

INFOKOMMUNIKÁCIÓS SZOLGÁLTATÁSOK ÉS ALKALMAZÁSOK

Transzport protokollok II. / II.

Szabó Sándor

Huszák Árpád

BME Híradástechnikai Tanszék

szabos@hit.bme.hu



2011. május 11.,
Budapest

- ISO/OSI és TCP/IP architektúra
 - Szállítási réteg
- Alkalmazások típusai
- Vezeték nélküli hálózatok jellemzői
- Protokollok
- Mobilitás támogatás a szállítási rétegben
- TCP
 - protokoll ismertetése
 - torlódásszabályozás
 - TCP vezeték nélküli környezetben
 - TCP variánsok mobilitás támogatására:
 - Indirect-TCP
 - Snoop TCP
 - Mobile TCP (M-TCP)
 - Fast retransmit/fast recovery
 - Transmission/time-out freezing
- UDP

- További protokollok
 - UDP-Lite
 - DCCP
 - torlódáskezelő mechanizmusai
 - hangolás
 - SCTP
 - Multistreaming/multihoming
 - M-SCTP
 - RTP, RTCP
 - RTSP
- Teszt alkalmazások

UDP (emlékeztető)

- User Datagram Protocol [RFC-768]
- 1980
- Az UDP sokkal gyorsabb protokoll, mint a TCP protokoll
- Nem megbízható adatátvitel
- Multimédiás alkalmazások esetén jól alkalmazható, ahol a késleltetés a kritikus
- A TCP-vel ellentétben nem ellenőrzi az adatok sértetlen átvitelét
 - ezért nem képes az elveszett vagy sérült csomagok pótlására
- Ezen kívül a fogadás sorrendjét sem garantálja a vételi oldalon.

UDP/IP fejléc (emlékeztető)

- Source Port
 - A forrásportot azonosítja
 - Válaszolni erre a portra lehet
- Destination Port
 - Célportot azonosítja
- Length
 - A datagram mérete bájtokban
 - A fejléc és a felhasználói adat egy
- Checksum
 - 16-bit ellenőrzőösszeg
 - A fejléc és a felhasználói adatokra együtt számolandó

Source Port (16 bits)	Destination Port (16 bits)
Length (16 bits)	Checksum (16 bits)
Data....	

- Lightweight User Datagram Protocol (UDP-Lite) [RFC 3828]
- 2004
- Az UDP módosítása
- Az UDP és az UDPLite minden jellemzője megegyezik
 - Nem kapcsolat orientált
 - Nincs hibajavítás
 - Nincs nyugtázás
- Részleges ellenőrzőösszeg alkalmazása (partial checksum)

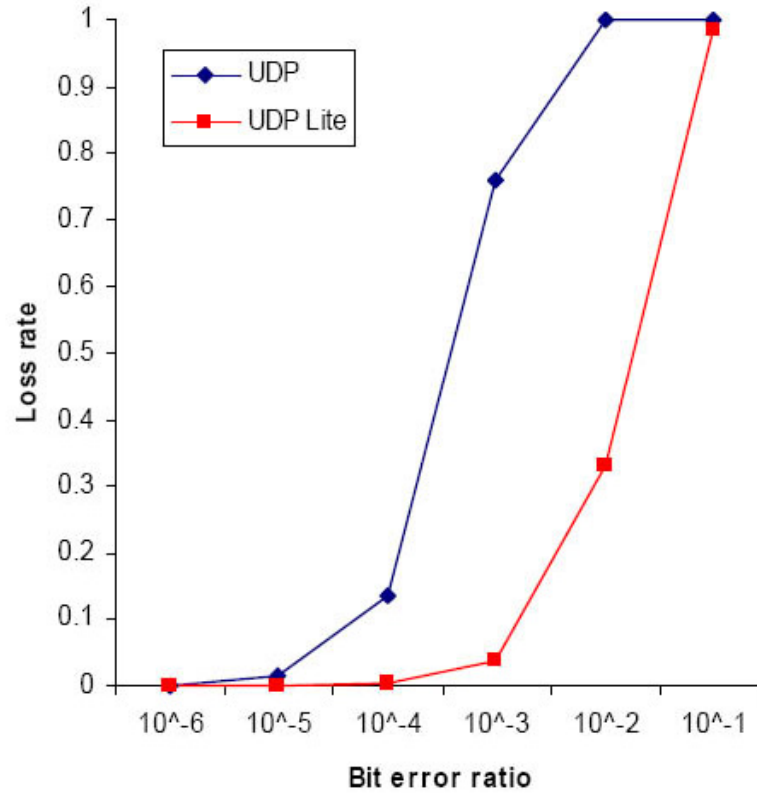
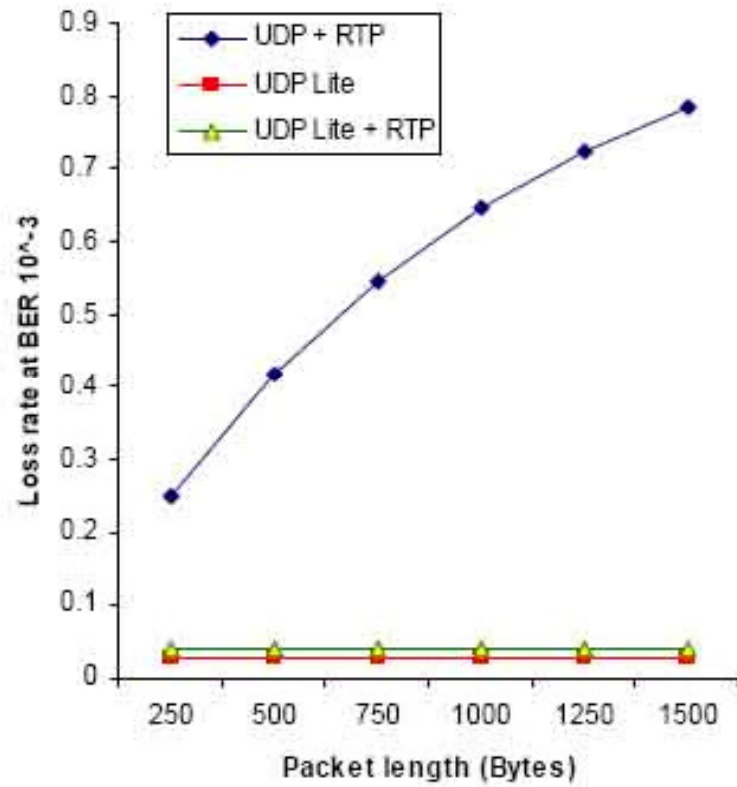
UDP-Lite fejléc

Source port	Dest port
Checksum Coverage	Checksum
Payload	

- Checksum Coverage
 - A csomag adatmezőjében mekkora terület legyen lefedve ellenőrzőösszeggel (Checksum)
 - ChCov=0 a teljes csomag lefedve
 - ChCov=1..7 hibás (a fejléctet mindig le kell fedni)
 - ChCov=8 csak a fejléc
 - ChCov>8 az adatmező egy része is le van fedve

- Ha a csomagnak abban részében keletkezik bithiba, amelyet a részleges ellenőrzőösszeg lefed, akkor a vevő érzékeli a hibát és eldobja a csomagot
- Ha a hiba olyan helyen van, amit a részleges ellenőrzőösszeg nem fed le, akkor nem dobja el
 - Ebben az esetben az alkalmazásnak kell kezelnie a hibás csomagot
 - Ha az UDP ellenőrzőösszege az egész csomagra kiterjed, az UDP Lite működése megegyezik a hagyományos UDP működésével.
- Mobil környezetben előnyös

- Az alkalmazások egy csoportja kezelni tudja a hibás csomagokat is
- A felhasználó által megfigyelhető minőség jobb lesz, ha a hibás csomagok nem kerülnek eldobásra, hanem az alkalmazásig eljutnak a hibás csomagok
 - Több hang és video codec is ehhez az alkalmazás csoportba tartozik:
 - ITU-H.263
 - ITU-H.265
 - MPEG-4 video codec (ISO-14496)
 - Ezek a kódolók jobb minőséget nyújtanak hibás csomagok kezelésével, mint ha egyáltalán nincs is csomag.



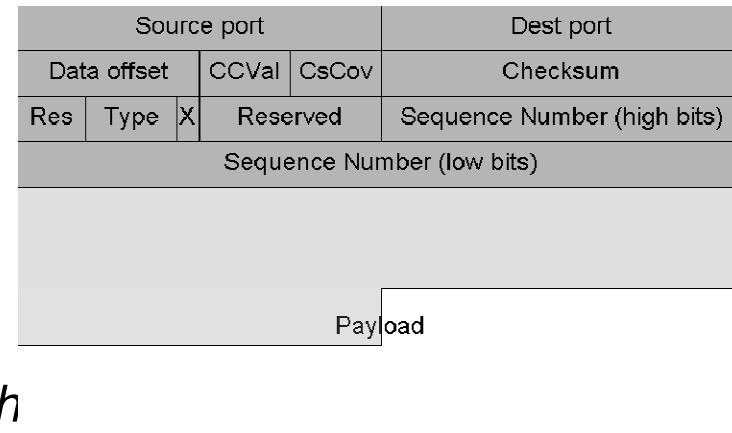
Video packet loss rate for 1400 byte packet size

- Az UDPLite jelentősen jobban teljesít az UDP-vel összehasonlítva, amennyiben hibázás megengedett.
- A csomagvesztés is lényegesen csökkenthető.

- Datagram Congestion Control Protocol
- Első draft 2001-ben
- RFC-4340, 2006 március
- Linux kernelben már implementálva
 - 2.6.14 – DCCPv4
 - 2.6.16 – DCCPv6
- Megbízhatatlan transzport protokoll
 - Nincs újraküldés
 - Van nyugtázás
- Kapcsolatorientált
 - Kapcsolatkiépítés
 - Három-utas kézfogással
- Torlódásszabályozási algoritmust használ
- Sorrendhelyes csomagtovábbítás
- Cél: TCP és az UDP előnyeit egy protokollként valósítsák meg

DCCP fejléc

- A fejléc hossza
 - minimálisan 12/16 byte (*X*)
 - maximálisan pedig a 1024 byte-t is elérheti, ha az opcionális mezőket, és az egyes csomag típusok esetén használt pótlólagos mezőket is használjuk
- Részleges ellenőrzőösszeg (*CsCov*, *Ch*)
 - UDP Lite-hoz hasonlóan, a DCCP is lehetővé teszi az adatok részleges lefedését ellenőrzőösszeggel
- Torlódásszabályozó algoritmus választása (*CCVal*)
 - TCP-like Congestion Control
 - TCP-Friendly Rate Control
- Üzenettípusok (*Type*)
 - Request, Response, Data, Ack, DataAck, CloseReq, Close, Reset, Sync, SyncAck



DCCP tulajdonságai

- TCP és UDP alapján egy új protokoll
- Átvett módszerek
 - Portok használata
 - Ellenőrzőösszeg
 - Sorszámozás
 - Különbözik a TCP-től
 - Nyugtázási mechanizmus
 - Adatcsomaghoz csatolt nyugta
 - Háromutas kézfogás kapcsolat-felépítésnél
- Új jellemzők
 - Választható torlódásszabályozás
 - Paraméterek beállítása kapcsolat-felépítésnél
 - Két egyirányú kapcsolat ($A \rightarrow B$, $B \rightarrow A$)

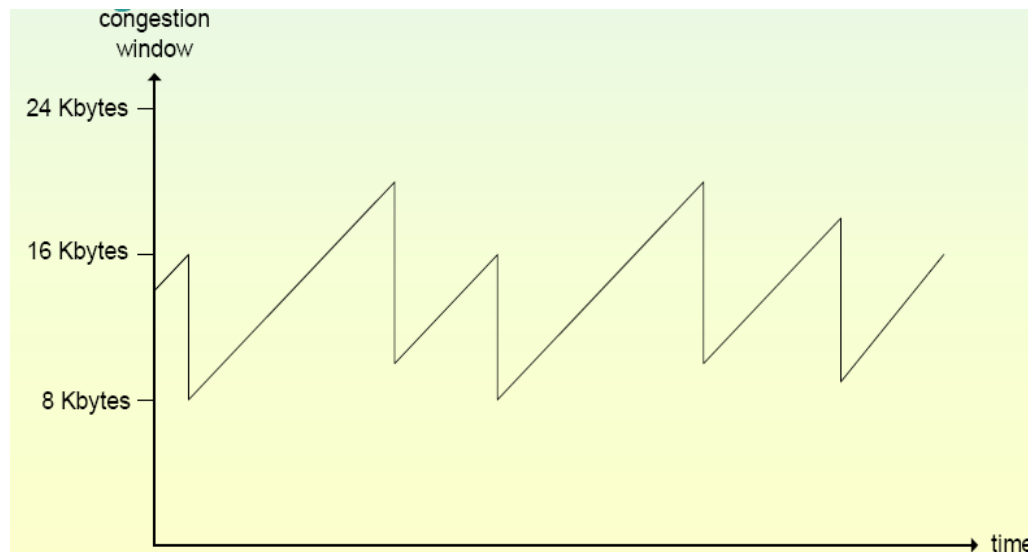
DCCP sorszámozás

- Követelmények
 - Hibadetektálási képesség
 - Csomagonként sorszám, bájttonként sorszámozás (TCP) helyett
 - A megbízhatatlan adatfolyamot feldolgozó alkalmazások számára sokkal fontosabb, hogy melyik csomag hiányzik, mint az hogy melyik adatrész
- Megoldás
 - Nyugtázási sorszám sok esetben része a csomagnak
 - Sorszám minden csomag fejlécében
 - Nincs együttes (kumulatív) nyugtázás, hiszen újraküldés sincs
 - A nyugtázási sorszám, a legutolsó megkapott csomag sorszáma, nem pedig a legelső meg nem kapott csomagé (~TCP)

DCCP torlódásszabályozás

- Lehetőség van a választásra
 - Sőt az algoritmus kommunikáció közben is megváltoztatható
- Jelenleg elérhető torlódáskezelő algoritmusok:
 - CCID2: TCP-like
 - RFC-4341
 - Slow start, AIMD (Additive Increase, Multiplicative Decrease), csomagvesztési esemény
 - Hasonló mint a TCP + SACK
 - Hirtelen sebességingadozás
 - CCID3: TFRC
 - RFC-4342
 - Kiegyensúlyozottabb mint az AIMD

- A TCP-hez hasonlóan megfelel a torlódási ablakot (cwnd)
- SlowStart
- Gyorsan alkalmazkodik az igénybe vehető sávszélességhez
- Hirtelen sebességcsökkenésre lehet számítani



CCID3: TCP-Friendly Rate Control

- Csomagvesztés esetén a sebességcsökkentés annyira hirtelen történik
- Minimalizálja a küldési sebesség hirtelen változásait
- A TFRC algoritmus a hálózati paramétereket használja a küldési sebesség (~ szabad sáv szélesség) kiszámításához

- s – csomagmennyiség
- $T = \frac{s}{R\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1 + 32p^2)}$
- R – RTT súlyozott átlaga
 $R = q \cdot R + (1-q) \cdot R_{sample}; q=0,9$
- t_{RTO} – a TCP újraküldési időzítője ($t_{RTO}=4RTT$)
- p – csomagvesztési arány

- A DCCP képes annak meghatározására is, hogy milyen okból történt csomagvesztés

Data Dropped Option:

Drop Code Meaning

0	Protocol Constraints
1	Application Not Listening
2	Receive Buffer
3	Corrupt
4-6	Reserved
7	Delivered Corrupt

- Ha például bithiba kerül sor csomageldobásra, nincs szükség torlódásszabályozó algoritmus beavatkozására.
- A DCCP-t olyan alkalmazások számára fejlesztették ki, mint például a streaming médiaalkalmazások, amelyek ki tudják használni a DCCP beépített szabályozási módszereit
- Mobil környezetben jól teljesít

- draft-phelan-dccp-lite-00.txt
- 2003 augusztus
- A DCCP egyszerűsített változata
- Néhány opciót kivettek, így csökkentve a protokoll komplexitását
- Nincs részleges ellenőrzőösszeg funkció sem
- DCCP-Lite fejléc

Source port		Dest port
Sequence Number (low bits)		
Type	TypeSpec	Checksum

- Legfontosabb állítható paraméter a CCID (congestion control ID), vagyis az alkalmazott torlódáskezelő algoritmus
- Beállítás a `sysctl` paranccsal történik:

- **CCID2:**

```
sysctl -w net.dccp.default.rx_ccid=2
```

```
sysctl -w net.dccp.default.tx_ccid=2
```

- **CCID3:**

```
sysctl -w net.dccp.default.rx_ccid=3
```

```
sysctl -w net.dccp.default.tx_ccid=3
```

- A *sequence_window* (default =100) paraméter adja meg, hogy a csomagsorszámok mekkora halmaza érvényes.
- Túl kis érték esetén a végpontok kieshetnek a szinkronból
- Javasolt nagyobb értékre állítani:

```
sysctl -w net.dccp.default.seq_window=10000
```

- Próbálkozások száma a kapcsolatfelépítésre

```
net.dccp.default.request_retries = 5
```

- Nyugta újraküldési próbálkozások száma, kapcsolatmegszakadás előtt

```
net.dccp.default.retries2 = 15
```

DCCP kliens/szerver alkalmazás

- A módszer hasonlatos a TCP/UDP esethez
- A *Socket* létrehozásánál az UDP protokollt kell megadnunk
`socket(AF_INET, SOCK_DCCP, IPPROTO_DCCP)`

- Stream Control Transmission Protocol
- RFC-2960
- 2000
- A Linux kernel része a 2.6.x verziókban
- Megbízható
 - Hibamentes
 - Duplikáció-mentes
 - Nem sorrendhelyes/vagy sorrendhelyes (beállítható)
- Több folyam kezelésére egy kapcsolaton belül
- Multihoming
 - Több IP-cím
- Torlódásszabályozás
- Slow start
- MTU (Maximum Transfer Unit) felderítés

- A TCP, UDP nem elégíti ki az összes alkalmazás igényeit
- Fejlődését leginkább az IP telefónia és az ott alkalmazott jelzésrendszer indította
- A TCP-hez hasonlóan megbízható és full-duplex kapcsolatot alkalmaz
- A TCP-vel és UDP-vel ellentétben olyan opciókat is nyújt, amelyek a multimédiás alkalmazások esetén jelent előnyt
- TCP-hez hasonló torlódáskezelő algoritmust használ
- Azonos hosztok közötti folyamatok összefogása
- Kapcsolatfelépítés: 4-utas kézfogás
- Kapcsolatbontás: 3-utas kézfogás

- Csomag formátum
 - A TCP-vel ellentétben, az SCTP üzenet-orientált adattovábbítást nyújt
- Common header
 - forrás és cél portszám
 - ellenőrző rész
 - Checksum
- – Chunk header
 - Chunk hossz
 - Típus (14 különböző chunk típus)

DATA chunk

INIT chunk

INIT ACK chunk

SACK chunk

HEARTBEAT chunk

HEARTBEAT ACK chunk

ABORT chunk

SHUTDOWN chunk

SHUTDOWN ACK chunk

ERROR chunk

COOKIE ECHO chunk

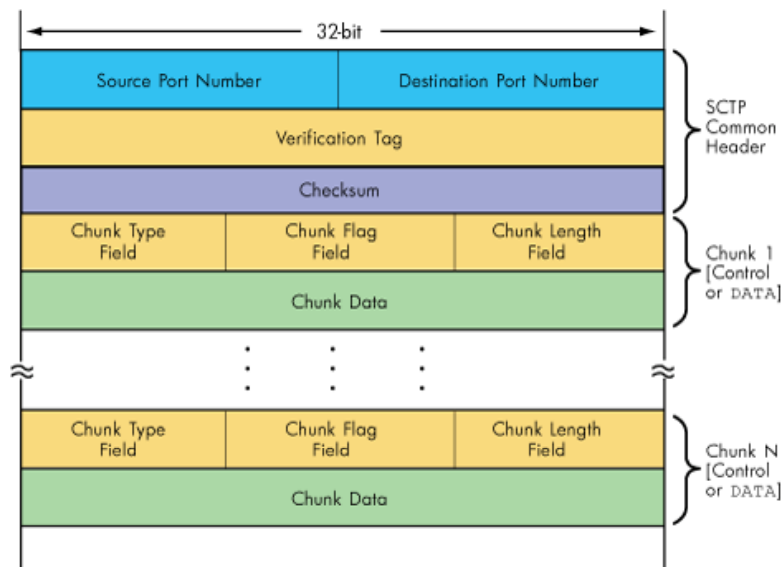
COOKIE ACK chunk

ECNE chunk

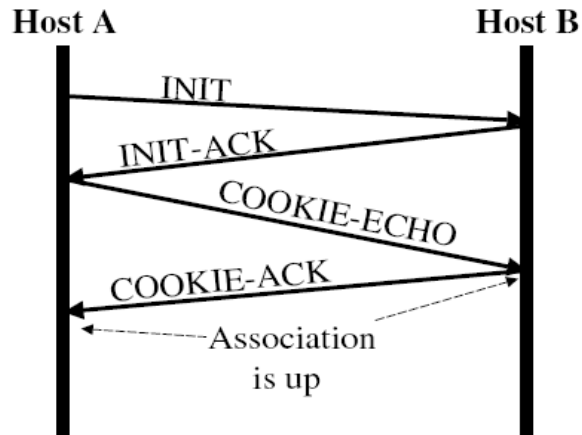
CWR chunk

SHUTDOWN COMPLETE chunk

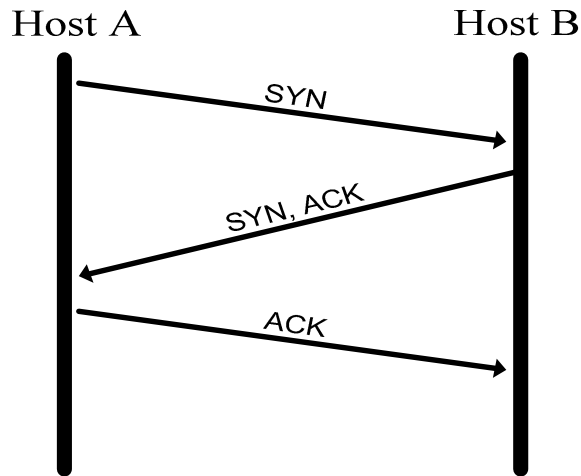
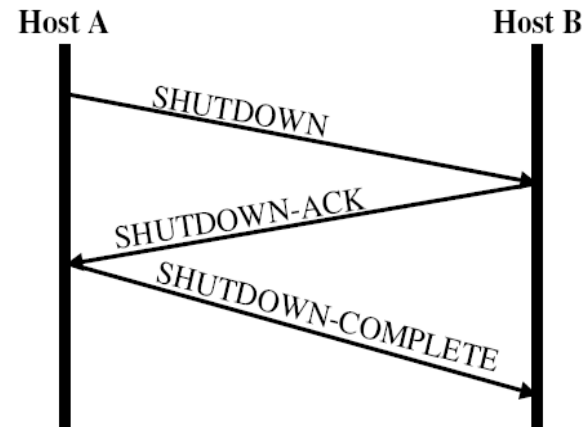
- speciális flag-ek



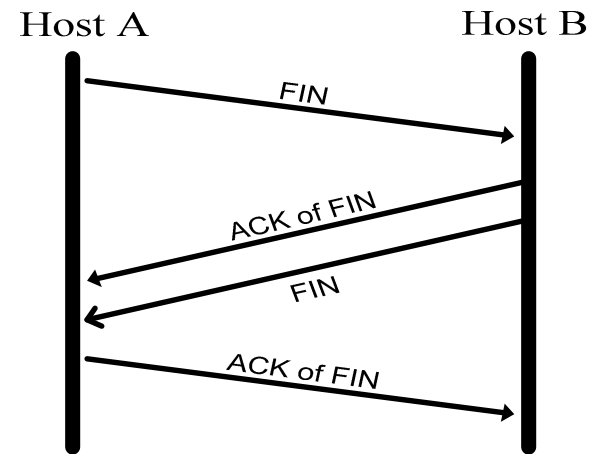
SCTP kapcsolatfelépítés



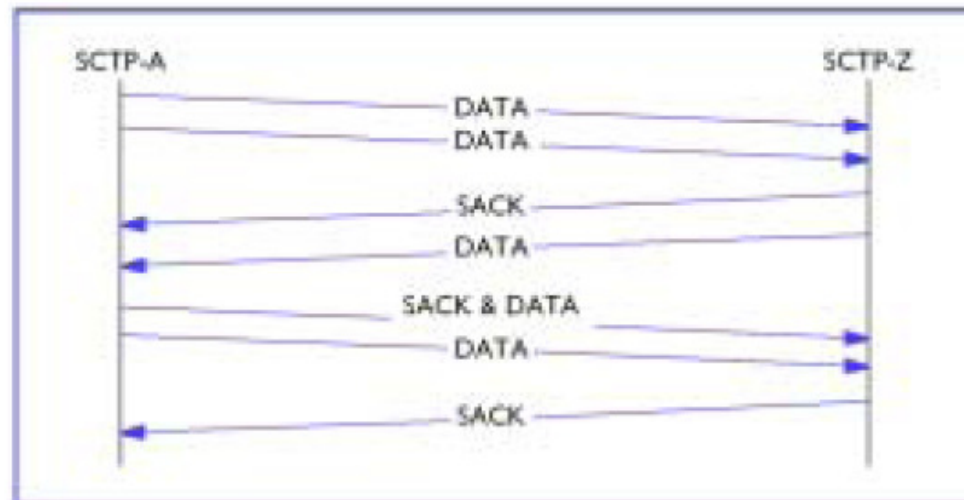
SCTP



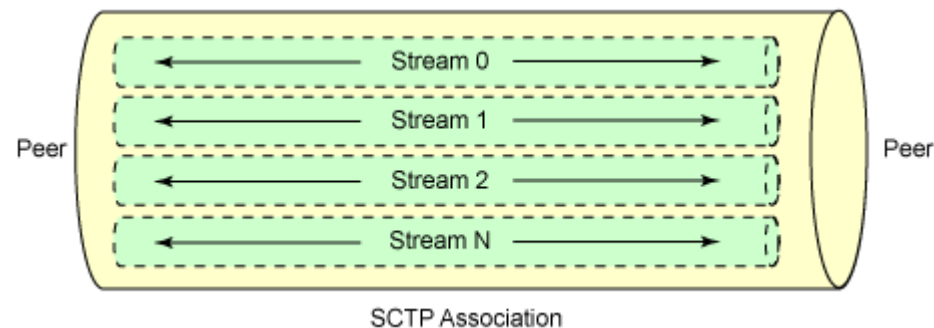
TCP



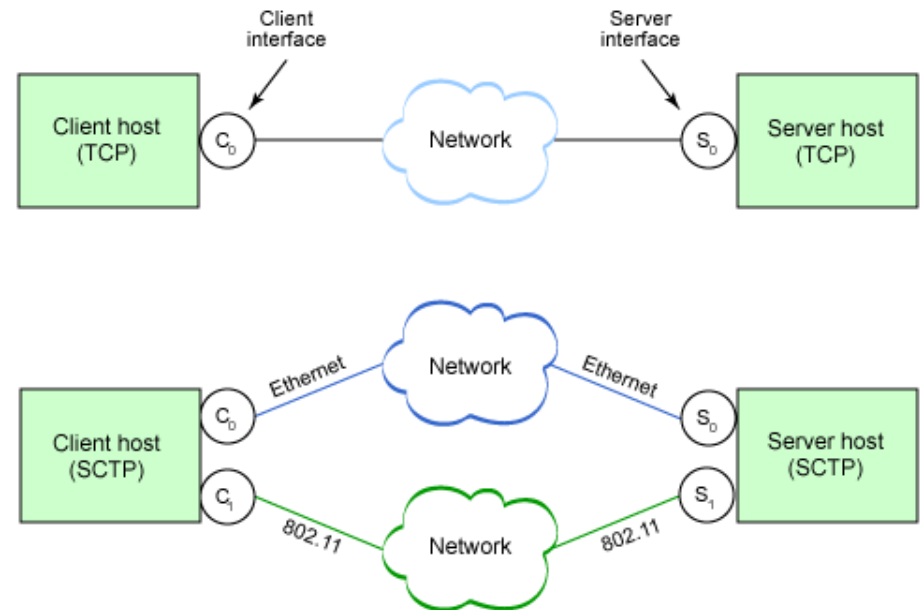
- Selective Acknowledgement
 - kumulatív nyugta
 - nyugta az utolsó hibátlanul fogadott TSN-re (Transmission Sequence Number)
 - max. 500 ms időközönként (állítható)



- Rendkívül fontos tulajdonsága az SCTP-nek, hogy egy kapcsolaton belül képes több adatfolyamot továbbítani
- Míg a TCP-ben ehhez külön kapcsolatokra van szükség
- A független adatfolyamok külön chunk-okban kerülnek továbbításra, de egy csomagon belül
- Jó felhasználási lehetőség pl. a vezérlő és felhasználói adatok szétválasztása
 - TCP esetében meg kell várni, hogy a felhasználói adat továbbítódjon és csak utána érkezik a nagyobb prioritású vezérlő adat
- Az SCTP párhuzamossá teszi a folyamatok továbbítását, így csökkentve a késleltetést is
- A független folyamatokra, különböző tulajdonságokat állíthatunk be, mint pl. a sorrendhelyesség



- Egy multihome hoszt azzal a tulajdonsággal rendelkezik, hogy több interfészen érhető el, azaz több IP címe is van
- Az SCTP képest tehát egy összeköttetés adatait több interfészen küldeni és fogadni
- Jelenleg ez az egyetlen transzport protokoll, amely erre képes
 - Ha az elsődleges címen nem lehet elérni, akkor átvált a másik címre



- Több útvonal közül 1 elsődleges (*primary*) link - *primary address*
 - Erre a linkre (címre) próbálja először küldeni a csomagokat
 - Többi cím: *secondary address*
- Monitorozás
 - Mely linkek „élnek”?
 - HEARTBEAT üzenetek folyamatos kiküldése a bekötött címekre, így kiderül, hogy mely link inaktív.

A szolgáltatások és jellemző tulajdonságok áttekintése

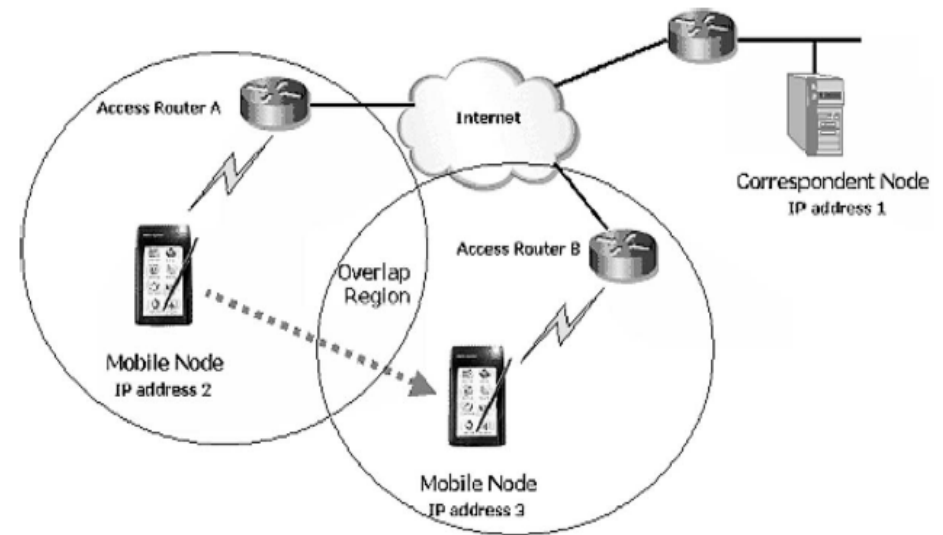
	SCTP	TCP	UDP
• Full-duplex adatátvitel	igen	igen	igen
• Kapcsolat orientált	igen	igen	nem
• Megbízható adatátvitel	igen	igen	nem
• Megbízhatatlan adatátvitel	igen	nem	igen
• Részlegesen megbízható adatátvitel	igen	nem	nem
• Sorrendhelyes átvitel	igen	igen	nem
• Nem sorrendhelyes átvitel	igen	nem	igen
• Flow- és Congestion Control	igen	igen	nem
• ECN támogatás	igen	igen	nem
• Szelektív nyugtázás	igen	igen	nem
• Fragmentálás	igen	igen	nem
• Multistreaming	igen	nem	nem
• Multihoming	igen	nem	nem
• SYN flooding támadás elleni védelem	igen	nem	n/a
• Half-closed kapcsolat	nem	igen	n/a

Mobile SCTP (mSCTP)

- Az SCTP protokollt arra tervezték, hogy a TCP-t és esetleg még az UDP-t is leváltsa
- Hasonlít a TCP-re, de jóval többre képes annál, például multi-streaming és multi-homing támogatása
- A multi-homing az az új tulajdonság, ami miatt az SCTP alkalmas lehet mobilitás kezelésére, méghozzá úgy, hogy nincs szükség agent-re
- A mobilitás úgy van megvalósítva, hogy a végpont úgy változtassa meg az IP címét, hogy közben a végpont-végpont kapcsolat nem szakad meg
 - ennek dinamikusan kell történnie
- Egy asszociáció felépítése során a kommunikáló felek kicserélik egymással a lehetséges transzport címeiket (IP és port párosok)

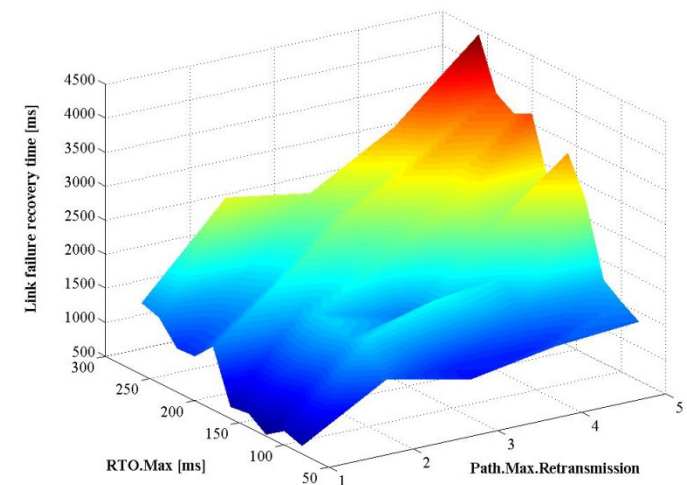
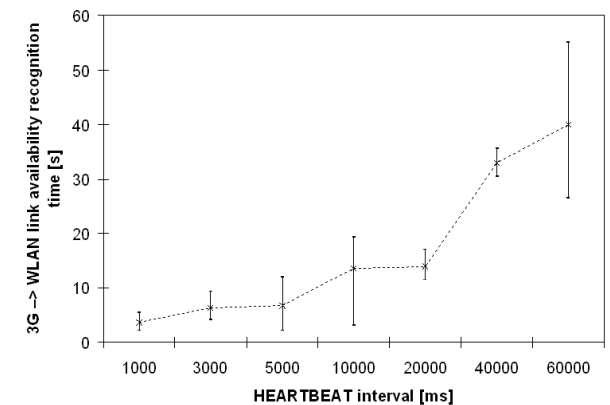
Mobile SCTP (mSCTP)

- Az ADDIP-vel kiegészített SCTP-t mobile SCTP-nek (mSCTP) nevezik
- ADDIP (Dynamic Address Reconfiguration)
 - kiegészítés az SCTP-hez
 - lehetővé teszi, hogy hozzáadjunk, elvegyünk és megváltoztassunk IP címeket (MN – Mobile Node) egy aktív kapcsolat alatt
 - Események: ADD; DELETE;
 - CHANGE
- A CN-t (Correspondent Node) is értesíteni kell a változásról
 - SCTP ASCONF (Address Configuration Change) chunk (MN → CN)
 - SCTP ASCONF-ACK chunk (CN → MN)

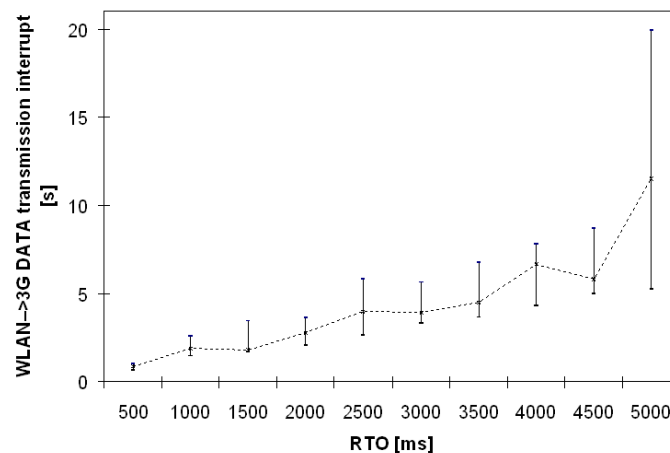


SCTP hangolás

- A multihoming tulajdonság alkalmazásával tudjuk a protokoll mobilitáskezelését kihasználásával.
- A sysctl paranccsal változtatható protokoll paraméterek:
 - mSCTP engedélyezése
`net.sctp.addip_enable = 1`
 - Heartbeat interval
 - Milyen gyakran ellenőrizze az útvonalak épségét
 - Nagy hatással van a handover sebességére, érdemes alacsonyabb értékre állítani
`net.sctp.hb_interval = 30000`
 - Újraküldési próbálkozás maximuma (PMR)
 - Path Max. Retransmission
 - Hányszor próbálja meg az elveszett csomagot újraküldeni, mielőtt a másodlagos útvonalakra váltana
 - Nagy érték esetén, nagyon sokáig tart mire rájön, hogy a link nem él
`net.sctp.path_max_retrans = 5`

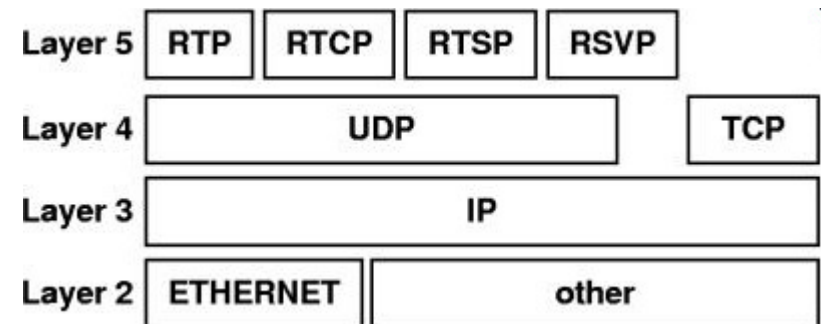


- További paraméterek:
 - Újraküldési időzítő
 - Mikor tekinthető egy csomag elveszettnek, vagyis mekkora késleltetéssel lehet újraküldeni
 - Ha nagy az érték, sokáig tart mire a megadott újraküldési próbálkozások (PMR) megtörténnek
 - A handover lelassul

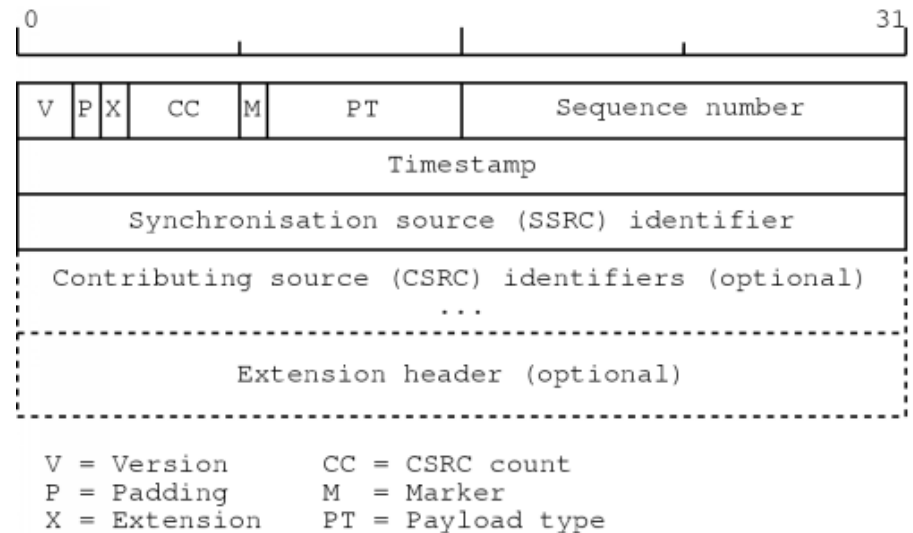


- A módszer hasonlatos a többi esethez
- A *Socket* létrehozásánál az SCTP protokollt kell megadnunk
`socket(AF_INET6, SOCK_STREAM, IPPROTO_SCTP)`
- A programkódban adható meg, hogy a küldendő adatokat mely folyamba (stream) helyezzük
`sctp_sendmsg(.,.,,stream_no,.,.)`
- Új IP címekeket adhatunk meg
`sctp_bindx()`

- Real-Time Transport Protocol
- RFC 1889 (RFC 3550)
- 1996 (2003)
- Az RTP protokoll nem egy valódi szállítási rétegbeli szállítási protokoll
- Általában alkalmazásokba integrálva jelenik (nem önálló hálózati réteggént)
 - audio alkalmazások
 - video alkalmazások
 - audio és video információk egyidejű továbbítása
- Multicastra készült, de unicast felett is működik



- RTP szolgáltatásai:
 - Típus azonosítás
 - Sorszámozás
 - Időzítés jelzés (időbélyeg)
 - UDP-t használ általában (felhasználja annak multiplexelési és ellenőrzőösszeg képzési szolgáltatásait)



- Időzítés információ felhasználása:
 - az audió és videó adatok összehangolása,
 - az audió és videó adatok kijelzése
 - elveszett vagy soron kívül érkezett csomagok meghatározása.

- A felhasználók hálózati hozzáférésük alapján lehetőség van többféle kódolást alkalmazni
 - pl. lassú hozzáférés esetén erős tömörítés - gyengébb minőség
 - Különböző QoS jellemzőkkel rendelkező felhasználóknak egyidejű szolgáltatás nyújtása
- A multimédia folyam azonosítása az *összehangolási forrással* (*synchronization source, SSRC*) történik
 - Az SSRC meghatározza a folyam forrását (véletlen azonosító)
 - Egy összehangolási forrás változtathatja a videó- és hangkódolást.
- Közreműködő forrás (*Contributing source, CSRC*) olyan forrás, melyből érkező adatfolyamot egy RTP keverő felhasznál
 - A keverő a csomag létrehozásakor CSRC listát készít a felhasznált források SSRC azonosítója alapján

- Realtime Transport Control Protocol
- RFC 3550
- Szabályozza az RTP kapcsolatot
 - statisztikákat küld
 - szinkronizálja az RTP kapcsolatokat (pl. videókonferenciánál a hangot és a képet)
- A szolgáltatás minőségével kapcsolatos visszajelzéshez használják.
- Periodikus kontroll információkat juttat el a résztvevőkhöz
- Feladatai:
 - visszajelzési lehetőség az adattovábbítás minőségéről (torlódáskezelés)
- minden RTP forrásnak azonosítót továbbít (CNAME)
- a résztvevők kontroll üzeneteit elküldi a többi résztvevőnek, szükség esetén saját kontroll csomagokat is küld

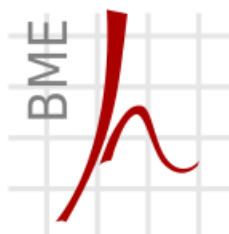
- Real-Time Streaming Protocol
- Média streaminghez olyan vezérlő protokoll, amellyel a kliens a szerver felé videomagnó szerű parancsokat küldhet (pl. lejátszás, szünet).
- A tartalom átviteléhez RTP-t, vagy esetleg más, gyártó-specifikus protokollt használ.

- Számos alkalmazás elérhető, melyek lehetővé teszik a bemutatott transzport protokollok vizsgálatát
- Ezek közül a legismertebbek:
 - Iperf
 - IPv4/IPv6
 - TCP/UDP
 - Küldési sebesség beállítás
 - Fogadó/küldő oldali statisztikák
 - D-ITG
 - IPv4/IPv6
 - UDP, TCP, ICMP, SCTP, DCCP
 - Számos protokoll beállítás
 - Natsend
 - IPv4/IPv6
 - UDP, TCP, SCTP, DCCP, UDPlite
 - Multicast

- Bevezetés
 - ISO/OSI és TCP/IP architektúra
 - Alkalmazások típusai
 - Vezeték nélküli hálózatok jellemzői
- Protokollok
 - TCP és variánsai
 - UDP
 - DCCP
 - SCTP
 - RTP/RTCP/RTSP
- Hangolási lehetőségek
- Kliens/szerver alkalmazás
- Eszközök
 - Iperf, natsend, D-ITG, stb.

Kérdések?

KÖSZÖNÖM A FIGYELMET!



Híradástechnikai Tanszék

Szabó Sándor
Huszák Árpád
BME Híradástechnikai Tanszék
szabos@hit.bme.hu

