



Řízení toku v přístupových bodech

13.6.2009

Lukáš Turek
8an@praha12.net

O čem to bude

- Co způsobuje velkou latenci na Wi-Fi?
- Proč na Wi-Fi nefunguje běžný traffic shaping?
- Je možné traffic shaping vyřešit jinak a lépe?
- Kdy ani traffic shaping nepomůže?

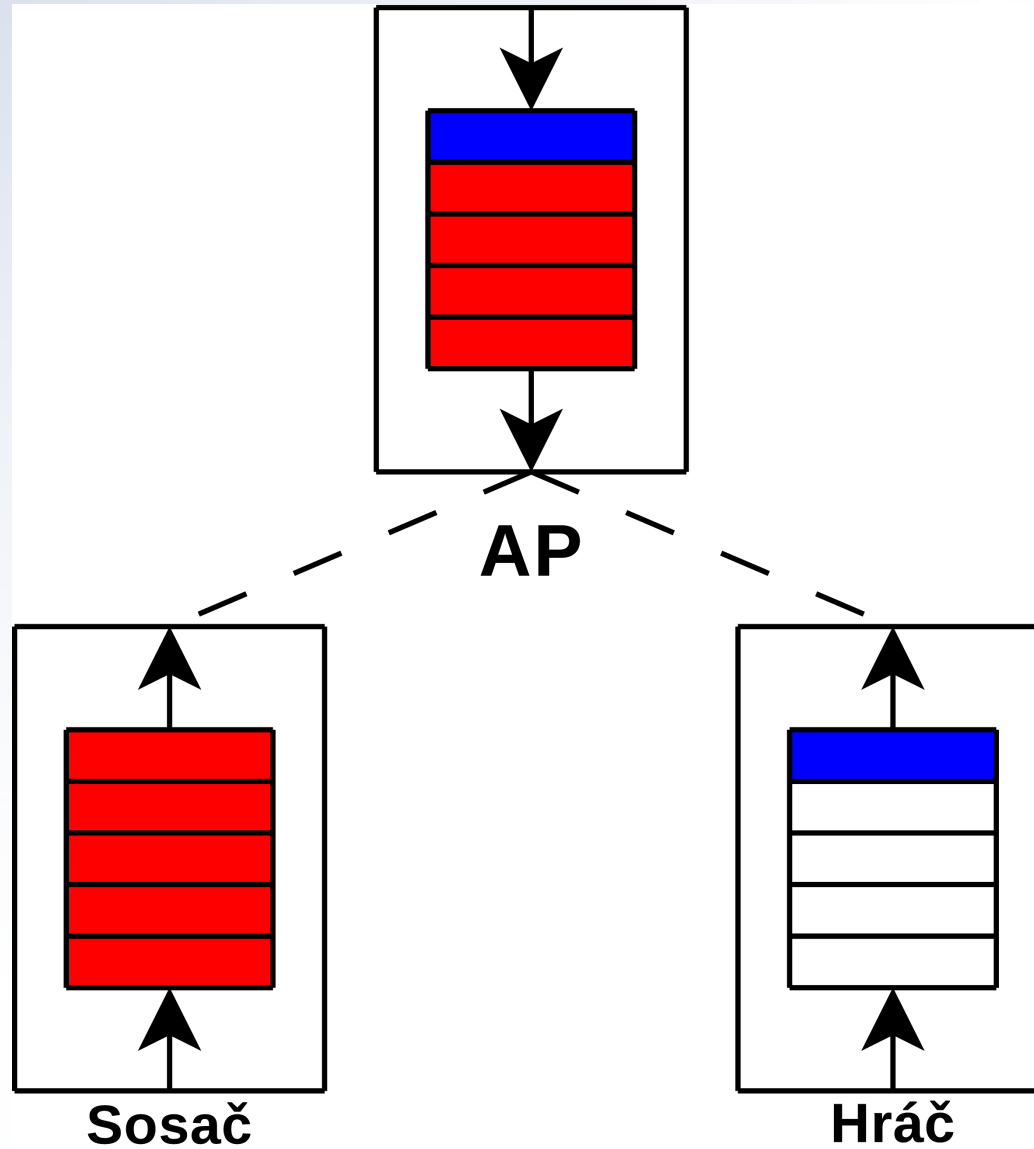


Zdroj latence



- Zpoždění paketu má dvě složky
 - doba přenosu po linkách
 - i u rychlosti 1 Mbit nejvýš několik milisekund
 - doba čekání ve frontě
 - má-li AP frontu na 100 paketů a bude zaplněna samými pakety velikosti 1500B, které se budou posílat jen 1 Mbit/s, bude její odesílání trvat více než vteřinu!
- Latenci tedy způsobuje fronta v AP
 - nelze ji zkrátit, pak by byly další pakety zahozeny
- Při velké latenci TCP zvětšuje okno
 - více paketů je odesláno bez potvrzení \Rightarrow víc čeká ve frontě
 - TCP se s latencí vyrovná, ale například VoIP ne
- Rozpor mezi dvěma typy uživatelů: „Sosač“ a „Hráč“
 - jak uspokojit oba najednou?

Fronty paketů



Klasický traffic shaping

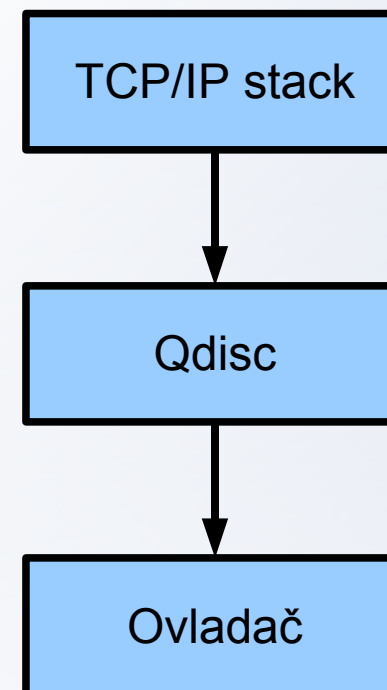


- Jak funguje traffic shaping v Linuxu nebo FreeBSD?
 - Nastaví se omezení datového toku na rozhraní (např. 5 Mbit)
 - Hlavní třída se rozdělí na podtřídy pro jednotlivé klienty
- Funguje na Ethernetu, ale ne na Wi-Fi
 - Wi-Fi je half duplex – upload snižuje přenosovou rychlost
 - Wi-Fi používá různé modulační rychlosti (1 – 54 Mbit/s)
 - Není-li přenos úspěšný, je opakován (až 7x)
 - Režie přenosů závisí na velikosti paketů
- U 802.11b je rozdíl efektivity 20x, u 802.11g až 100x
 - a to nebyly brány do úvahy retransmise
- Nastavení pevné rychlosti nepomůže
 - odstraní se jen jedna ze čtyř proměnných

Princip traffic shapingu



- Proč je potřeba nastavit u traffic shapingu rychlost?
 - Traffic shaping je implementován mezi TCP/IP stackem a ovladačem hardwaru
 - ovladač má vlastní frontu
 - u Atherosu na 100 paketů
 - aby traffic shaping fungoval, musí si vytvořit frontu u sebe
 - musí tedy posílat pakety hardwaru pomaleji, než je hardware odesílá
- Tento princip na Wi-Fi nemůže fungovat – je nutno najít úplně jiné řešení!



Wi-Fi hardware



- Ovladač přijaté pakety „zabalí“ a vloží do fronty hardwaru
 - v RAM, ale hardware si rámce bere sám od sebe přes DMA
 - pouze oznamuje odeslání přerušením
 - pořadí vložených paketů nelze měnit
 - u hardwaru podporujícího 802.11e je front několik
 - alespoň 4, u Atherosu 10
 - fronta hardwaru ale může být krátká
 - alespoň dva rámce: jeden se odesílá, druhý je připraven
 - po oznámení o odeslání přidám další
- Řešení se neobejde bez úpravy ovladače hardwaru
 - mám-li router a hardwarové AP, musím měnit firmware AP, žádná úprava na routeru nepomůže
 - ovladač musí být open-source
 - Ovislink-5460, WRT-314 a WA-2204 mají jen binární ovladač!



- Ovladač přijatý paket místo do fronty hardwaru vloží do zvláštní fronty pro každého klienta
- Dokončí-li hardware přenos, vybere se paket některého klienta a přesune se do fronty hardwaru
- Vypadá to jednoduše, ale...
 - Jak vybírat klienta?
 - rozlišení Sosače, Hráče a člověka, co jen prohlíží WWW
 - Jaký hardware zvolit?
 - Realtek nemá open-source ovladače, zbývá Atheros
 - Jaký ovladač zvolit?
 - Ath5k padá, kód Madwifi je strašný, ale alespoň funguje
 - Bude to dost rychlé?
 - HWAP mají jen okolo 16MB RAM a 150MHz MIPS CPU

Algoritmus řízení toku



- Navržen speciálně pro Wi-Fi
 - pracuje na 2. vrstvě, rozlišuje klienty podle MAC adres
 - nepočítá bajty, ale čas, na který klient zabral pásmo
 - sčítá u klienta upload i download
- Funguje automaticky
 - klienty hodnotí podle toho, kolik přenáší dat
 - nepotřebuje rozlišovat typy přenosů, je síťově neutrální
- Minimalizuje latenci
 - přijde-li paket pro prioritního klienta, bude čekat nanejvýš na odeslání dvou paketů
- Hodnotí klienty dlouhodobě
 - exponenciální průměr zatížení pásma za den i více
- Parametry jsou konfigurovatelné

Výsledek simulace



- Simulace chování algoritmu v prostředí OMNeT++

- Bez řízení toku

```
sent: 2000    drop rate (%) : 1.25
round-trip min/avg/max (ms) :
4.41913/1813.74/2676.66
stddev (ms) : 490.911    variance:0.240994
```

- S řízením toku

```
sent: 2000    drop rate (%) : 0
round-trip min/avg/max (ms) :
4.41913/56.8607/263.872
stddev (ms) : 29.3966    variance:0.000864162
```

- Funguje to: maximální latence je 10x menší,
průměrná dokonce 30x menší, packetloss zmizel

Skutečná implementace



- Zatím jen částečná, v ovladači chipsetu Atheros
- FreeBSD 7.1, Madwifi 0.9.4
- Madwifi 0.9.4 + nový HAL v OpenWRT pro AR531X
 - Ovislink WLA-5000 (otestováno)
 - Airca8-PRO
 - Nanostation2, Nanostation5
 - Bullet2, Bullet5





- Algoritmus nezasahuje do přístupové metody CSMA/CA
 - to by vyžadovalo úpravu software klientů
- Nemůže tedy snížit latenci CSMA/CA
 - AP musí čekat, než klient dokončí upload
 - klientů je několik, AP jen jedno, ale o pásmo se dělí rovným dílem
 - latence CSMA/CA stále může dosáhnout stovek ms
 - 802.11e nepomůže, prioritu určují klienti, ne AP
- Další postup je logicky úprava přístupové metody
 - Je to možné na stávajícím hardwaru?
 - u Atherosu je CSMA/CA implementována hardwarově
 - Může to být zpětně kompatibilní se stávajícími klienty?