Transportná vrstva

Prečo potrebujeme transportnú vrstvu?

K transportnej vrstve OSI sa z vyšších vrstiev dostávajú dáta, ktoré:

- vytvoril istý aplikačný program alebo služba (OSI Layer7): Napríklad webový prehliadač.
- majú správny spoločný formát (OSI Layer6): Napríklad webová stránka – formát HTML
- sú určené správnemu procesu a dialógu u príjemcu (OSI Layer5): Požiadavka na server.

Dáta v tomto tvare však nie sú na prenos sieťou vhodne pripravené. Zatiaľ nie sú segmentované na menšie úseky – sú v bloku, v akom ich vytvorila aplikácia, plus informácie z L7-L5 vrstvy.

Ak sa budú sieťou prenášať ako oddelené segmenty, môže dôjsť k ich pre usporiadaniu alebo strate. Riešiť tieto nedostatky je práve úlohou transportnej vrstvy.

- Transportná vrstva je na rozhraní medzi aplikačnými dátami a sieťovou infraštruktúrou
- Zabezpečuje prenos dát medzi konkrétnymi komunikujúcimi aplikáciami na koncových uzloch
- Dáta prenáša ako rad úsekov, tzv. **segmentov**
- Tieto segmenty môžu niesť poradové čísla a iné informácie pre ich správne doručenie a opätovné poskladanie u príjemcu
- Transportná vrstva zodpovedá za doručenie dát cez sieť v takom tvare, v akom boli odoslané.

Úlohy transportnej vrstvy:

Základné:

- Oddelenie konverzácií
- Segmentácia dát
- Spätná rekonštrukcia pôvodných dát zo segmentov
- Identifikácia komunikujúcich aplikácií

Rozširujúce(Ak to aplikácia potrebuje):

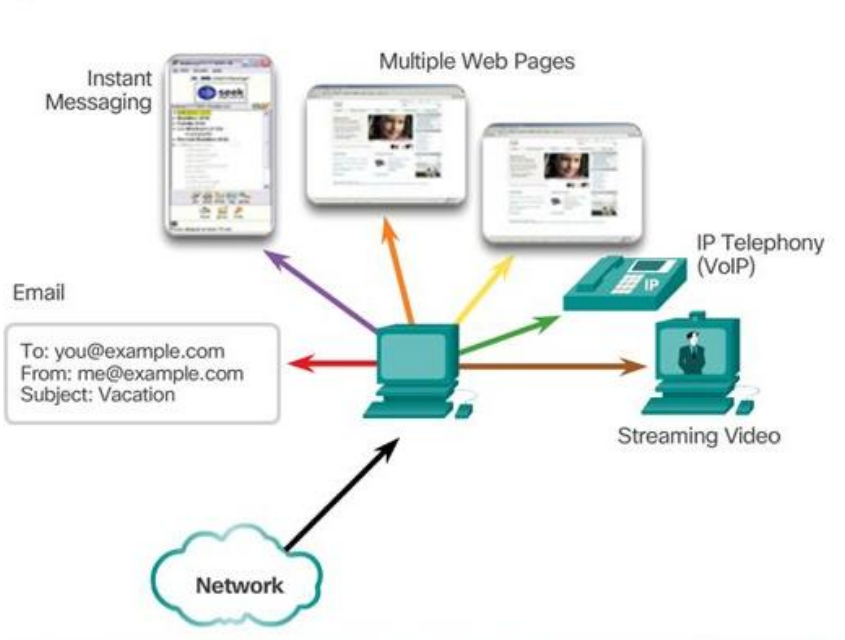
- Spojovanosť
- Spoľahlivosť
- Usporiadanosť
- Riadenie toku dát (Flow control)

1. Oddelenie konverzácií

Umožňuje, aby sa komunikujúce aplikácie dokázali navzájom správne adresovať (napr. webový server a klient)

Umožňuje oddeliť od seba súbežné konverzácie medzi tým istým klientom a serverom.

Typický príklad Facebook:

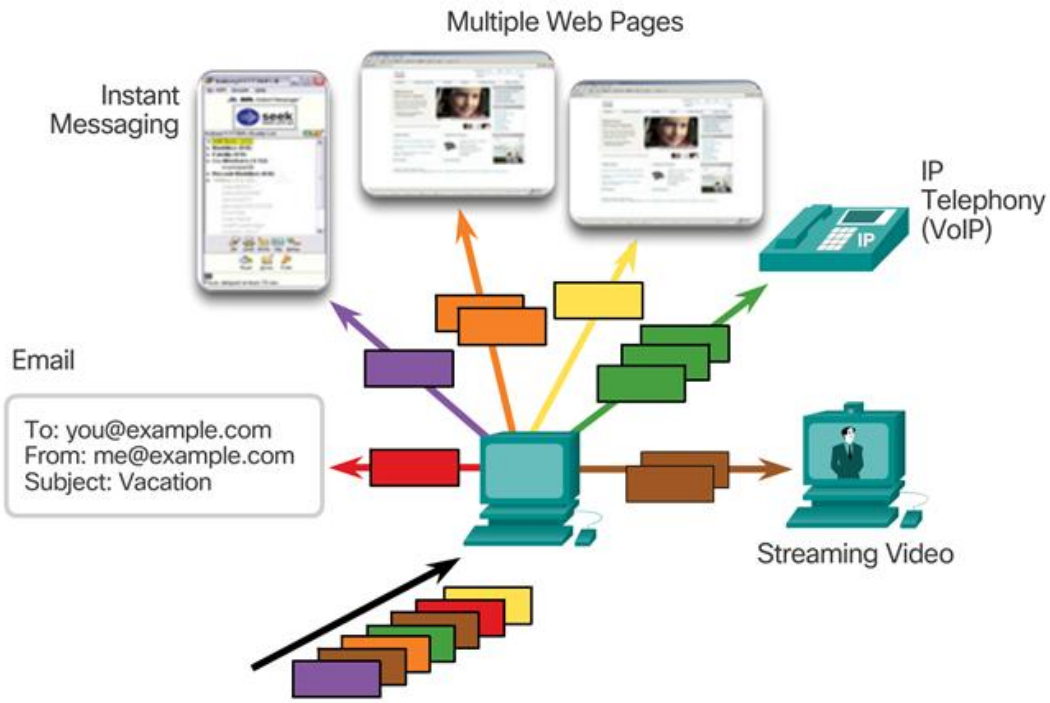


2. Segmentácia a spätná rekonštrukcia dát

Segmentácia dát (Segment data) a spätná rekonštrukcia pôvodných dát zo segmentov (Reassemble segments)

Transportná vrstva pridá každému bloku hlavičku, ktorá:

- identifikuje daný segment
- umožňuje rôznym transportným protokolom vykonávať rôzne funkcie, ktoré sú potrebné pre manažovanie danej komunikácie pre danú aplikáciu
- umožňuje multiplexovanie mnohých rôznych aplikácií od veľkého počtu používateľov na jednej sieti
- tým že sa dáta rozdelia na menšie bloky, môžu viaceré aplikácie súbežne odosielať a prijímať dáta
- uľahčuje prípadné riešenie problémov s poškodením prenášaných dát



S tým súvisí aj:

3. Identifikácia komunikujúcich aplikácií

Zabezpečí, že aj keď na jednom zariadení beží niekoľko aplikácií, všetky dostanú tie dáta, ktoré sú pre ne určené.

4. **Spoľahlivosť**
   Proces, ktorý má zaistiť, že dáta budú prijaté presne v tom tvare, v akom boli odoslané.

   Spoľahlivý prenos (reliable)

   - garantuje, že prijaté dáta budú všetky
   - každý prenos dát musí byť potvrdený príjemcom, ak dáta neboli potvrdené, odosielateľ opakuje vysielanie kontrolujú sa chyby prenesených dát.

   Nespoľahlivý prenos (unreliable)

- negarantuje, že prijaté dáta budú všetky

5. **Usporiadanosť**
- Rekonštrukcia segmentov v pôvodnom/správnom poradí

- Rekonštrukcia segmentov v takom poradí ako prídu

6. **Riadenie toku dát**
   Schopnosť riadiť rýchlosť odosielania dát, aby sa predišlo zahlteniu prenosovej cesty a následným stratám segmentov.
- S riadením toku dát (flow control)
- Bez riadenia toku dát (no flow control)

Rozdielne nároky aplikácií na doručovanie správ

Aplikácie majú podľa svojho typu rôzne nároky na spôsob prenosu ich dát sieťou.

V TCP/IP architektúre sú najčastejšie používané transportné protokoly:

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

TCP	UDP
• Spojovo orientovaný ○ Vytvára obojsmerné spojenie	• Nespojovaný
• Spoľahlivý ○ Potvrdzovaný	• Nespoľahlivý ○ Nepotvrdzovaný
• Riadi tok dát	• Bez riadenia toku dát
• Bajtovo orientovaný ○ dáta sú transparentne prenášané ako kontinuálny tok dát	• Datagramovo orientovaný
• Veľkosť hlavičky: 20B bez prídavných informácií	• Veľkosť hlavičky: 8B

Ktorý transportný protokol pre ktorú aplikáciu?

TCP je vhodný pre aplikácie:

- v ktorých segmenty musia doraziť v presne stanovenom poradí, aby mohli byť úspešne spracované
- v ktorých všetky dáta musia doraziť do cieľa, aby boli použiteľné pre danú aplikáciu

napr. databázy, web prehliadač, emailový klient, ...

UDP je vhodný pre aplikácie:

- ktoré vedia tolerovať straty, ale pre ktoré je neprijateľné veľké oneskorenie
- napr. live audio/video streaming, Voice over IP, ...

Služby TCP

- TCP poskytuje spoľahlivú, spojovo orientovanú službu doručovania tokov dát (streams) s riadením toku (flow control)
- TCP je podstatne komplexnejší a zložitejší ako UDP
- 20 B réžia v hlavičke
- TCP je stavový protokol (statefull)
- Udržiava si záznam o stave relácie tým, že si zapamätá, ktorú informáciu už poslalo a ktorú informáciu už potvrdilo (že prišla)

Otvorenie TCP spojenia

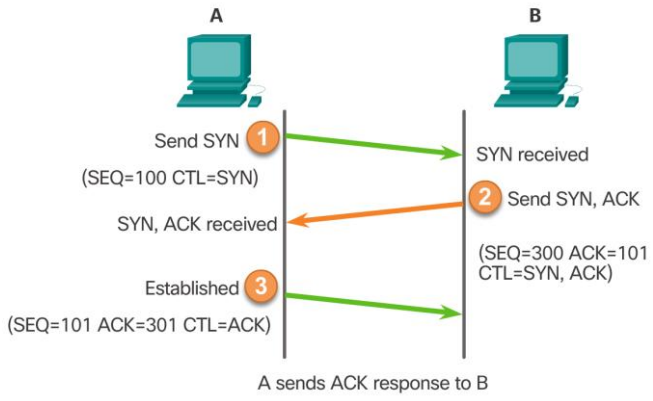
Pred výmenou dát v TCP je nutné zostaviť spojenie. Zostavením spojenia sa komunikujúce strany navzájom dohodnú na poradových číslach, počnúc ktorými budú číslovať svoje segmenty.

Až po tejto sekvencii môže začať prenos užitočných dát

Využívajú sa na to 2 príznaky v TCP hlavičke:
SYN (synchronization) a **ACK (acknowledgement)**

TCP spojenie je vytvorené v 3 krokoch (tzv. **3-way-handshake**):

- Klient iniciuje vytvorenie spojenia – požiada o client-to-server komunikačnú reláciu (session) so serverom.
- Server potvrdí túto reláciu a požiada klienta o server-to-client komunikačnú reláciu.
- Klient potvrdí server-to-client komunikačnú reláciu.



Služby UDP

- UDP poskytuje nespojovanú nespoľahlivú datagramovú službu.
- UDP v skutočnosti segmentáciu ani nerieši – dáta musí segmentovať odosielajúca aplikácia.
- UDP len priloží k dátam prijatým z aplikačnej vrstvy vlastnú hlavičku, ktorá identifikuje zdrojový a cieľový proces, popisuje veľkosť datagramu a obsahuje kontrolný súčet, a segment odošle
- U príjemcu sa prijatý segment po overení kontrolného súčtu odovzdá cieľovému procesu (podľa cieľového čísla portu)
- UDP neoveruje, či segmenty dorazili všetky a v pôvodnom poradí
- UDP nerieši opakovaný prenos chýbajúcich alebo poškodených segmentov

Najčastejšie služby využívajúce UDP:

- DHCP (porty 67 a 68)
- DNS (port 53)
- Voice over IP (porty 5060 a dynamické porty), video, IPTV
- Aplikácie, kde hrozí, že udržiavanie veľkého počtu spojení spôsobí **veľkú spotrebu systémových prostriedkov** na komunikujúcich uzloch, alebo si aplikácia vie/chce zabezpečiť spoľahlivosť prenosu **sama** (SNMP, TFTP)

Ako odlíšiť viaceré konverzácie?

- Rôzne aplikácie posielajú a prijímajú dáta cez sieť v jednom čase, s rôznymi požiadavkami na prenos.
- Transportná vrstva ich musí odlíšiť a manažovať.
- Používa na to jedinečné identifikátory – **čísla portov**

Rôzne procesy na tom istom uzle dostanú pre svoje spojenia nad daným transportným protokolom rôzne porty.

Čísla portov v rôznych transportných protokoloch sú nezávislé, t.j. **TCP/80** je iný ako **UDP/80**

Pojmom transportný port sa označuje číslo, ktoré operačný systém priradí konkrétnej komunikujúcej aplikácii (procesu)pre konkrétne sieťové spojenie v danom transportnom protokole.

Známe čísla portov:

Sockets and Ports (cont'd.)

Port number	Process name	Protocol used	Description
20	FTP-DATA	TCP	File transfer—data
21	FTP	TCP	File transfer—control
22	SSH	TCP	Secure Shell
23	TELNET	TCP	Telnet
25	SMTP	TCP	Simple Mail Transfer Protocol
53	DNS	TCP and UDP	Domain Name System
69	TFTP	UDP	Trivial File Transfer Protocol
80	HTTP	TCP and UDP	Hypertext Transfer Protocol
110	POP3	TCP	Post Office Protocol 3
123	NTP	TCP	Network Time Protocol
143	IMAP	TCP	Internet Message Access Protocol
443	HTTPS	TCP	Secure implementation of HTTP

Table 4-3 Commonly used TCP/IP port numbers