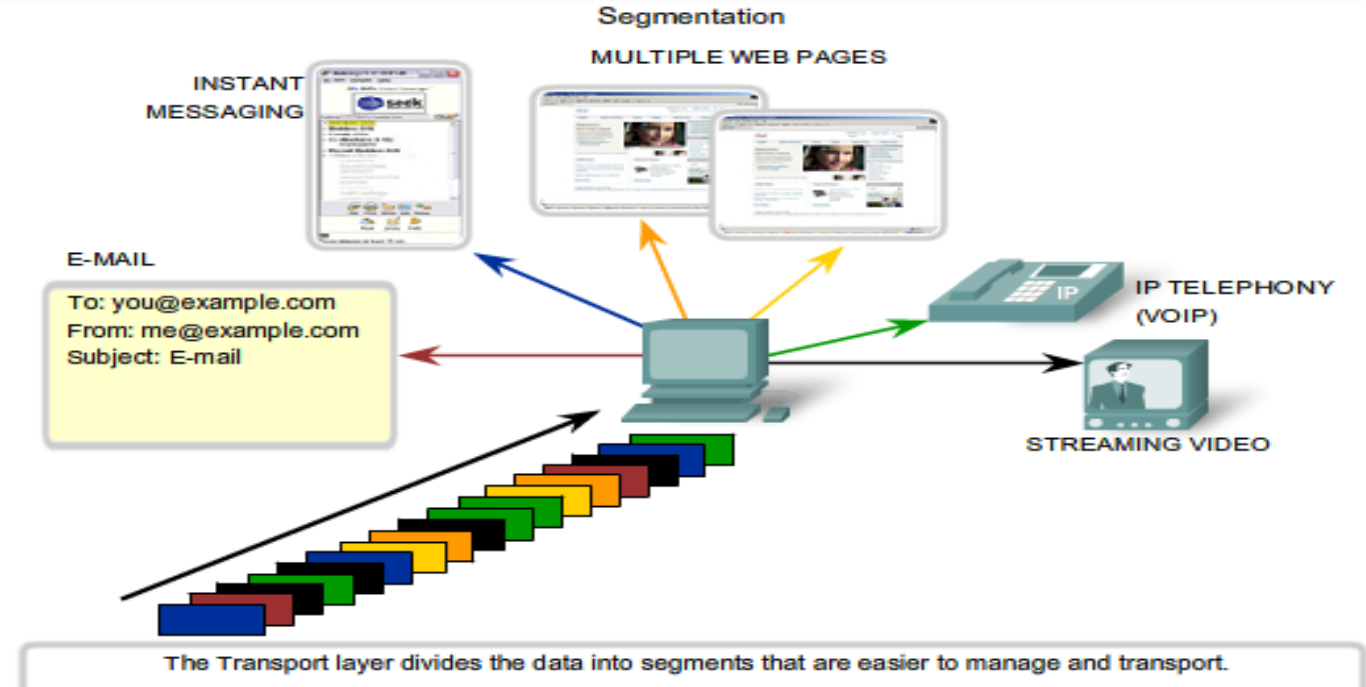
2.2. Qëllimi i Shtresës Transport

- **Përgjegjësitë e saj kryesore për të përmbushur këtë janë:**
  - **Gjurmimi i secilit komunikim** midis aplikacioneve -> për të mbajtur shumë linja komunikimi
  - **Segmentimi** i të dhënave dhe **menaxhimi** i secilës pjesë -> përzierja e komunikimeve të ndryshme
  - **Riasemblimi** i segmenteve në rrjedhë të dhënash të aplikacionit
  - **Identifikimi i aplikacioneve** të ndryshme -> numrat e portave



2.3. Qëllimi i Shtresës Transport

- **Aplikacionet** e ndryshme **kanë kërkesa të ndryshme për të dhënat** - ka **shumë protokolle** të shtresës Transport:
  - **Të merren të gjithë të dhënat**
  - **Mund të tolerohen humbjet**
  - **Të arrijnë në një sekuencë specifike**
- Protokollet ofrojnë **funksione bazike** ose funksione **shtesë**



2.4. Kontrolli i bashkëbisedimit

- **Disa protokolle** të shtresës transport sigurojnë:
  - **Komunikim conncetion-oriented**: Krijimi i **sesioneve** të komunikimit
  - **Shpërndarje e besueshme**: **Garantimi** që të gjitha **të dhënat** e dërguara **arrijnë** në destinacion.
  - **Rindërtimi i të dhënave sipas radhës**: **Numërtimi** dhe **renditja** (sequencing) e segmenteve ofron zgjidhje të problemit
  - **Kontrolli i rrjedhës**: Në varësi të **kufizimeve në burime ritmi i dërgimit** të të dhënave mund të **ulet**.

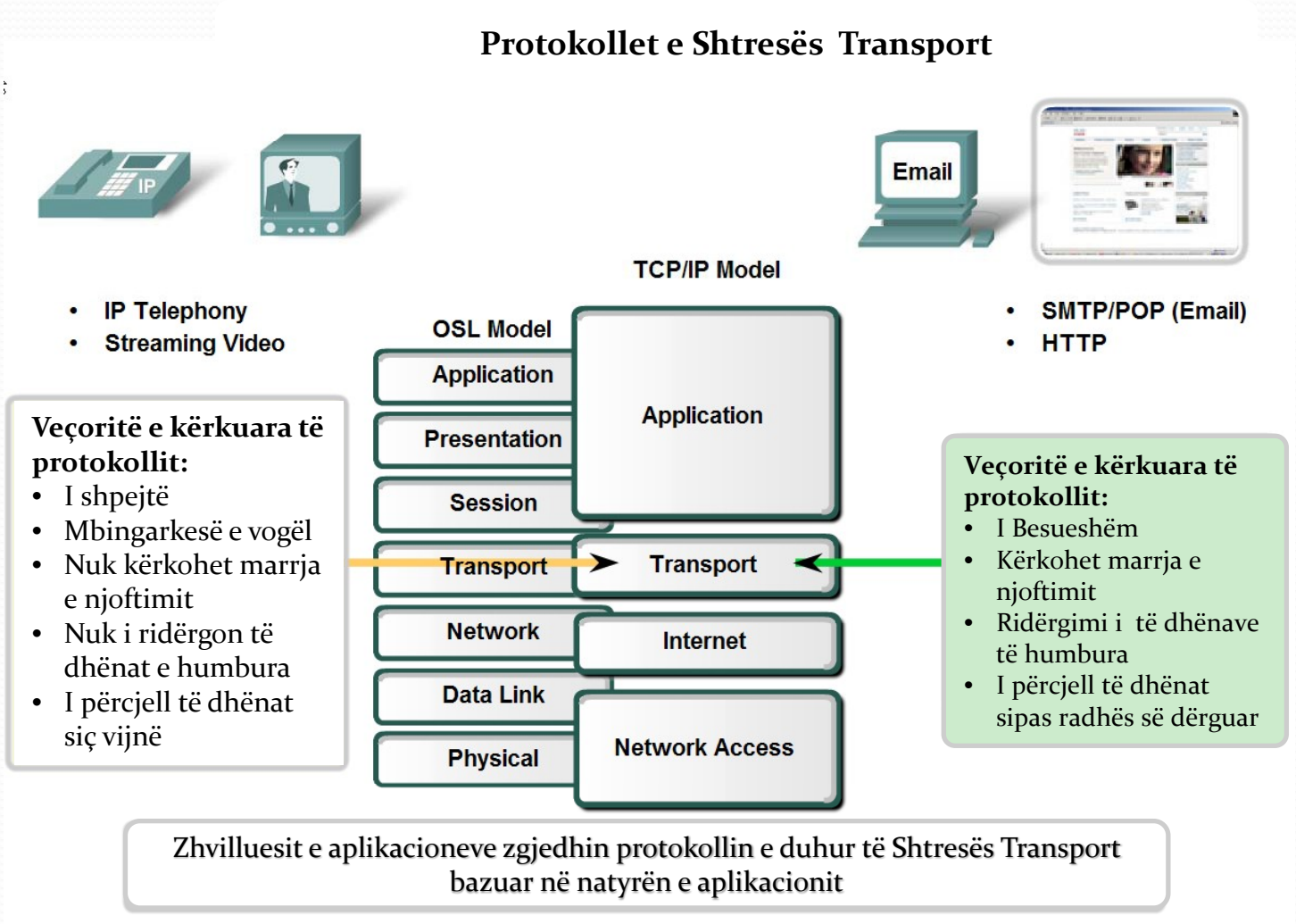
2.5. Realizimi i Dërgimit të Besueshëm

- Në shtresën transport **tre proçeset bazë të besueshmërisë** janë:
  - **Gjurmimi** i të dhënave të transmetuara
  - Marrja e **njoftimit** për të dhënat e transmetuara
  - **Ritransmetimi** i të dhënave për të cilat nuk merret njoftim
- **Hosti marrës** duhet të gjurmojë të dhënat siç i merr dhe **të njoftojë** marrjen e tyre. **Dërguesi** do të **ritransmetojë** n.q.s. nuk merr njoftim.
- Këto proçese besueshmërie krijojë **ngarkesë shtesë** në burimet e rrjetit

2.6. Realizimi i Dërgimit të Besueshëm

- Për shkak të kësaj mbingarkese, është e rëndësishme të përcaktohet **nevoja për transmetim të besueshëm**.
- **Disa** aplikacione e **kanë të nevojshme** (p.sh. web), ndërsa **disa** të tjera **jo** (p.sh. zëri)
- Ka protokolle që specifikojnë **metoda** për transmetim të besueshëm, që **garanton dërgimin** ose që bën **dërgimin më të mirë**.
- **Dërgimi më i mirë** referohet si **jo i besueshëm**, sepse nuk merret njoftim nëse të dhënat janë marrë nga destinacioni

2.7. Realizimi i Dërgimit të Besueshëm



2.8. TCP dhe UDP

- Të dy protokollet **menaxhojnë komunikimin** e aplikacioneve
  - **TCP:** Transmission Control Protocol
  - **UDP:** User Datagram Protocol

2.8. TCP dhe UDP

Header-at e TCP dhe UDP

FUSHAT e SEGMENTIT & HEADER-it të TCP

Bit 0	Bit 15 Bit 16	Bit 31
Source Port (16)	Destination Port (16)	
Sequence Number (32)		
Acknowledgement Number (32)		
Header Length (4) Reserved (6) Code Bits (6)	Window (16)	
Checksum (16)	Urgent (16)	
Options (0 or 32 if any)		
APPLICATION LAYER DATA SEGMENT (Size varies)		

20 Bytes

FUSHAT e DATAGRAMËS & HEADER-it të UDP

Bit (0)	Bit (15) Bit (16)	Bit (31)
Source Port (16)	Destination Port (16)	
Length (16)	Checksum (16)	
APPLICATION LAYER DATA SEGMENT (Size varies)		

8 Bytes

2.9. Transmission Control Protocol (TCP)

- **TCP** është protokoll **connection-oriented**, i cili përdor **shtesën e ngarkesës** për të shtuar funksionalitetet
- **Funksionalitetet shtesë** të specifikuara nga TCP janë: ruajtja e **radhës** së dërgesës, dërgim i **besueshëm** dhe **kontrolli** i rrjedhës
- **Aplikacione** që përdorin TCP-në janë:
  - Web browsers
  - E-mail
  - Transferimi i file-ve

2.10. User Datagram Protocol (UDP)

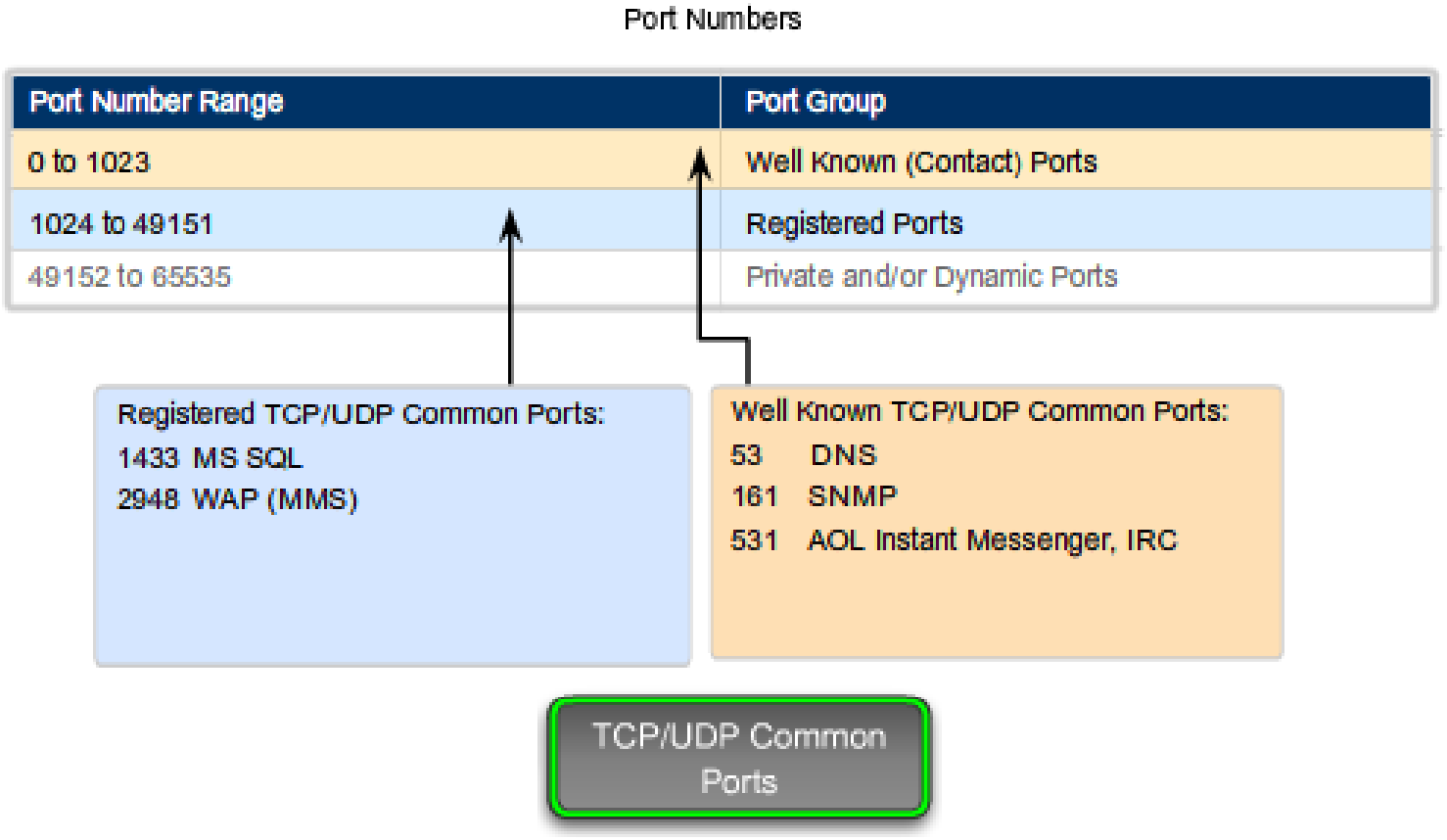
- **UDP** është i thjeshtë, protokoll që nuk bazohet në vendosjen e lidhjes (**connectionless**), siguron një dërgesë me **ngarkesë shtesë të vogël**.
- Pjesët e komunikimit në UDP quhen **datagrama**. Këto datagrama dërgohen si "**rasti më i mirë**" (best effort) nga ky protokoll i shtresës Transport.
- **Aplikacionet** që përdorin UDP përfshijnë:
  - Domain Name System (DNS)
  - Video Streaming
  - Voice over IP (VoIP)

2.11. Identifikimi i Komunikimit

- Për të bërë dallimin e segmenteve dhe datagram-eve për secilin aplikacion, si **TCP** ashtu edhe **UDP kanë në header fusha unike të cilat identifikojnë aplikacionet**. Këto identifikues unik janë **numrat e portave**.
- **Proçeset server** kanë të caktuar **numra portash statikë**, **klientët** zgjedhin në mënyrë **dinamike** një numër porte ku do të zhvillojnë komunikimin.
- **Kombinimi i numrit të portës** së shtresës së transportit **me adresës IP** të caktuar për hostin nga shtresa Network identifikojnë në mënyrë unike një proces specifik që ekzekutohet në host . Ky kombinim quhet **socket**.

2.12. Adresimi i Portave

- **Internet Assigned Numbers Authority** (IANA-Autoriteti i Caktimit të Numrave të Internetit) përcakton numrat e portave



2.13. Adresimi i Portave

- **Portat e Mirënjohura** (numrat **0-1023**): janë **të rezervuara** për **shërbime** dhe **aplikacione të përgjithshme** (aplikacionet **server**). Zakonisht kërkohen të drejta administratori për përdorimin e tyre.
- **Porta të Rregjistruara** (numrat **1024-49151**): Këto porta janë për **aplikacione të veçanta** që **përdoruesi** vendos të instalojë. Kur nuk përdoren si burime për server, këto porta mund të përdoren nga hosti në rolin e klientit për tu përzgjedhur në mënyrë dinamike si porta burim
- **Portat Dinamike** ose **Private** (numrat **49152-65535**): Gjithashtu njihen si **Portat Afatshkurtër**. Zakonisht i **caktohen në mënyrë dinamike një aplikacioni klient** kur incializion një lidhje.

2.14. Netstat

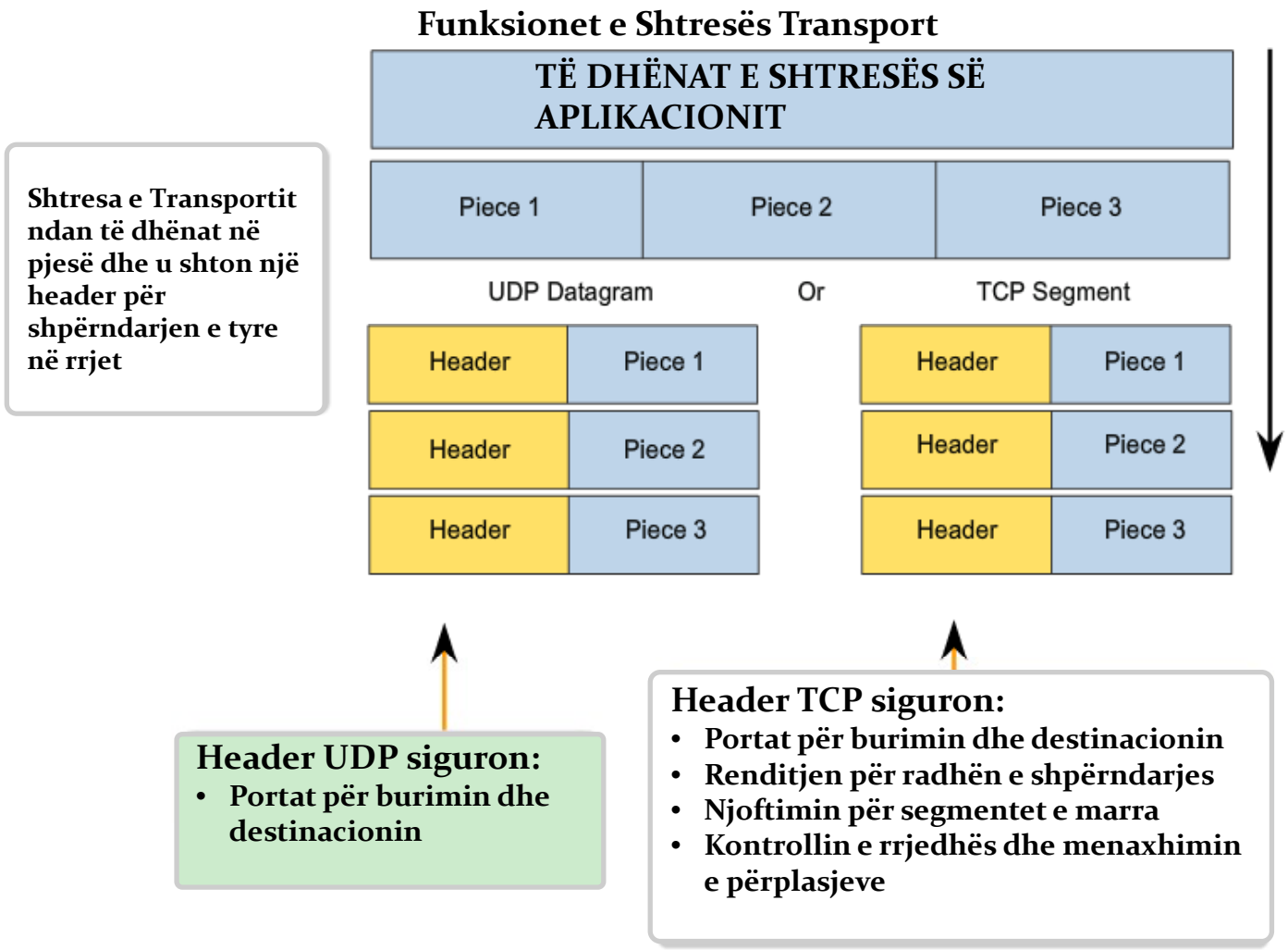
- **Lidhjet** e pashpjegueshme **TCP** mund të përbëjë një **kërcënim** të madh të **sigurisë**.
- Lidhjet e panevojëshme TCP mund të **konsumojnë burime** të vlefshme të sistemit dhe në këtë mënyrë të ngadalësojnë performancën e hostit.
- **Netstat** është një **shërbim i rrjetit** që mund të përdoret për të **verifikuar cilat lidhje aktive TCP janë hapur dhe po ekzekutohen** në një host në rrjet.
- Netstat **liston protokollet** në përdorim, **adresat** lokale dhe **numrat e portave**, adresat e jashtëme dhe numrat e portave, dhe **gjendjen** e lidhjes.

Netstat Output

```
C:\>netstat

Active Connections

Proto    Local Address          Foreign Address        State
TCP      kenpc:3126             192.168.0.2:netbios-ssn    ESTABLISHED
TCP      kenpc:3158             207.138.126.152:http       ESTABLISHED
TCP      kenpc:3159             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3160             207.138.126.169:http       ESTABLISHED
TCP      kenpc:3161             sc.msn.com:http            ESTABLISHED
TCP      kenpc:3166             www.cisco.com:http         ESTABLISHED

C:\>
```

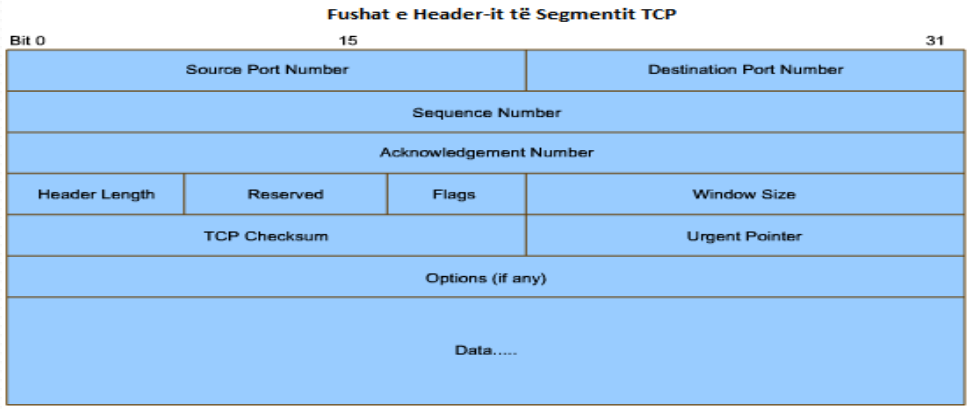
2.14. Segmentimi dhe Riasemblimi

TCP dhe **UDP** trajtojnë segmentimin në mënyra të ndryshme



3.1. Protokolli TCP – komunikimi i besueshëm

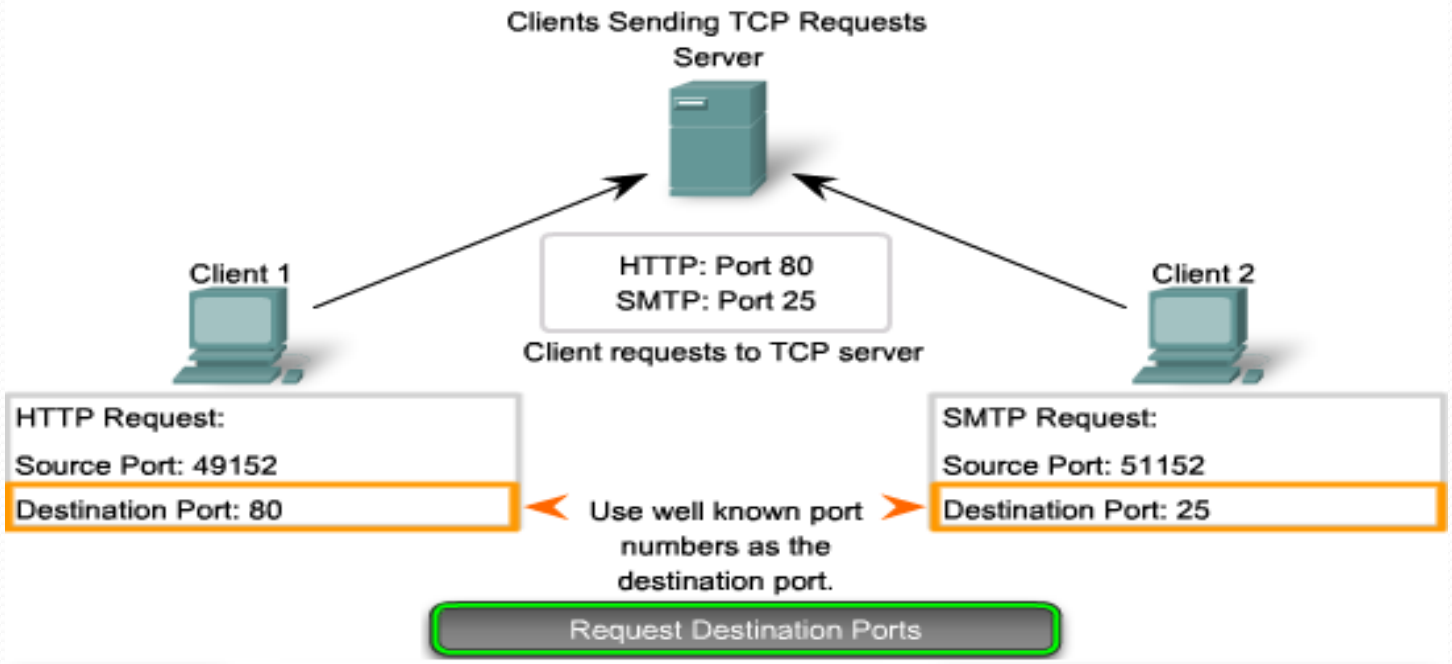
- **TCP** është **lidhje connection-oriented**, p.sh. kur dy hoste komunikojnë duke përdorur TCP, lidhja midis tyre duhet të vendoset përpara se të shkëmbehen të dhëna.
- Pasi krijohet sesioni, **destinacioni i dërgon njoftim** burimit për segmentet që merr.
- Nëse **burimi** nuk merr një njoftim brenda një kohe të paracaktuar, ai i **ridërgon të dhënat** tek destinacioni.
- Besueshmëria arrihet duke shtuar **fusha në segmentin TCP**, secila nga këto me një funksion specifik.



Fushat e header-it TCP sigurojnë një komunikim të besueshëm të dhënash dhe connection-oriented (të orjentuar nga lidhja)

3.2. Proçeset server TCP

- Është e zakonshme për **një server të ofrojë më shumë se një shërbim**, të tillë si web server dhe FTP server në të njëjtën kohë.
- Secili proçes aplikacion që ekzekutohet në server është i **konfiguruar të përdorë një portë**, ose by default ose manualisht nga administratori i sistemit.
- Një server **nuk mund të ketë dy shërbime** të caktuara **tek i njëjti numër porte** brenda të njëjtave shërbime të shtresës Transport.



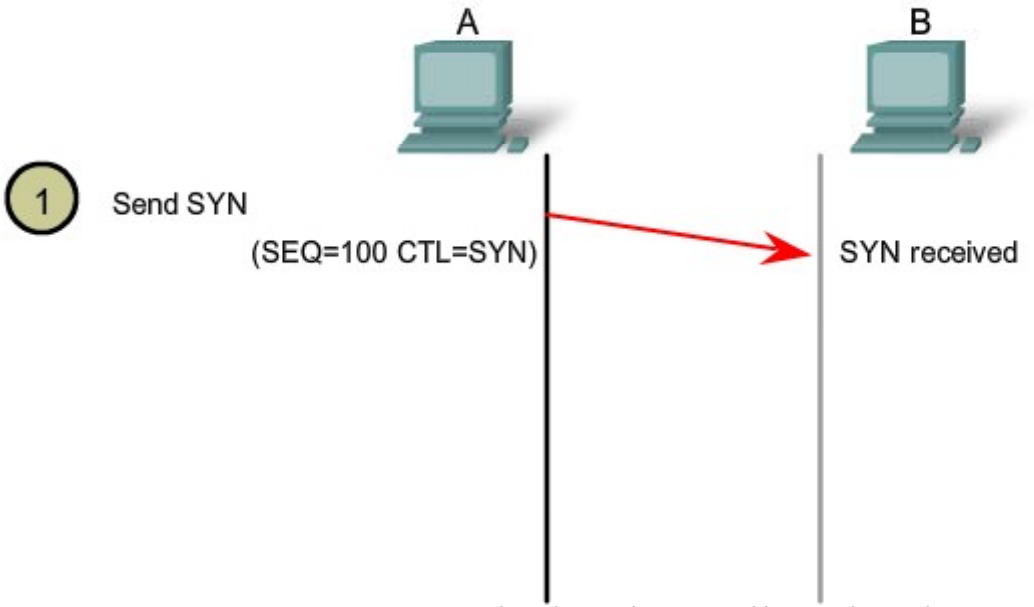
3.3. Vendosja dhe mbyllja e lidhjes

- **TCP** është **lidhje connection-oriented**, p.sh. kur dy hoste komunikojnë duke përdorur TCP, **lidhja midis tyre duhet të vendoset** përpara se të shkëmbehen të dhëna.
- Pasi komunikimi përfundon, **sesionet mbyllen** dhe lidhja përfundon.
- **Për të vendosur lidhjen**, hosti kryen një **takim me tre hapa** (three-way handshake). **Bitet e kontrollit** në header-in TCP tregojnë **progresin dhe statusin e lidhjes**.
- **Lidhja me 3-Hapa**:
  - Përcaktimi nëse **pajisa destinacion është prezente** në rrjet
  - Verifikimi nëse **pajisja destinacion ka një shërbim aktiv dhe pranon kërkesa** në numrin e portës destinacion që klienti iniciues planifikon të përdorë për sesionin
  - **Informimi i pajisjes destinacion** që klienti burim planifikon të krijojë një lidhje në atë numër porte

3.4. Krijimi i Lidhjes TCP me 3 hapa

- **Klienti iniciues** dërgon një **segment** që përmban një **vlerë sekuenciale iniciuese (SYN)**, e cila shërben si kërkesë tek serveri për të filluar një sesion komunikimi.

Krijimi dhe Përfundimi i Lidhjes TCP



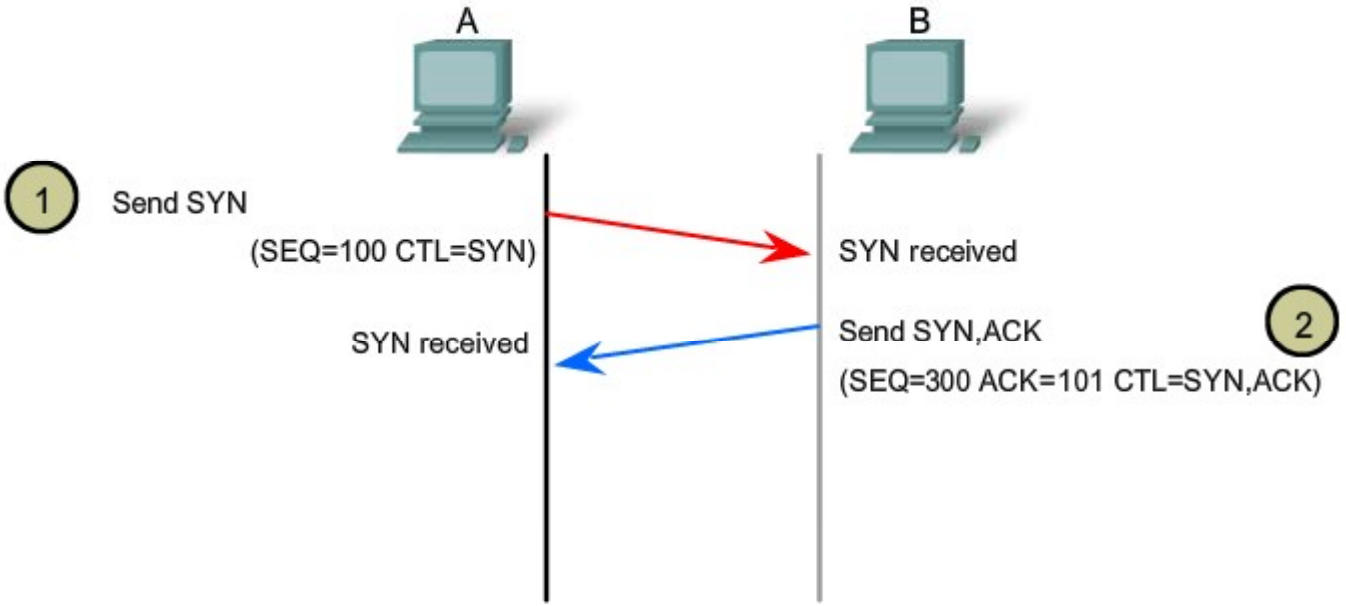
CTL = **Cilët bite kontrolli në header-in TCP kanë vlerën 1**

A i dërgon B kërkesën SYN

3.5. Krijimi i Lidhjes TCP me 3 hapa

- **Serveri përgjigjet** me një segment që përmban një **njoftim (ACK)** me vlerë të barabartë me sekuencën e marrë plus 1, plus vlerën e vet të **sinkronizimit (SYN)**.

Krijimi dhe Përfundimi i Lidhjes TCP



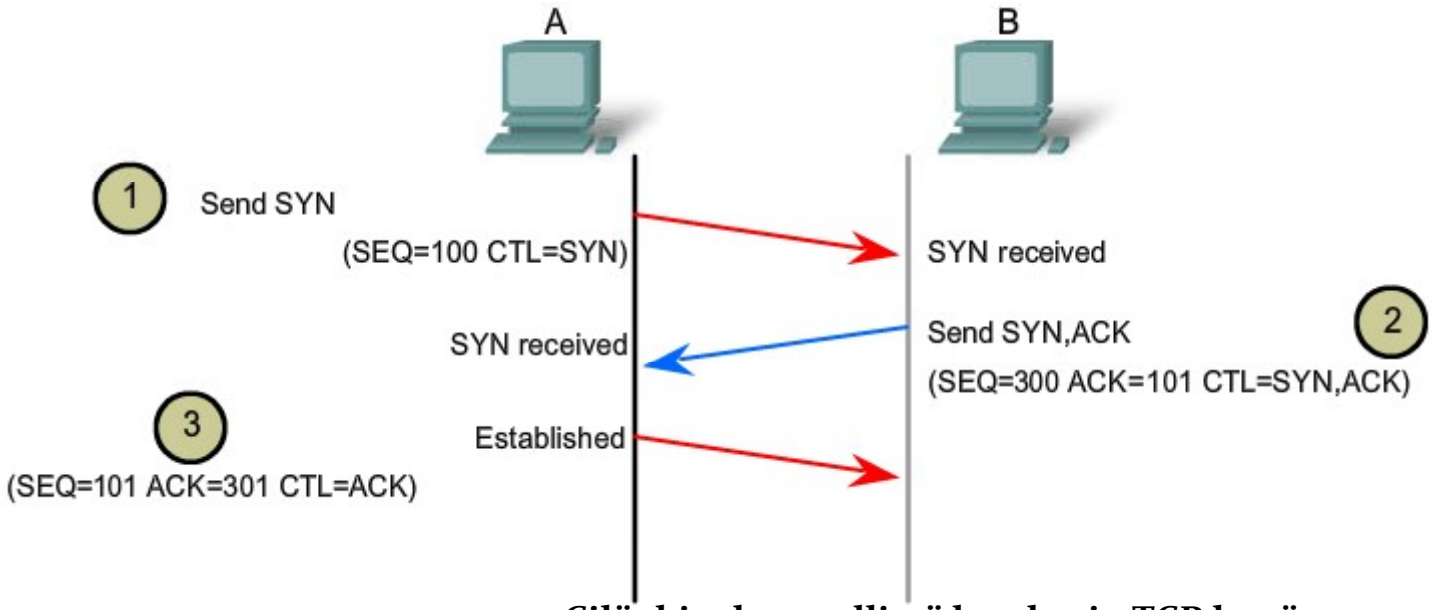
CTL = **Cilët bite kontrolli në header-in TCP kanë vlerën 1**

B dërgon përgjigjen ACK dhe kërkesën SYN tek A

3.5. Krijimi i Lidhjes TCP me 3 hapa

- **Klienti iniciues** përgjigjet me një **njoftim (ACK)** me vlerë të barabartë me vlerën e sekuencës që merr plus një.

Krijimi dhe Përfundimi i Lidhjes TCP

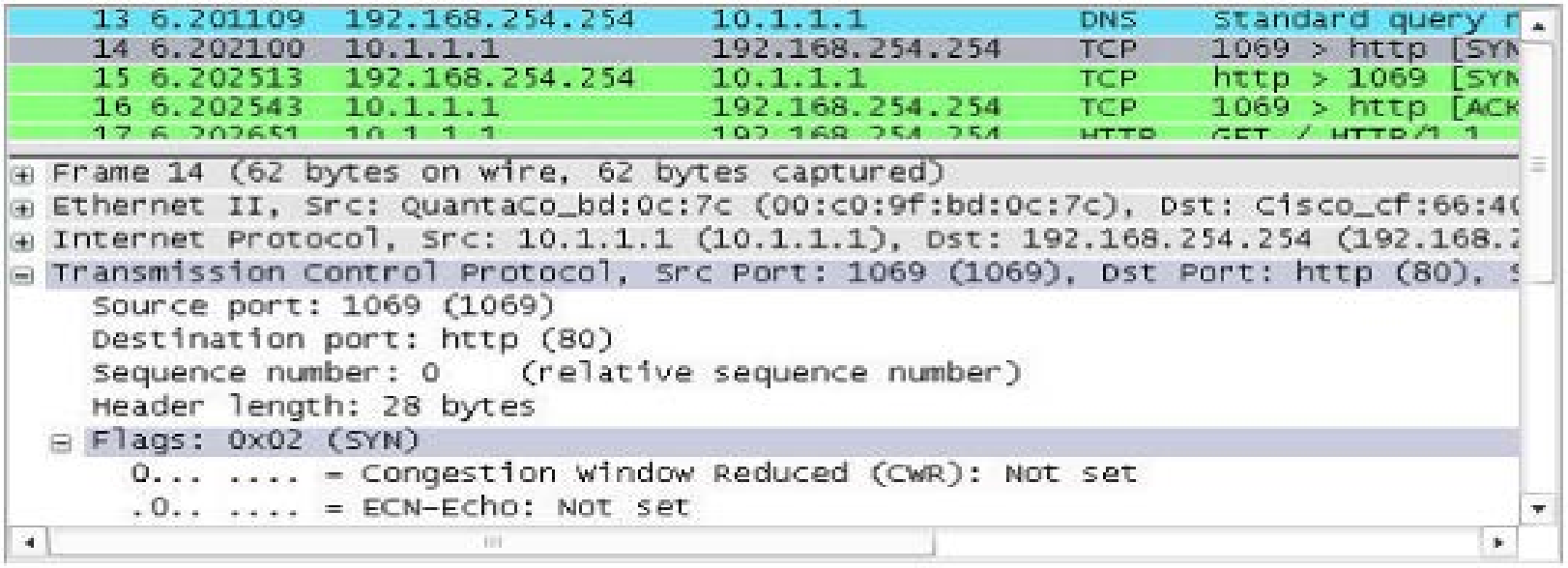


CTL = **Cilët bite kontrolli në header-in TCP kanë vlerën 1**

A dërgon përgjigjen ACK tek B

3.6. Krijimi i Lidhjes TCP
Eksplorimi me Wireshark

TCP 3-way Handshake (SYN)



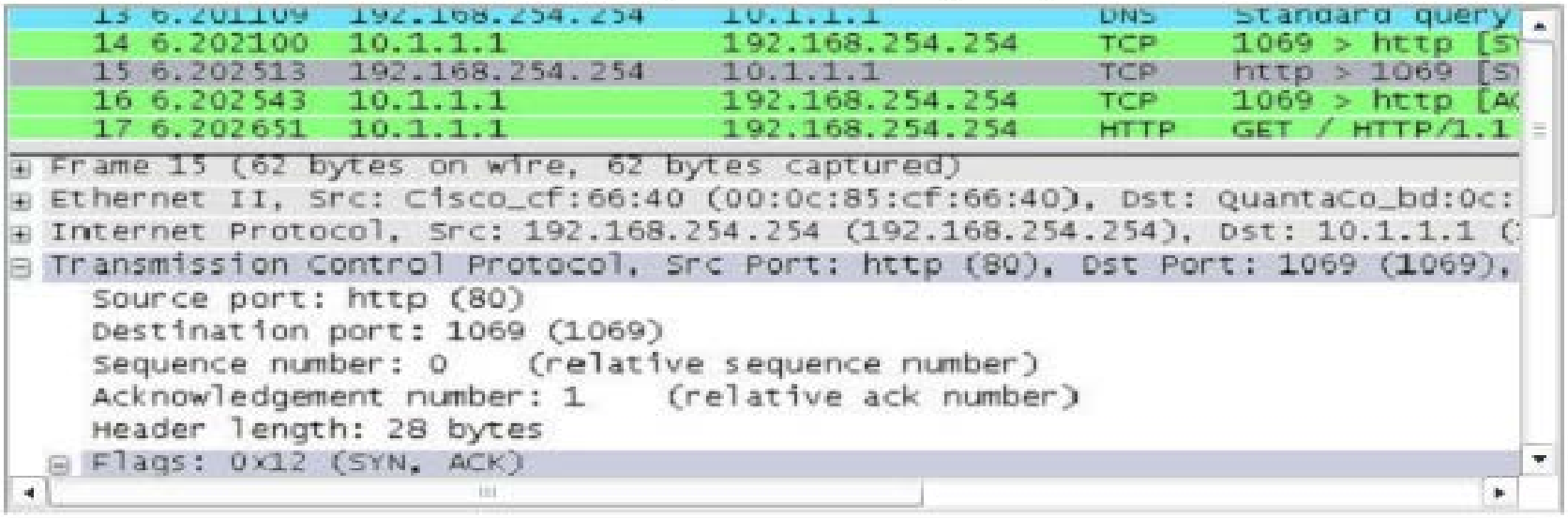
Protocol Analyzer shows initial client request for session in frame 14

TCP segment in this frame shows:

- SYN flag set to validate an initial Sequence number
- Randomized sequence number valid (relative value is 0)
- Random source port 1069
- Well known destination port is 80 (HTTP port) indicates web server (httpd)

3.7. Krijimi i Lidhjes TCP
Eksplorimi me Wireshark

TCP 3-way Handshake (SYN, ACK)

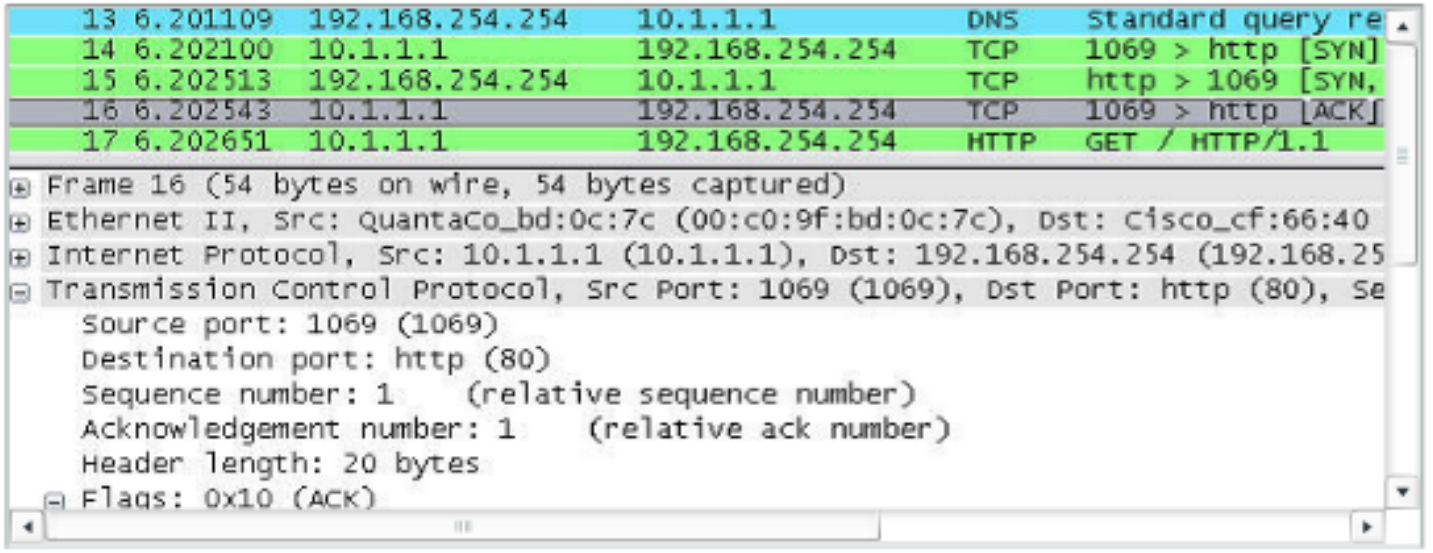


A protocol analyzer shows server response in frame 15

- ACK flag set to indicate a valid Acknowledgement number
- Acknowledgement number response to initial sequence number as relative value of 1
- SYN flag set to indicate the Initial sequence number for the server to client session
- Destination port number of 1069 to corresponding to the clients source port
- Source port number of 80 (HTTP) indicating the web server service (httpd)

3.8. Krijimi i Lidhjes TCP
Eksplorimi me Wireshark

TCP 3-way Handshake (ACK)

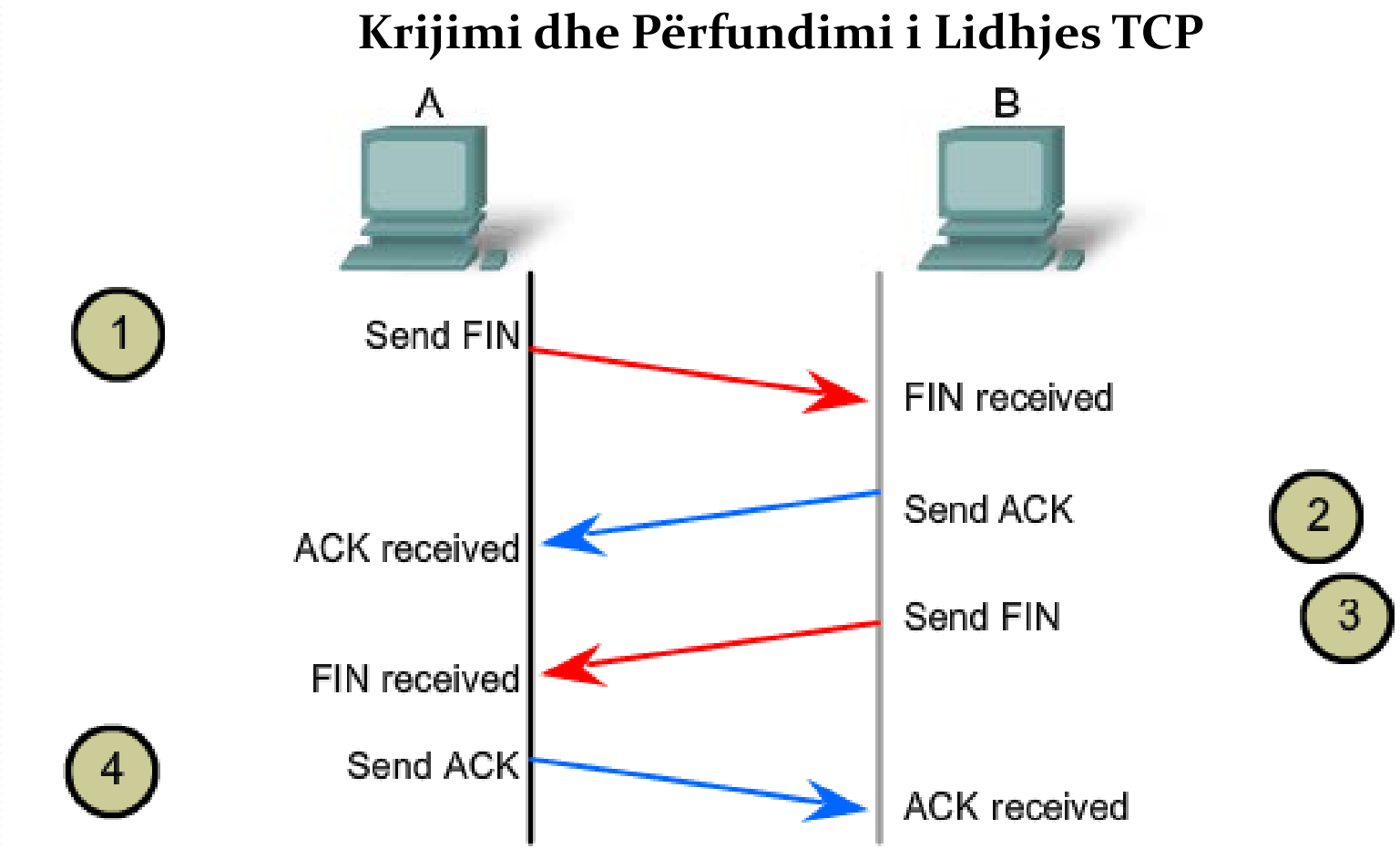


Protocol Analyzer shows client response to session in frame 16

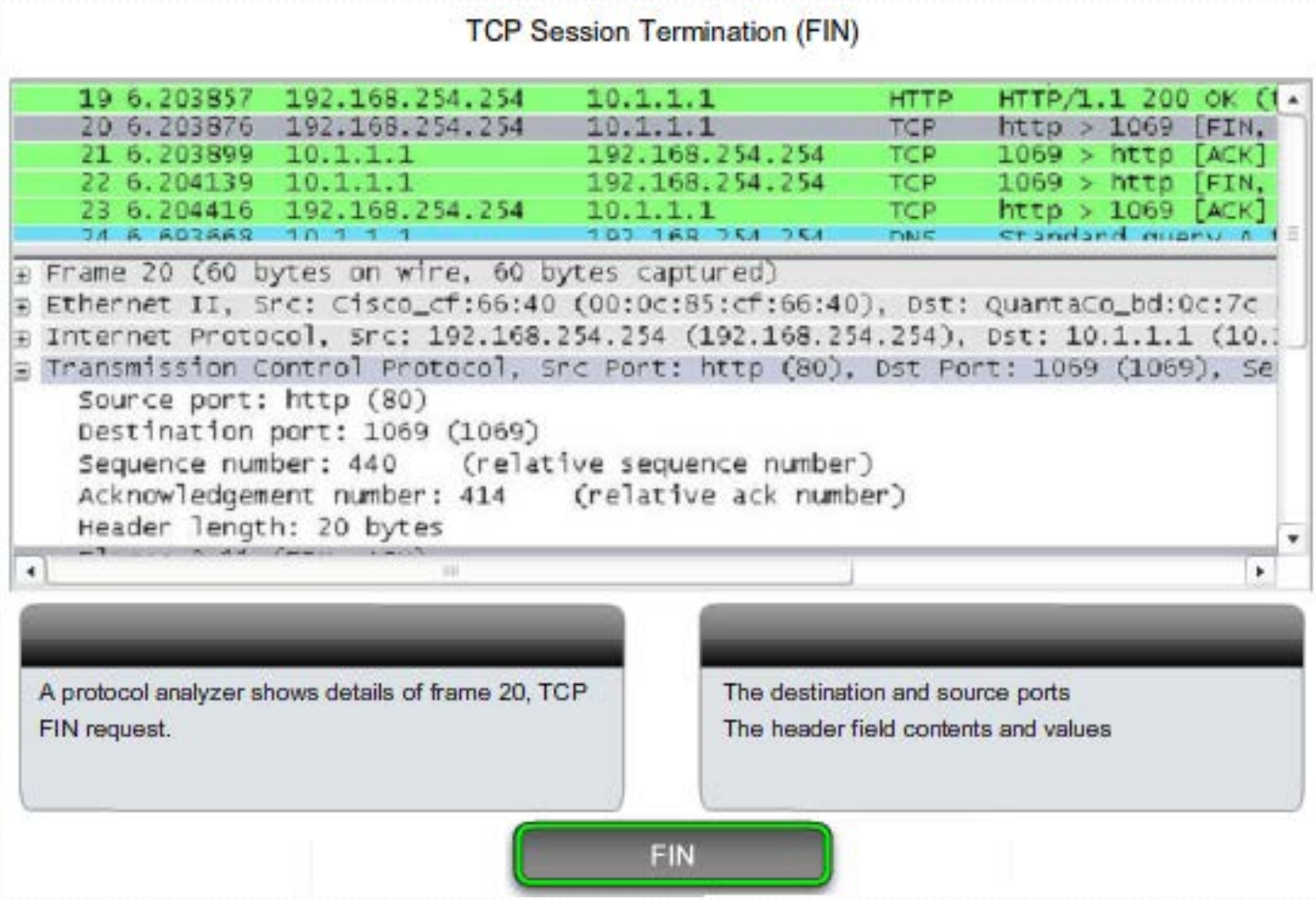
The TCP segment in this frame shows:

- ACK flag set to indicate a valid Acknowledgement number
- Acknowledgement number response to initial sequence number as relative value of 1
- Source port number of 1069 to corresponding
- Destination port number of 80 (HTTP) indicating the web server service (httpd)

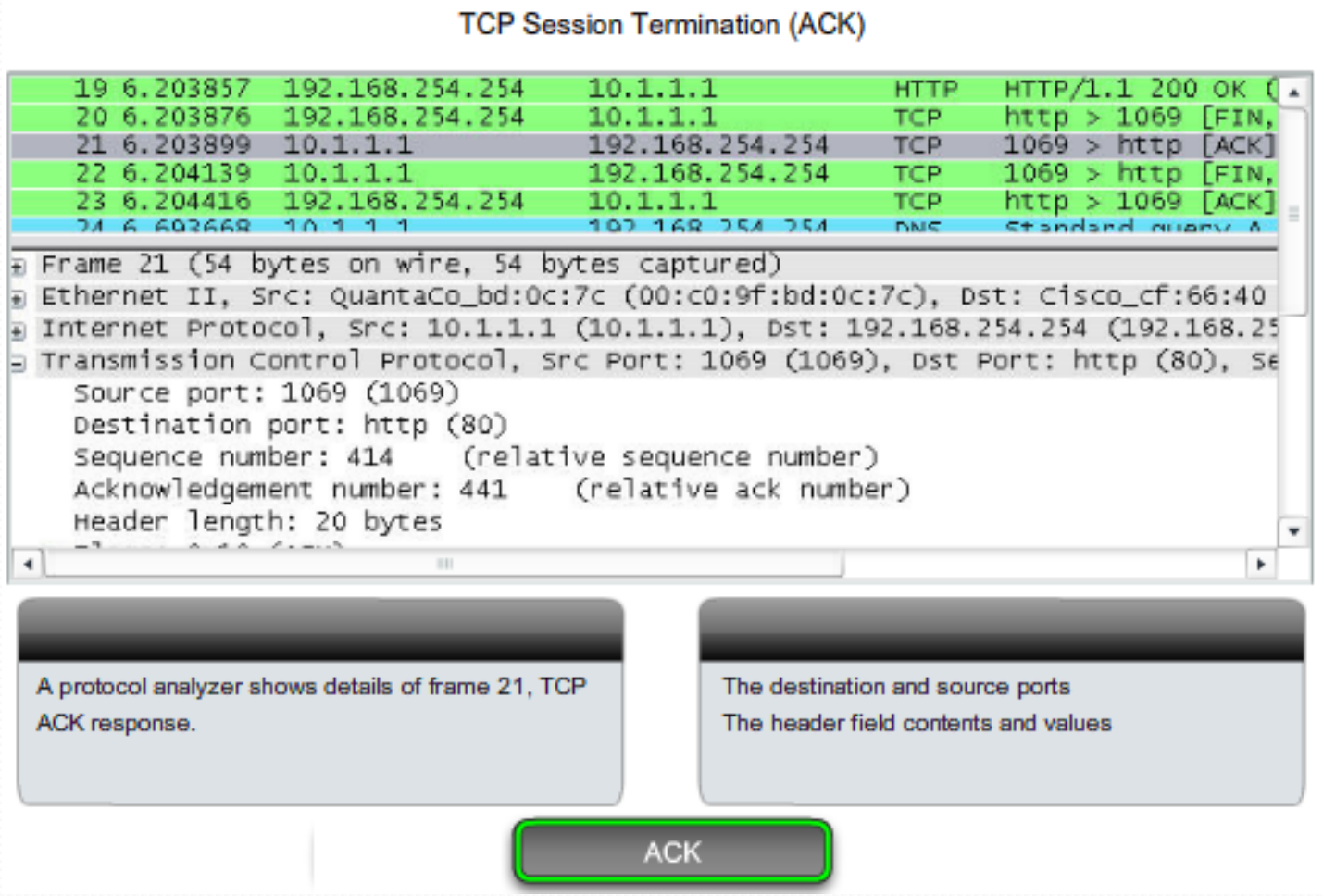
3.9. Përfundimi i Lidhjes TCP me 4 hapa



3.10. Mbyllja e sesionit TCP
Eksplorimi me Wireshark



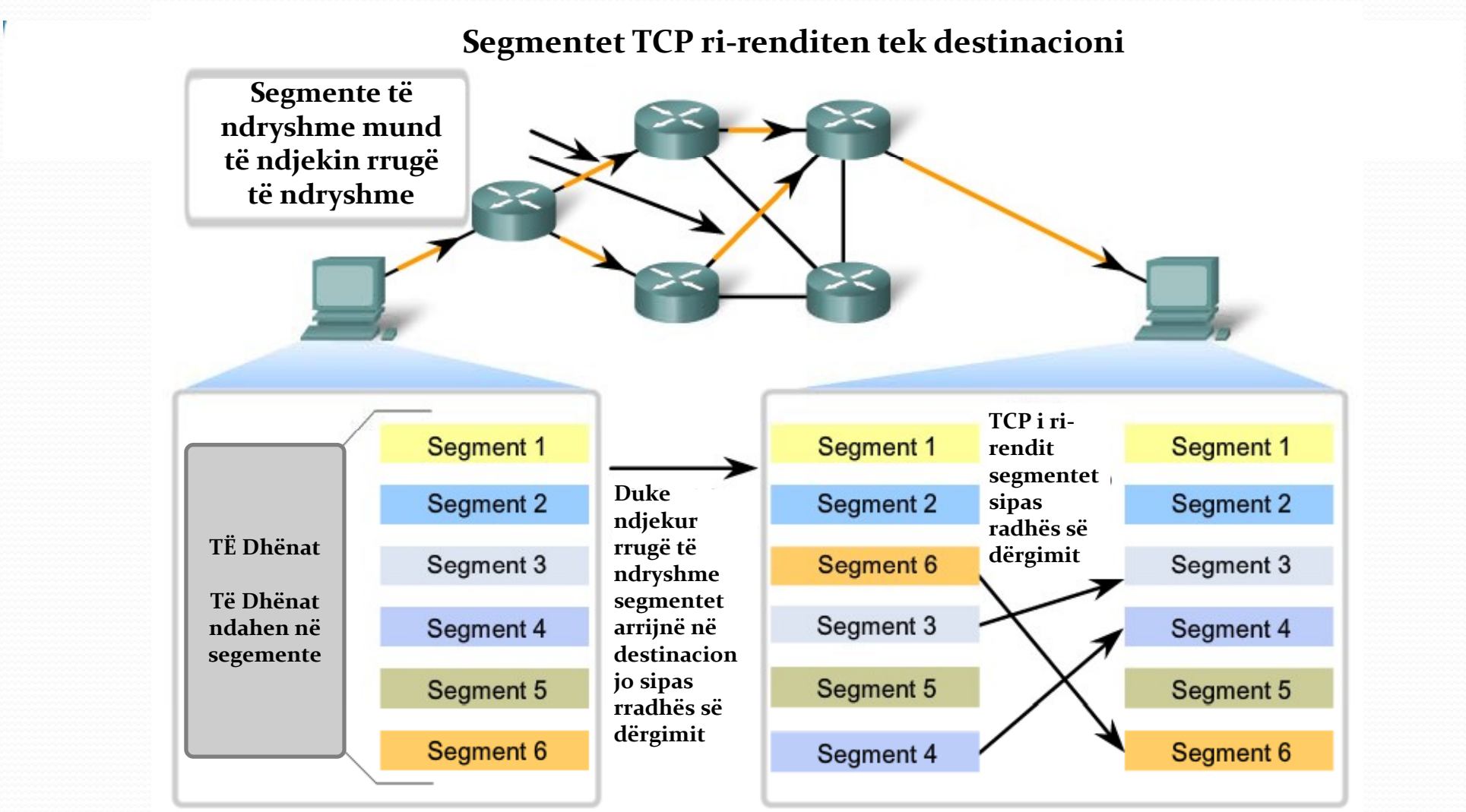
3.11. Mbyllja e sesionit TCP
Eksplorimi me Wireshark



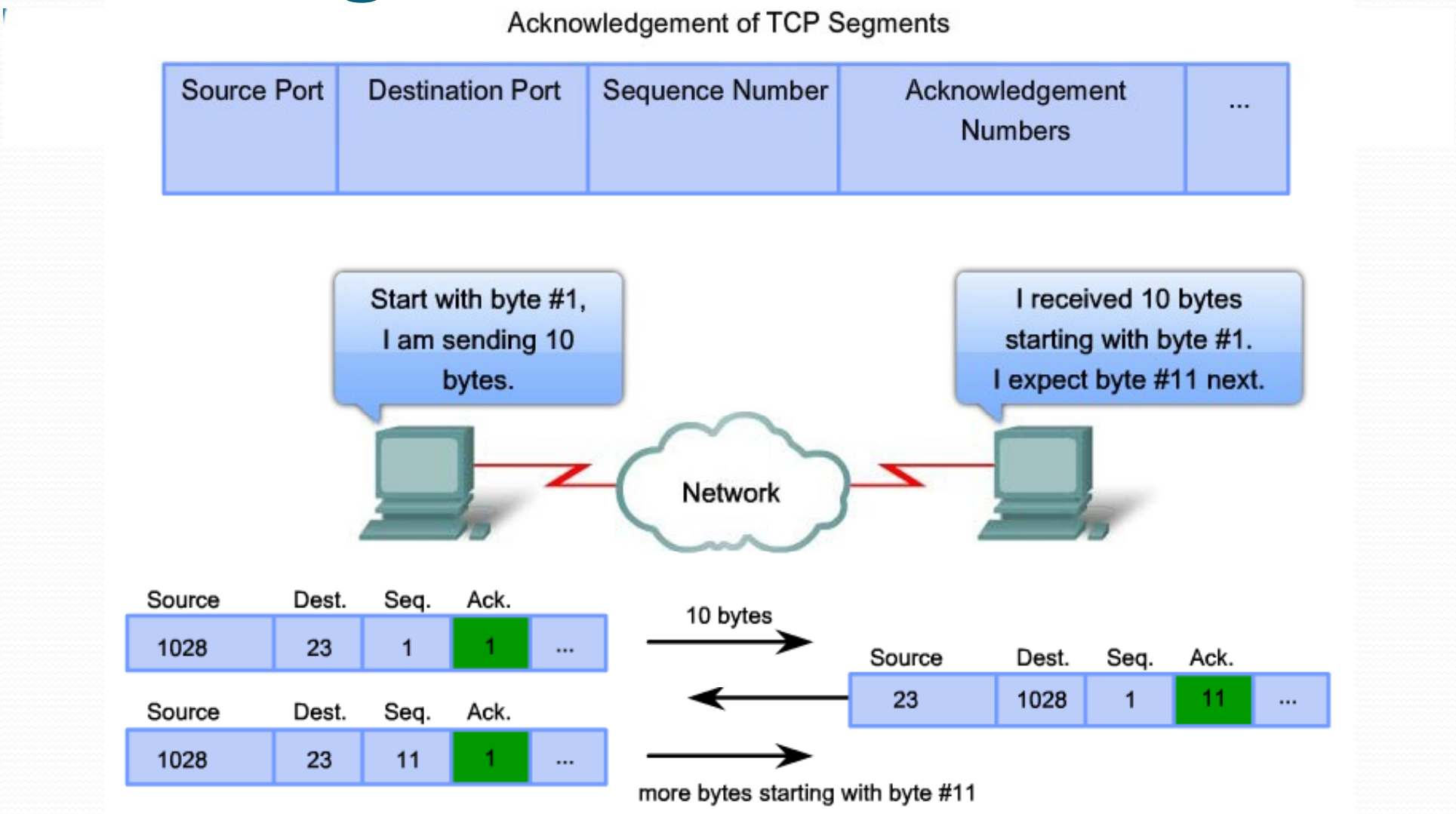
4.1. Riasemblimi i segmenteve

- Gjatë krijimit të lidhjes vendoset një **numër fillestar sekuencial (ISN)**
- Kjo ISN përfaqëson **vlerën fillestare të byteve që do të transmetohen në këtë sesion** për tek aplikacioni pritës. Gjatë transmetimit të të dhënave në sesion, **ISN rritet me numrin e byteve të transmetuara**.
- Ky gjurmin i byte-ve i mundëson **çdo segmenti të identifikohet dhe njoftohet në mënyrë unike**. Segementet e munguara mund të identifikohen.

4.2. Riasemblimi i segmenteve



4.3. Konfirmimi i marrjes së segmenteve



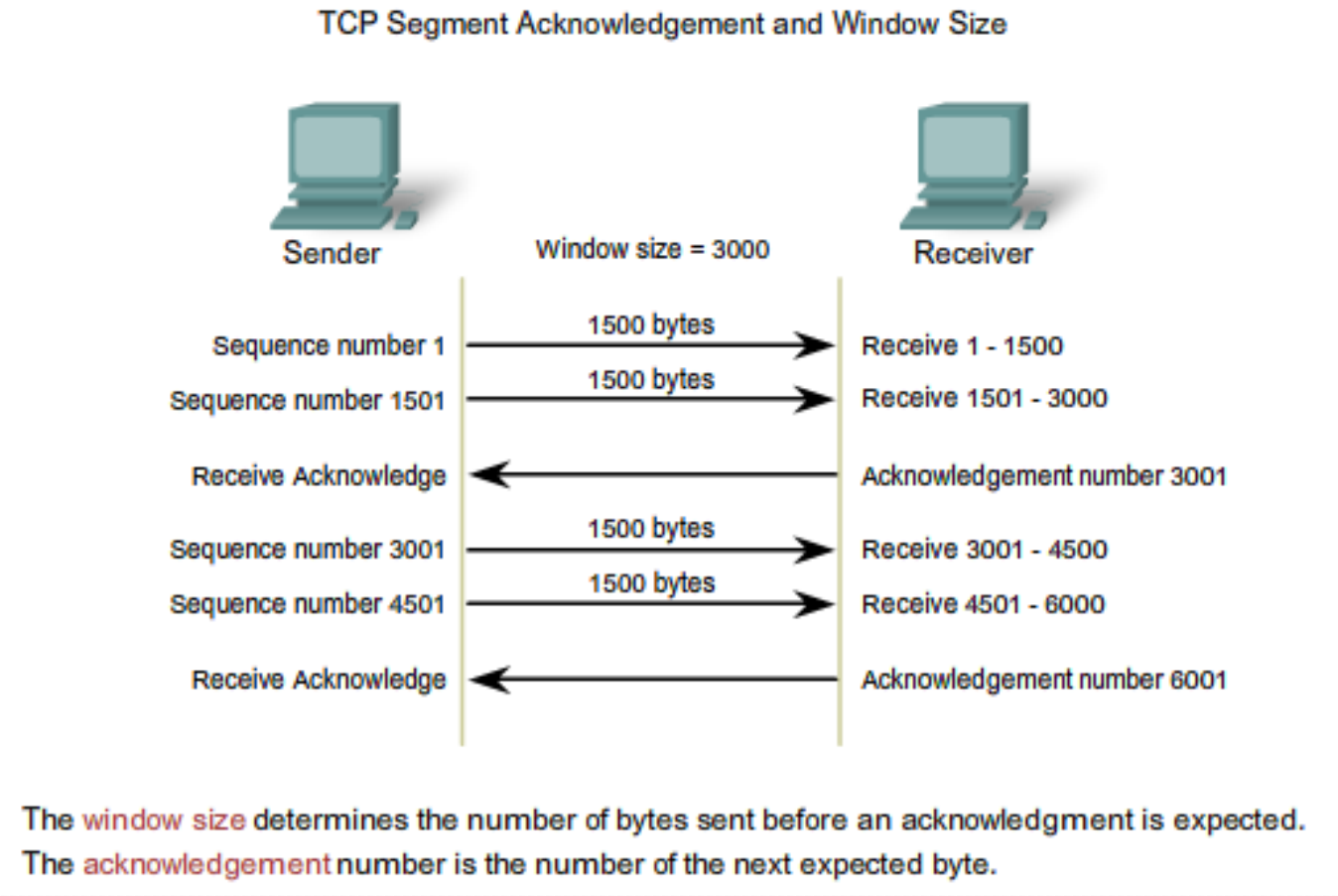
4.4. Trajtimi i humbjeve

- **Menaxhimi i humbjes** me anë të mekanizmave të **ritransmetimit**.
- **Numri i sekuencës** tregon **numrin relativ të byte-ve** që janë **transmetuar** në këtë sesion.
- **TCP përdor numrin e njoftimit** në segmentet e dërguara mbrapsht tek burimi **për të shënuar byte-t e rradhës .**
- Ky quhet **njoftim i pritshmërive.**
- Hosti **dërgues** njofton vetëm segmentet që kanë një **sekuencë të vazhduar**.
- **Njoftim selektiv** – destinacioni njofton edhe bytet që nuk janë në vazhdimësi

4.5. ACK me Dritare

- **Madhësia e të dhënave** që një burim mund të transmetojë **përpara se të marrë një njoftim** quhet madhësia a dritares (**Window size**)
- Madhësia fillestare e **dritares përcaktohet gjatë hapjes së sesionit** nëpërmjet takimit me tre rrugë.
- Madhësia e dritares është një fushë në header-in TCP që **mundëson menaxhimin e humbjes** së të dhënave dhe **kontrollin e rrjedhës**
- Kur TCP në hostin burim **nuk ka marrë një njoftim** pas kohës së lejuar, ai **do të shikojë numrin e njoftimit të fundit** që ka marrë dhe do të **ri-transmetojë** të dhënat duke filluar **që aty**

4.6. ACK me Dritare

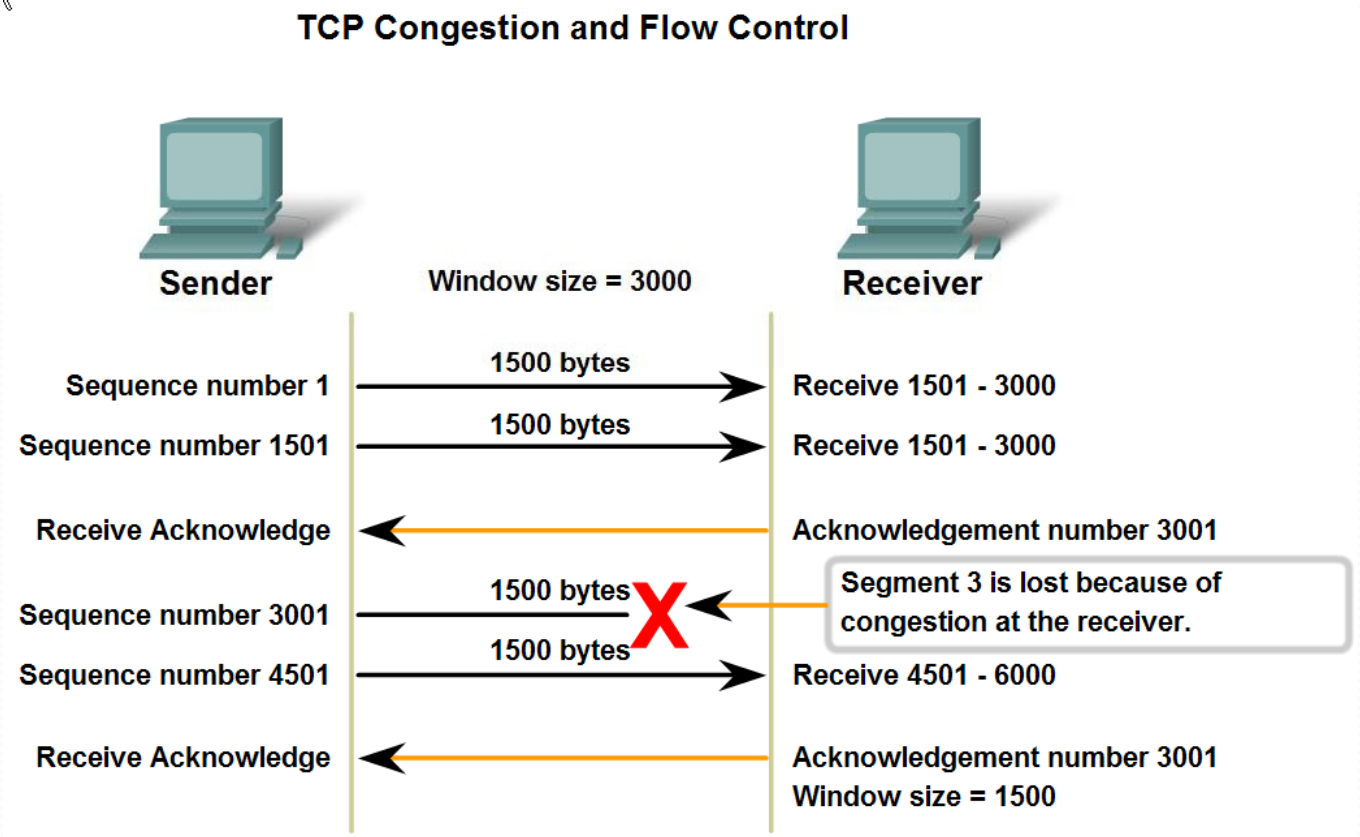


The window size determines the number of bytes sent before an acknowledgment is expected.
The acknowledgement number is the number of the next expected byte.

4.7. Kontrolli i rrjedhës

- Kur ka **përplasje** në rrjet ose **burimet** e hostit marrës **janë të zëna**, **vonesa në dërgimin e njoftimit** mund të rritet
- Me rritjen e vonesës, **ulet shkalla e efektivitetit** të transmetimit të sesionit
- Një mënyrë tjetër për të kontrolluar rrjedhën është përdorimi i **madhësisë dinamike të dritares**
- **Hosti marrës** i TCP **i dërgon vlerën e madhësisë së dritares** hostit dërgues të TCP **për ti treguar numrin e byte që ai mund të presë** si pjesë e këtij sesioni

4.8. Reduktimi i madhësisë së dritares

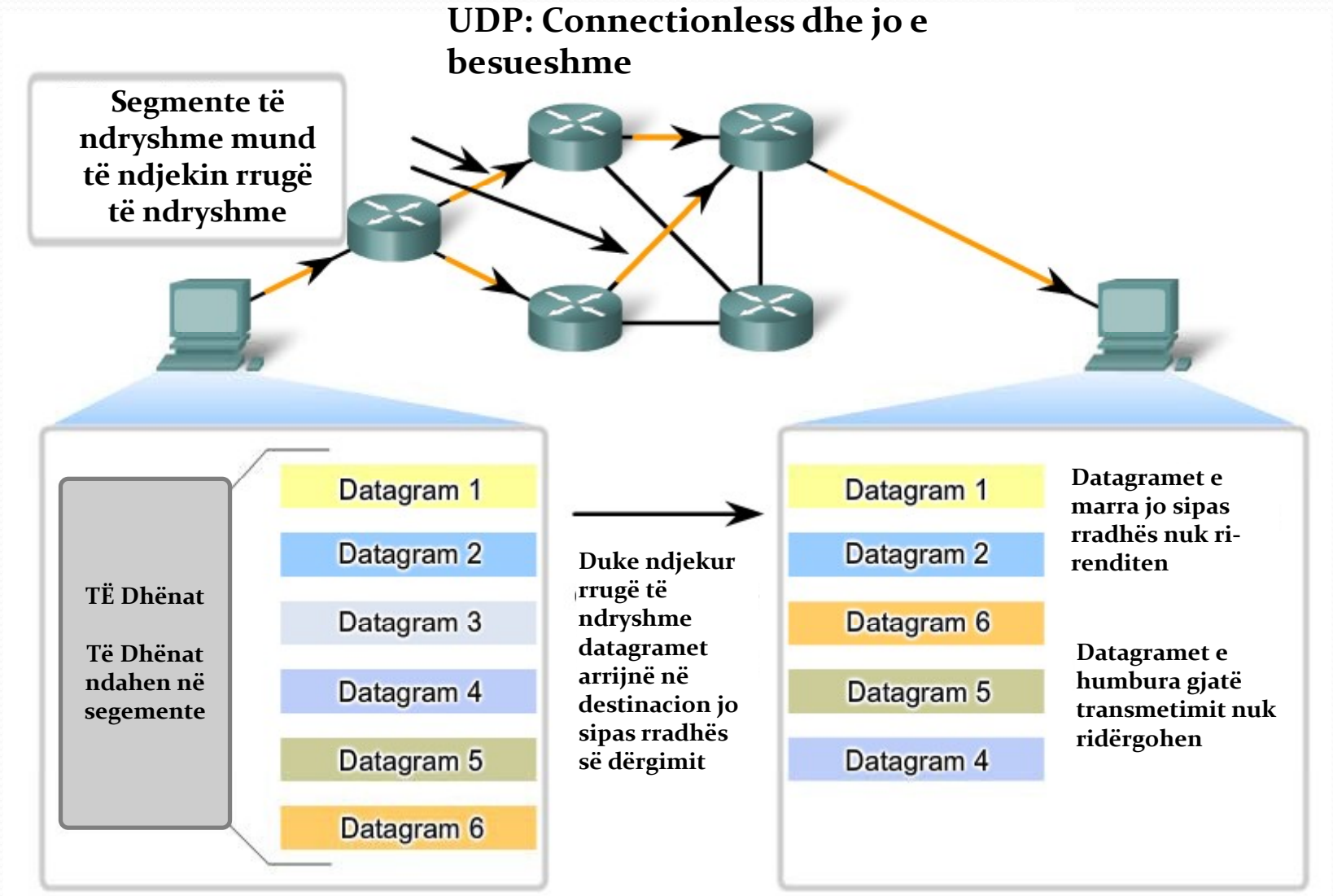


N.q.s. segmentet humbasin për shkak të përplasjeve, Marrësi do të njoftojë për marrjen e segmentit të fundit sekuencial dhe do të përgjigjet me një madhësi dritare të reduktuar

5.1. User Datagram Protocol (UDP)

- **UDP** është një protokoll i thjeshtë, **connectionless** që siguron **mbingarkesë të vogël** në transmetimin e të dhënave. Thuhet që UDP është **transaction-based**, d.m.th. dërgon direkt të dhënat
- Pjesët e komunikimit në UDP quhen **datagrama** dhe dërgohen nga UDP **bazuar në përpjekjen më të mirë** (best-effort)
- **Aplikacione** që përdorin UDP janë:
  - **Domain Name System (DNS)**
  - **Video Streaming**
  - **Voice over IP (VoIP)**
  - **Simple Network Management Protocol (SNMP)**
  - **Dynamic Host Configuration Protocol (DHCP)**
  - **Routing Information Protocol (RIP)**
  - **Trivial File Transfer Protocol (TFTP)**

5.2. UDP


UDP: Connectionless dhe jo e besueshme
Segmente të ndryshme mund të ndjekin rrugë të ndryshme
TË Dhënat
Të Dhënat ndahen në segemente
Datagram 1
Datagram 2
Datagram 3
Datagram 4
Datagram 5
Datagram 6
Duke ndjekur rrugë të ndryshme datagramet arrijnë në destinacion jo sipas rradhës së dërgimit
Datagram 1
Datagram 2
Datagram 6
Datagram 5
Datagram 4
Datagramet e marra jo sipas rradhës nuk ri-renditen
Datagramet e humbura gjatë transmetimit nuk ridërgohen

5.3. Proçeset Client/Server në UDP

- Si në rastin e TCP, **komunikimi client/server fillohet nga aplikacioni klient** që kërkon të dhëna nga një proçes server
- **Procesi klient i UDP** zgjedh në mënyrë rastësore një **numër porte** nga segmenti i portave **dinamike** dhe e përdor atë si portë burim për komunikimin
- **Porta destinacion** zakonisht do të jetë një portë e **rregjistruar ose e mirënjohur** e cila i është caktuar **proçesit server**

5.4. Proçeset Client në UDP

- Për shkak se **UDP nuk ka sesion për të krijuar**, sapo **të dhënat të jenë gati** për tu dërguar dhe **portat janë identifikua**, UDP mund ta **përcjellë datagram-ën në shtresën e Network** për tu adresuar dhe dërguar në rrjet.