

3. gyakorlat

Ismétlés

Számítási példák

Illusztrációk

Demonstrációk

Áttekintés – 1. rész

- Weighted Fair Queueing (WFQ)
 - ismétlés
 - példa
 - animáció
- Csomagtovábbítás adatkapcsolati és hálózati rétegben (példa)
- IP csomagtördelés (példa)

Áttekintés – 2. rész

- IPv4 a gyakorlatban (Soproni Péter)
 - IP-beállítások
 - Windowsban (GUI és parncssor)
 - FreeBSD-ben
 - Parancssori eszközök
 - Hálózatmonitorzás Wiresharkkal
- OSPF routing (Bessenyei Csilla)
 - Routing topológia kialakítása fizikai topológián az OSPF segítségével
 - Routing topológia változása költségek és a linkek állapota alapján

Weighted Fair Queueing (WFQ)

Ismétlés

Példa

Animáció

A WFQ (Weighted Fair Queueing)

- Alapötlet: kiszámítjuk (szimuláljuk) a csomagok távozási időpontjait, mintha GPS szerint szolgáltuk volna ki azokat, és ezt a sorrendet alkalmazzuk a kiszolgálásra
- Nem a távozási időpont, hanem a sorrend érdekes
 - **ezért a távozást jellemző értéket így *befejezési számnak (finish number)* nevezik**
- (Legyen bitenkénti kiszolgálás)
- **Ciklusszám (*round number*)** – pozitív nem egész szám
 - az aktív felhasználók számának reciprokával arányos
- **Ciklushossz:** arányos az aktív felhasználók számával
- Ha ismerjük a ciklusszámot, a befejezési szám kiszámolható:
- $F(i,k,t) = \max[F(i,k-1,t), R(t)] + P(i,k,t)$
 - $F(i,k,t)$ – az i -edik felhasználó befejezési száma t időpontban
 - $P(i,k,t)$ – az i -edik felh. t -ben beérkező k -edik csomagjának hossza
 - $R(t)$ – ciklusszám a t -ben

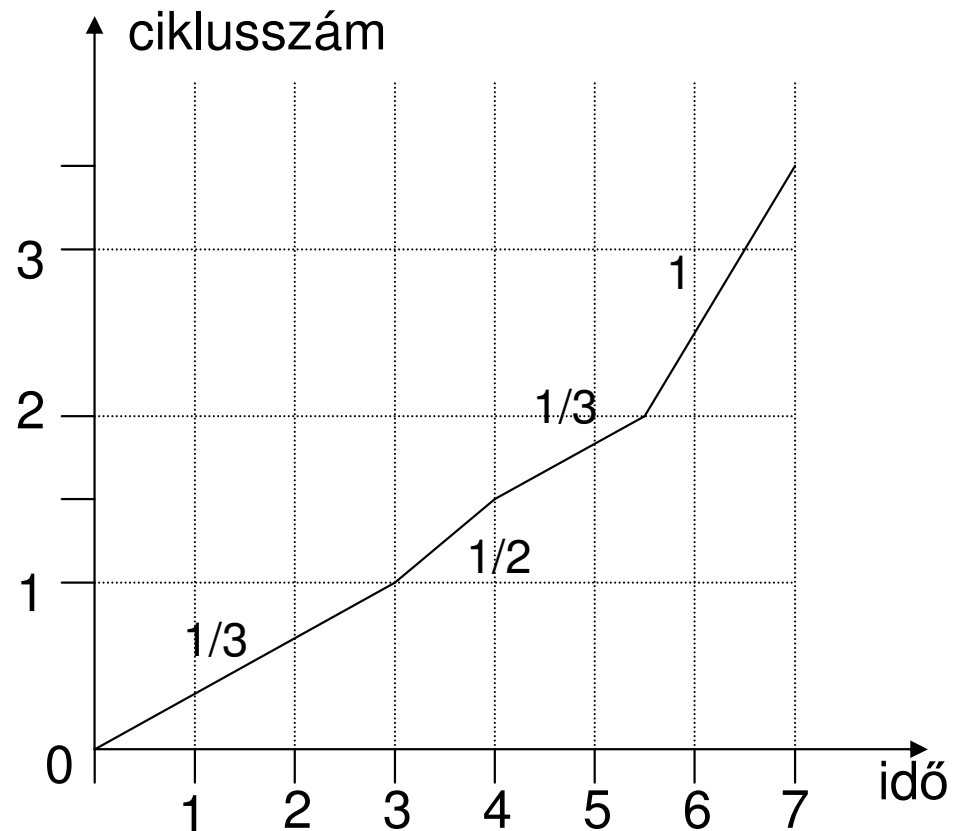
Hogy jön ki a WFQ összefüggés?

- *Inaktív* felhasználónál a befejezési szám:
 - az aktuális ciklusszám plusz a csomagméret
 - pl. ha egy 10 bites csomag érkezik, amikor a ciklusszám 3, a kiszolgálása akkor befejeződik be, amikor a ciklusszám 13 lesz
- *Aktív* felhasználó csomagja beérkezésekor annak befejezési száma:
 - a sorában lévő csomagok közül a legnagyobb befejezési számú plusz a csomagméret
 - pl. ha a 10 bites csomag beérkezésekor a sorban van egy 20-as befejezési számú, akkor 30 lesz
- Kombinálva ezt a két állítást:
$$F(i,k,t) = \max[F(i,k-1,t), R(t)] + P(i,k,t)$$

Példa a WFQ működésére

$$F(i,k,t) = \max[F(i,k-1,t), R(t)] + P(i,k,t)$$

- **A, B, C, 1; 2; 2 a csomaghosszak**
t=0-ban
és A még generál egy 2 hosszút
t=4-ben
- **Kiszolgálás:**
1 csomagegység/időegység
- **$R(0)=0$**
- **$F(A,1,0) = \max[F(A,0,0), R(0)] +$**
 $P(A,1,0) = \max[0,0] + 1 = 1$
- **$F(B,1,0) = F(C,1,0) = 2$**
- **A első csomagjával kezdjük**
- **majd B vagy C, ami t=3-ban**
fejeződik be
- **a harmadikat t=3-ban kezdjük és**
5-ben fejeződik be
- **t=4-ben A második csomagja**
- **A második csomag finish no-hoz**
kell tudnunk a round no-t, t=4-ben
- **A görbe menetéből...**



A WFQ értékelése

- Súlyozott esetben a befejezési szám:

$$F(i,k,t) = \max[F(i,k-1,t), R(t)] + P(i,k,t)/w(i)$$

- Az aktuális ciklusszám meghatározása jelent gondot:

egy kieső (kiszolgált) felhasználó megváltoztatja a ciklus-szám változási sebességét → felgyorsul a kiszolgálás → „láncreakció” következik/-het be

- Egyre általánosabb a használata a korszerű routerekben

További ütemezések

Ne feledkezzünk el az alábbi ütemezőkről se!

- ❑ Max-min algoritmus
- ❑ Súlyozott max-min algoritmus
- ❑ Weighted round-robin
- ❑ Deficit round-robin

Csomagtovábbítás adatkapcsolati és hálózati rétegben

Példa

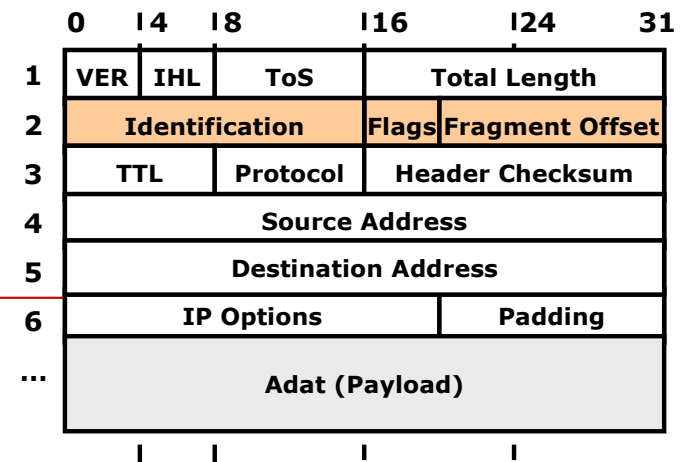
IP csomagtördelés

Ismétlés és példa

Tördelés szükségessége

- A hálózatok alsóbb rétegei meghatározzák a keret maximális méretét
 - Az adatkapcsolati réteg fej- és farokrészét leszámítva ez az MTU (Maximum Transmission Unit) (Ethernetnél 1500 byte)
- Eltérő technológiák
 - ⇒ eltérő MTU-val rendelkező kapcsolatok
 - ⇒ tördelni kell

Tördeléssel kapcsolatos mezők az IP-fejlécben



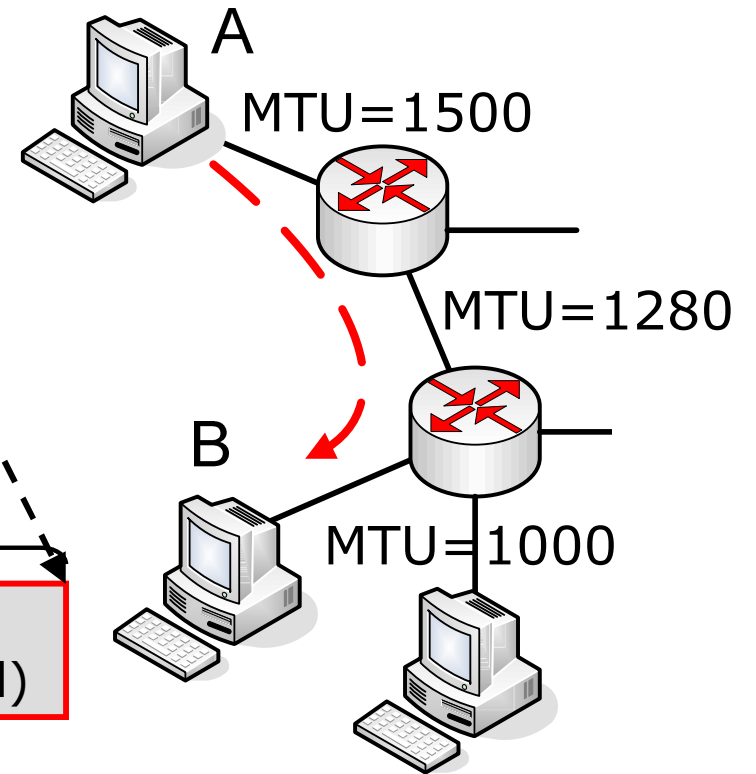
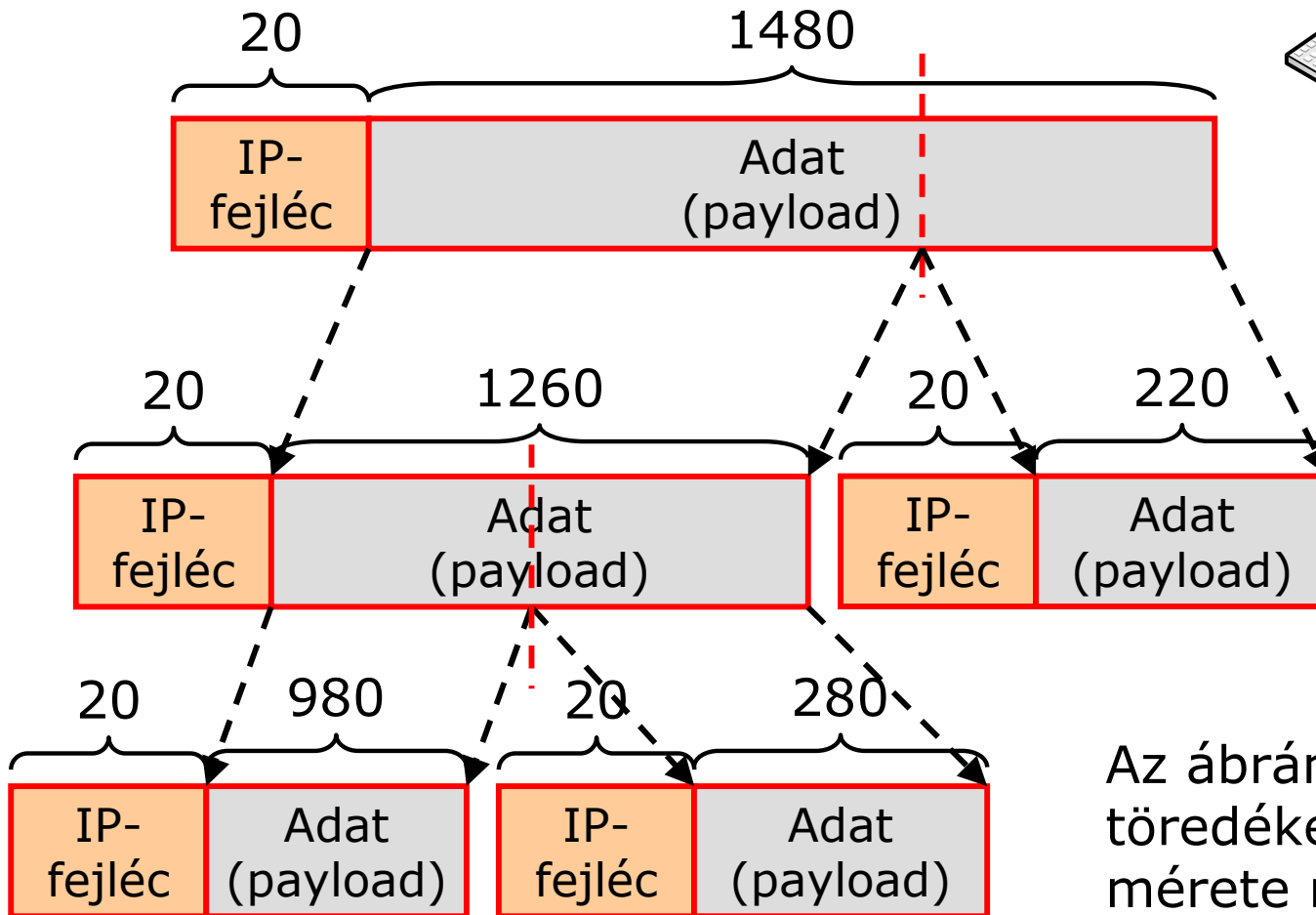
- Identification (16 bit)
 - Az IP-töredékek egyedi azonosítása
- Flags (3 bit)
 - 0: Fenntartott
 - 0-nak kell lennie
 - „Evil bit” (RFC 3514) (áprilisi tréfa 2003-ban)
 - 1: DF – Don’t Fragment
 - 1: ha tördelni kellene, el kell dobni
 - Ezt használják az MTU (Maximum Transmission Unit) Path Discovery néhány TCP verzióban és az IPv6-ban
 - 2: MF – More Fragment
 - 1: ha nem az utolsó töredék
 - 0: utolsó töredék vagy nem tördelt csomag
- Fragment Offset (13 bit)
 - Az eredeti csomagban lévő kezdőpozícióját adja meg e töredékben lévő adatnak **8 bájtos egységekben**
 - $(2^{13}-1) \times 8 = 65528 > 65515$ bájt (max. adat)
(így a maximális méretű csomag is tördelhető)

IP-fejléc változása tördelés közben

- Ha tördelés szükséges
 - Ha nem volt Identification mező érték, akkor generálni kell
 - Adat tördelése 8 bájtos egységekre
 - A tördelésnek megfelelő „fragmentation offset”-et beállítani az új csomagokban
 - Minden csomagban az MF bitet 1-re kell állítani, kivéve annak a csomagnak az utolsó töredékét, amelyben az MF nincs beállítva

A tördelés menete

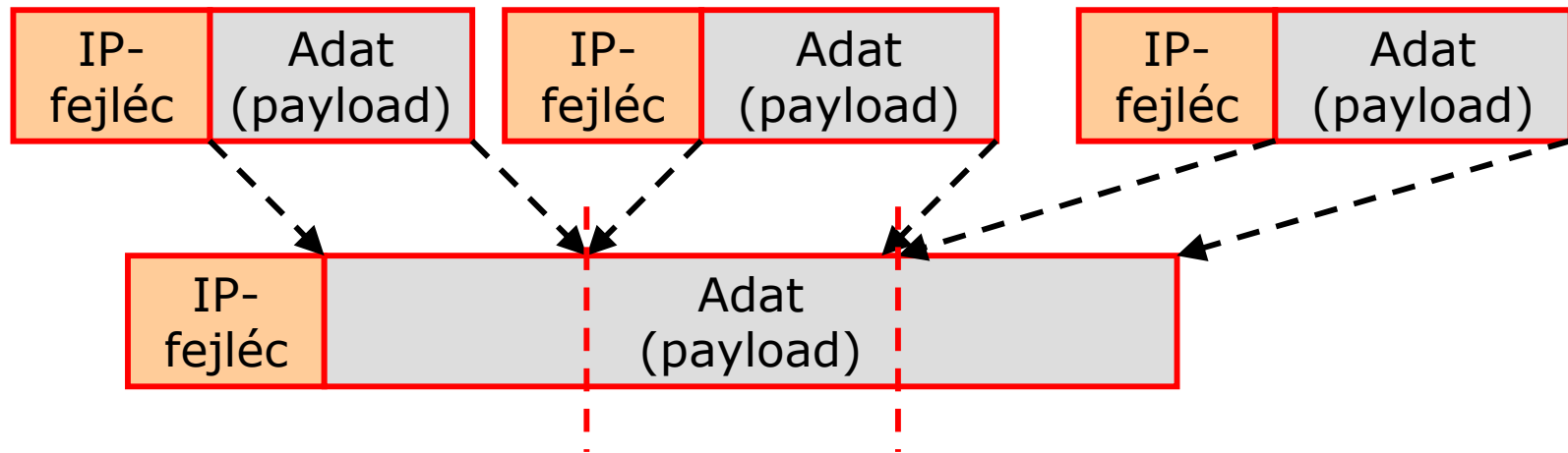
- Minden csomagnak kell fejléc!



Az ábrán helytelenül, a töredékek adatrészének a mérete nem 8 bájt többszöröse!

A töredékek összeállítása

- ❑ Csak a címzett végezheti el
- ❑ Ha valamely darab nem érkezik meg, akkor a többi is eldobja
- ❑ Csak teljesen összeállított csomagokat továbbít a felsőbb réteg felé
- ❑ A „fragment offset”-ekből a töredék helye meghatározható



Számoljunk!

- Egy 1500 bájtos (IP fejrészsel együtt) IPv4 csomagot küldünk 3 linken keresztül, melynek rendre a hálózati rétegre vonatkoztatott MTU-i 1500, 1300 és 1150 bájt.
- Számoljuk ki milyen hosszú töredékek érkeznek a címzetthez, és ezekben milyen értékűek az alábbi mezők:
 - Fragment offset
 - More fragment
- Mekkora a fej okozta overhead tördeletlen és tördelt esetben?

2. rész - Demók

- IPv4 a gyakorlatban (Soproni Péter)
 - IP-beállítások
 - Windowsban (GUI és parncssor)
 - FreeBSD-ben
 - Parancssori eszközök
 - Hálózatmonitorzás Wiresharkkal
- OSPF routing (Bessenyei Csilla)
 - Routing topológia kialakítása fizikai topológián az OSPF segítségével
 - Routing topológia változása költségek és a linkek állapota alapján