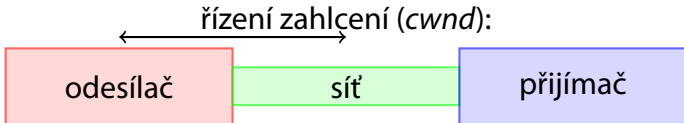
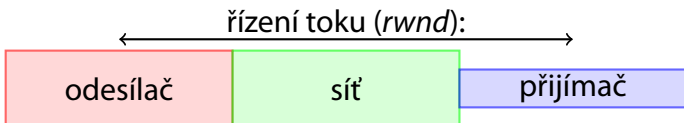
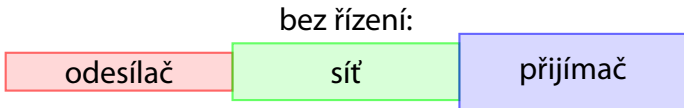


Tradiční TCP

- řízení toku (flow control) vs. řízení zahlcení (congestion control)



Tradiční TCP

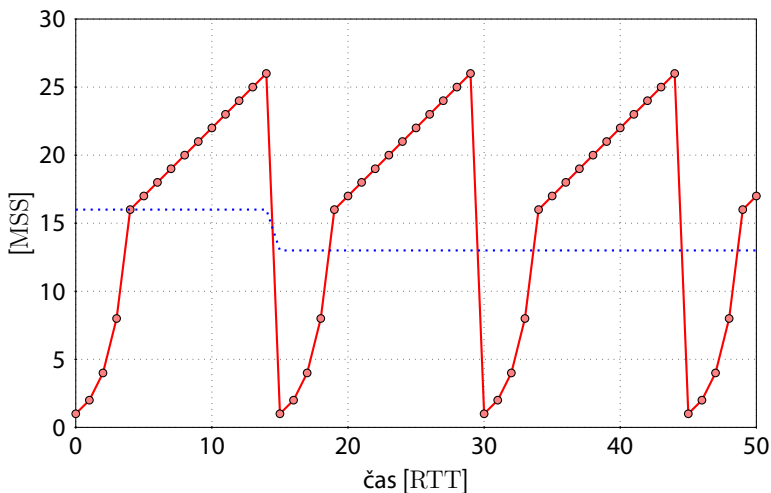
- Řízení toku
 - explicitní zpětná vazba od příjemce pomocí *rwnd*
 - deterministické
- Řízení zahlcení
 - přibližný odhad pomocí odesílatelem určovaného *cwnd*
- Finální výslední výstupní okno *ownd*

$$ownd = \min\{rwnd, cwnd\}$$

Použitá šířka pásma *bw* je pak

$$bw = \frac{8 \cdot ownd \cdot MTU}{RTT} \quad (1)$$

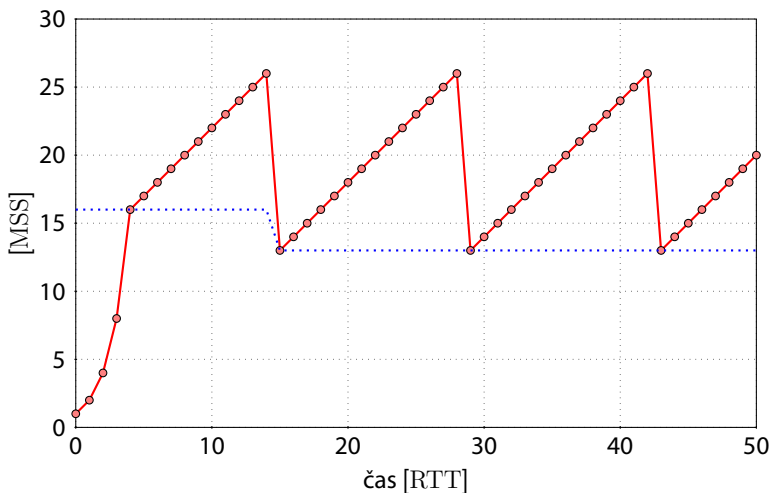
Tradiční TCP – Tahoe



○—○ *cwnd*

⋯ *ssthresh*

Tradiční TCP – Reno



○—○ *cwnd*

⋯ *ssthresh*

Tradiční TCP

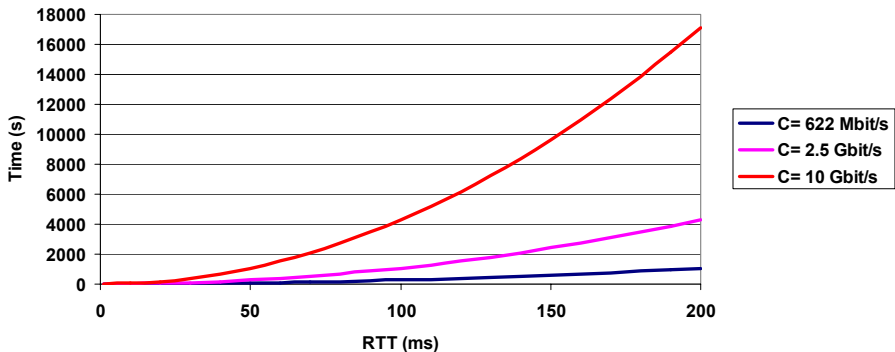
- Reakce na ztrátu dat – přeoslání
 - Tahoe: celé současné okno *ownd*
 - Reno: jeden segment v režimu Fast Retransmission
 - NewReno: více segmentů v režimu Fast Retransmission
 - Selective Acknowledgement (SACK): pouze ztracené pakety
- Základní otázka:

*Jak dosáhnout za realistických podmínek dostatečně velké *cwnd* na síti s velkým součinem kapacity a RTT?*

...a jak přitom neznemožnit přístup k síťové kapacitě pro „běžné“ uživatele?

Tradiční TCP – Responsivnost

TCP responsiveness

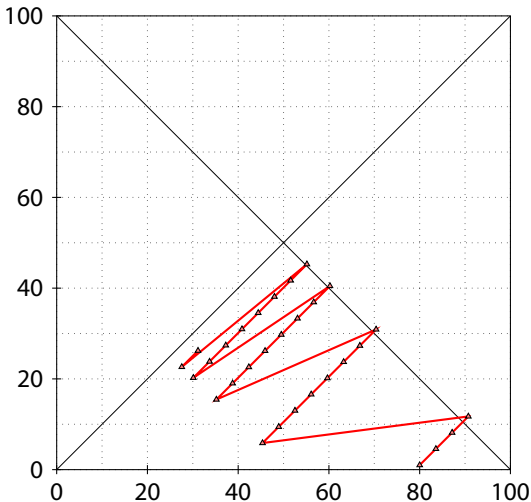


Tradiční TCP – Férovost

- Férovost v rovnovážném stavu
- Posuzování férovosti
 - pro proudy s různou RTT
 - pro proudy s různou MTU
- Podstatná je také rychlost konvergence do rovnovážného stavu!

Tradiční TCP – Férovost

- $cwnd \ += \text{MSS}, cwnd \ *= \ 0,5$



Ladění implementace TCP

- Spolupráce s HW
 - Rx/Tx TCP Checksum Offloading
 - běžně dostupné (ale někdy obsahuje chyby)
- Zero copy
 - přístup k síti obvykle zahrnuje několik kopií dat: user-land ↔ kernel ↔ síťová karta
 - page flipping – přesun user-land ↔ kernel
 - podpora např. pro `sendfile()`
 - implementace pro Linux, FreeBSD, Solaris, ...

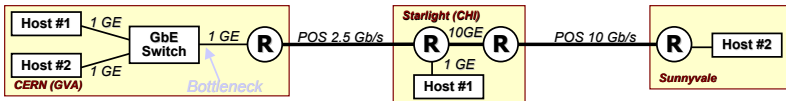
Ladění implementace TCP

- Web100 [4, 5]
 - instrumentace TCP/IP stacku pro Linux – TCP Kernel Instrumentation Set (TCP-KIS)
 - více jak 125 „táhel“
 - informace jsou dostupné přes `/proc`
 - knihovna pro přístup k instrumentaci
 - klientské nástroje v uživatelském prostoru (command-line, GUI)
 - monitoring
 - ladění parametrů
 - podpora pro auto-tuning

GridDT

- sbírka ad-hoc modifikací :(
- korekce *sstresh*
 - rychlejší slowstart
- modifikace AIMD řízení zahlcení
 - $cwnd = cwnd + a$
...pro úspěšné RTT
 - $cwnd = b cwnd$
...pro výpadek
- modifikace pouze na straně odesílače

GridDT – příklad



TCP Reno performance (see slide #8):

First stream GVA <-> Sunnyvale : RTT = 181 ms ; Avg. throughput over a period of 7000s = 202 Mb/s

Second stream GVA <-> CHI : RTT = 117 ms; Avg. throughput over a period of 7000s = 514 Mb/s

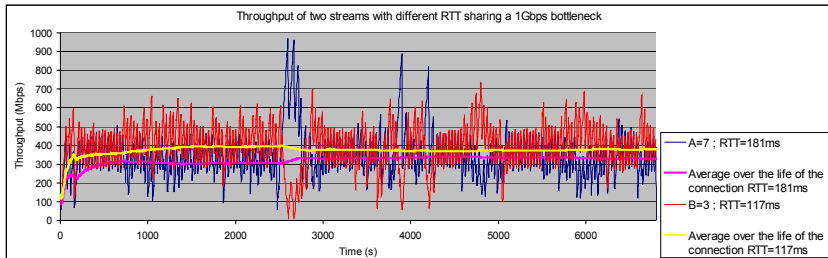
Links utilization 71,6%

Grid DT tuning in order to improve fairness between two TCP streams with different RTT:

First stream GVA <-> Sunnyvale : RTT = 181 ms, Additive increment = A = 7 ; Average throughput = 330 Mb/s

Second stream GVA <-> CHI : RTT = 117 ms, Additive increment = B = 3 ; Average throughput = 388 Mb/s

Links utilization 71.8%



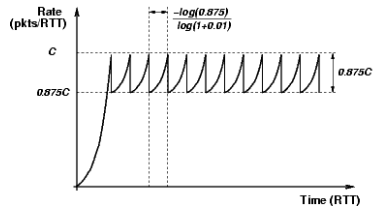
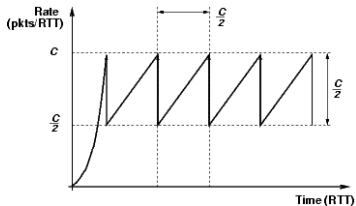
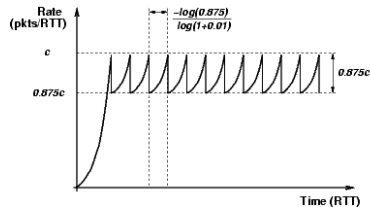
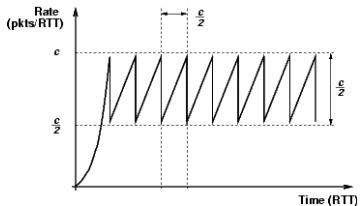
Scalable TCP

- navrhl Tom Kelly [1]
- řízení zahlcení již není AIMD:
 - $cwnd = cwnd + 0,01 cwnd$
...pro úspěšné RTT
 $cwnd = cwnd + 0,01$
...per-ACK
 - $cwnd = 0,875 cwnd$
...pro výpadek

⇒ Multiplicative Increase Multiplicative Decrease (MIMD)

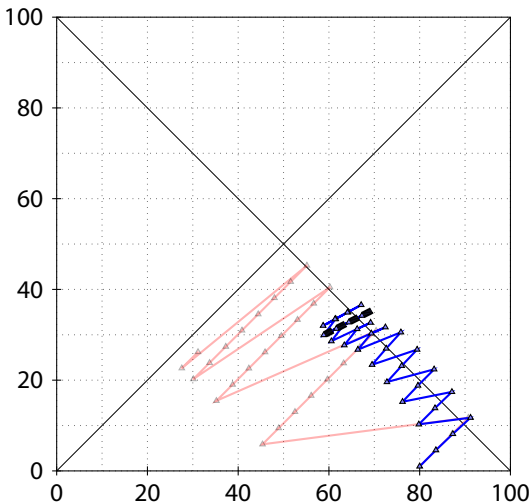
- pro malé velikosti okna a/nebo větší množství ztrát v síti se přepíná od AIMD režimu

Scalable TCP



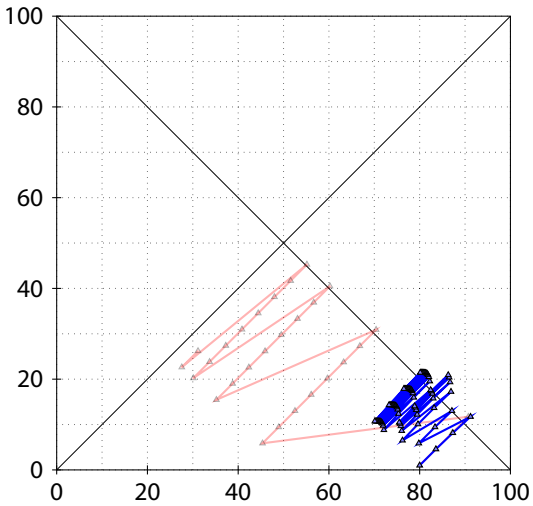
Scalable TCP – férovost

dva soutěžící Scalable TCP proudy, přepnutí na Scalable řízení
@ >30Mb/s, dvojnásobek kroků oproti předchozím simulacím

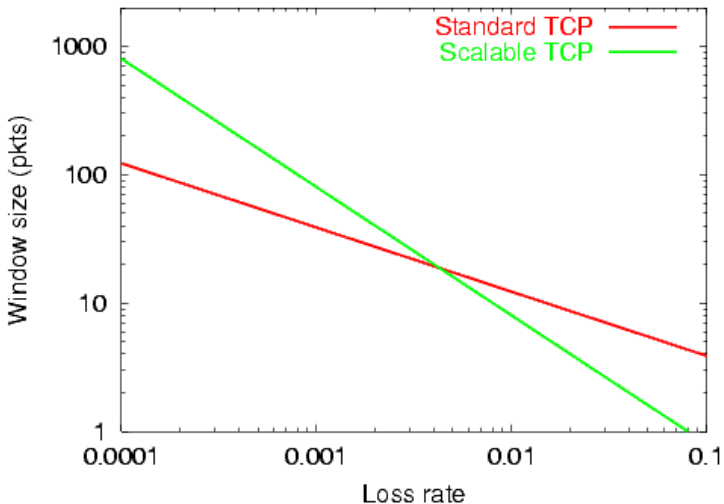


Scalable TCP – férovost

soutěžící Scalable TCP a tradiční TCP proudy, přepnutí na Scalable řízení @ >30Mb/s, dvojnásobek kroků



Scalable TCP – response curve



High-Speed TCP (HSTCP)

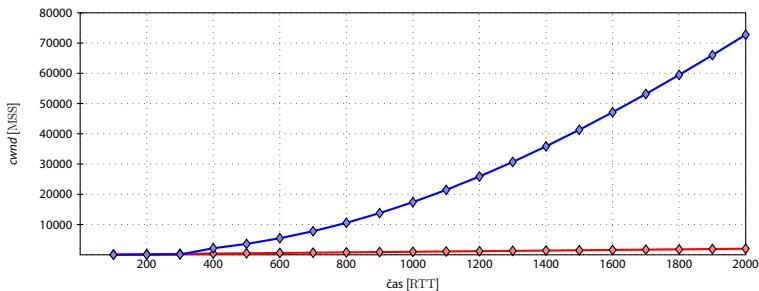
- Sally Floyd, RFC3649, [2]
- řízení zahlcení AIMD/MIMD:
 - $cwnd = cwnd + a(cwnd)$
...pro úspěšné RTT
 $cwnd = cwnd + \frac{a(cwnd)}{cwnd}$
...per-ACK
 - $cwnd = b(cwnd) cwnd$
...pro výpadek
- emuluje chování tradičního TCP pro malé velikosti okna a/nebo větší množství ztrát v síti

High-Speed TCP (HSTCP)

- navržená parametrizace MIMD:

$$b(cwnd) = \frac{-0,4(\log(cwnd) - 3,64)}{7,69} + 0,5$$

$$a(cwnd) = \frac{2cwnd^2 b(cwnd)}{12,8(2 - b(cwnd))w^{1,2}}$$



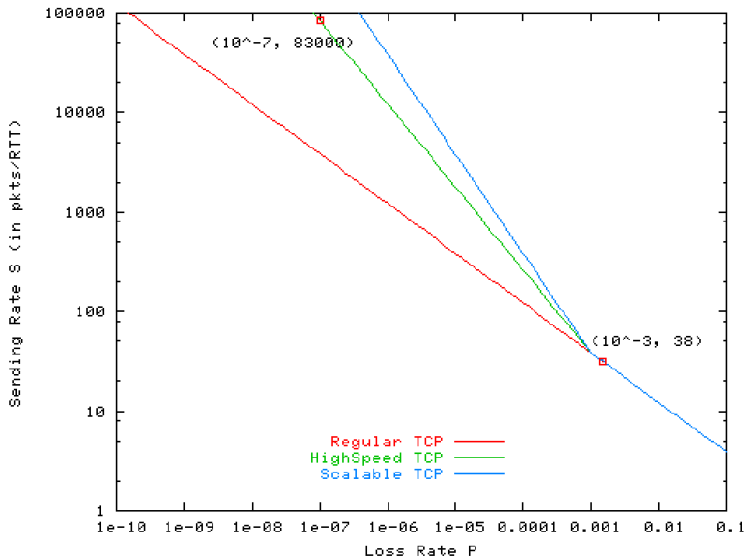
High-Speed TCP (HSTCP)

- možná parametrizace ekvivalentní Scalable TCP: Linear HSTCP
- porovnání s víceproudovým TCP

$$N(cwnd) \approx 0,23cwnd^{0,4}$$

- ani Scalable TCP ani HSTCP neřeší nijak sofistikovaně fázi slow-start

HSTCP – response curve



H-TCP

- Δ ... čas uplynulý od minulého výpadku
- přírůstek $cwnd$ závisí na Δ jakožto indikaci (šířka pásma \times zpoždění) a také na RTT, aby se kompenzovala neférovost mezi toky s různým RTT
- Δ_L ... pro $\Delta \leq \Delta_L$ se používá TCP nárůst
- Δ_B ... hranice změny dostupné šířky pásma, nad níž se používá TCP pokles (pro velké změny dostupné šířky pásma se používá TCP pokles 0,5)
- T_{min}, T_{max} ... minimální resp. maximální změřené RTT
- $B(k + 1)$... měření maximální propustnosti za poslední interval bez výpadku

H-TCP

- $cwnd = cwnd + \frac{2(1-\beta) a(\Delta)}{cwnd}$
...per-ACK
- $cwnd = b(B) cwnd$
...pro výpadek

$$a(\Delta) = \begin{cases} 1 & \Delta \leq \Delta_L \\ \max\{a'(\Delta)T_{min}; 1\} & \Delta > \Delta_L \end{cases}$$

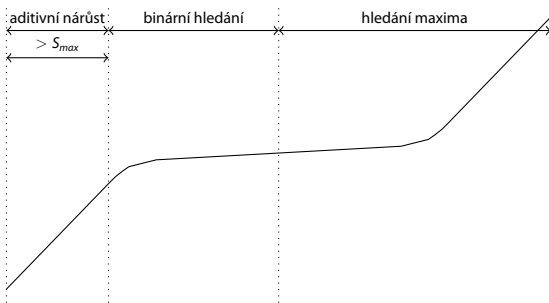
$$b(B) = \begin{cases} 0,5 & \left| \frac{B(k+1)-B(k)}{B(k)} \right| > \Delta_B \\ \min\left\{\frac{T_{min}}{T_{max}}; 0,8\right\} & \text{v opačném případě} \end{cases}$$

$$a'(\Delta) = 1 + 10(\Delta - \Delta_L) + 0,25(\Delta - \Delta_L)^2$$

...kvadratická přírůstková funkce

BIC-TCP

- K aktualizaci *cwnd* používá binární prohledávací algoritmus [3]
- 4 fáze fungování
 - (1) reakce na výpadek
 - (2) aditivní nárůst
 - (3) binární prohledávání
 - (4) hledání maxima



BIC-TCP

(1) Výpadek

- redukce okna
- původní okno $\rightarrow W_{max}$
- redukované okno $\rightarrow W_{min}$
- \implies protože k výpadku došlo při $cwnd \leq W_{max}$, budeme rovnovážné $cwnd$ hledat v intervalu $\langle W_{min}; W_{max} \rangle$

(2) Aditivní nárůst

- začít hledání od $cwnd = \frac{W_{min} + W_{max}}{2}$ by mohlo být pro síť příliš náročné
- pokud $\frac{W_{min} + W_{max}}{2} > W_{min} + S_{max}$, postupujeme aditivním nárůstem o konstantu $cwnd = W_{min} + S_{max}$

BIC-TCP

(3) Binární hledání

- $cwnd = \frac{W_{min} + W_{max}}{2}$
- pokud předchozí bod (i aditivní nárůst) prošel bez výpadku, $W_{min} = cwnd$, v opačném případě $W_{max} = cwnd$
- hledání pokračuje, pokud změna $cwnd$ není menší než konstanta S_{min} , kdy se nastaví $cwnd = W_{max}$
- výsledkem bodů (2) a (3) je obvykle lineární růst (aditivní nárůst), který se mění na logaritmický (binární hledání)

BIC-TCP

(4) Hledání maxima

- inverzní proces k bodům (3) a (2)
- nejdříve inverzní binární hledání, dokud nárůst není větší jako S_{max}
- lineární nárůst o velký inkrement po překročení předchozího bodu
- očekávané výhody
 - „přátelskost“ vůči TCP
 - během platu (3) mají TCP toky šanci „dorůst“
 - AIMD chování (byť rychlejší) ve fázích (2) a (4)
 - stabilnější velikost okna \implies lepší využití sítě
 - většinu času by BIC-TCP mělo trávit v platu (3)

Quickstart (QS)/Limited Slowstart

- existuje silné podezření, že slow-start fáze se nedá vylepšit bez interakce s níže položenými síťovými vrstvami
- návrh: 4-byte option v IP hlavičce, který zahrnuje pole QS TTL a Initial Rate
- odesílač, který chce použít QS, nastaví QS TTL na náhodnou hodnotu a Initial Rate na požadovanou rychlost, kterou chce začít vysílat, a pošle SYN paket

Quickstart (QS)/Limited Slowstart

- všechny routery po cestě, které podporují QS, sníží QS TTL o jedničku a sníží Initial Rate, pokud je to potřeba
- Příjímač pošle pole QS TTL a Initial Rate v SYN/ACK paketu odesílači
- Odesílač ví, jestli všechny směrovače po cestě podporují QS (porovnáním QS TTL a TTL)
- Odesílač si nastaví příslušné *cwnd* a začne používat svůj mechanismus řízení zahlcení (např. AIMD)
- **Vyžaduje změny v IP vrstvě! :-)**

E-TCP

- Early Congestion Notification (ECN)
 - součást Advanced Queue Management (AQM)
 - bit, který nastavují routery, když se blíží linky/buffery/fronty zahlcení
 - ECN příznak musí být odzrcadlen příjemcem
 - na ECN bit má TCP zareagovat stejně jako na výpadek
 - problém s tím, aby správci routerů konfigurovali AQM/ECN :-)

E-TCP

- E-TCP
 - navrhuje odzrcadlit ECN bit jen jednou (poprvé)
 - zamrzne *cwnd* když dorazí od přijímače ACK s nastaveným ECN bitem
 - vyžaduje (umělé) zavedení malých náhodných výpadků do sítě, aby se zajistil multiplikativní pokles kvůli férovému chování v čase
 - vyžaduje změnu chování k ECN bitu na přijímačích :-)

FAST

- Fast AQM Scalable TCP (FAST) [5]
- používá end-to-end delay, ECN a ztráty paketů pro detekci/vyhýbání se zahlcení
- T_{min} , T' ... minimální a průměrný pozorovaný RTT
- T_q ... odhad zpoždění front u RTT
- $f'_\alpha(B)$... (8, 20, 200) pro bw (< 10 Mb/s, 10 – 100 Mb/s, > 100 Mb/s), lze měnit přes `sysctl()`
- γ ... parametr návrhu ;-)

$$\text{ACK: } cwnd = \min \left\{ 2 \times cwnd, (1 - \gamma)cwnd + \gamma \left[\frac{T_{min}}{T'} cwnd + f_\alpha(B, T_q) \right] \right\}$$

výpadek: $cwnd = 0,5 cwnd$

$$f_\alpha(B, T_q) = \begin{cases} a \times cwnd & T_q = 0 \\ f'_\alpha(B) & T_q \neq 0 \end{cases}$$

tsunami

- TCP spojení pro out-of-band řídicí kanál
 - vyjednávání parametrů přenosu
 - požadavky na retransmisi – používá NACKy místo ACKů
 - vyjednávání ukončení přenosu
- UDP kanál pro přenos dat
 - řízení zahlcení MIMD
 - vysoce konfigurovatelné
 - parametry MIMD, nastavení prahu chyb, maximální velikost fronty pro retransmisi, interval zasílání požadavků na retransmisi

Reliable Blast UDP – RBUDP

- podobné jako tsunami – out-of-band TCP kanál pro řízení, UDP pro přenos
- vytvořeno pro přenosy z disku na disk, příp. takové, kde kompletní přenášená data lze udržet v paměti vysílače
- posílá data uživatelem definovanou rychlostí
 - `app_perf` (klon `iperfu`) pro odhad kapacity sítě a přijímače

XCP

- zpětná vazba od směrovačů per paket



Congestion Window = Congestion Window + Feedback

Jiné přístupy

- SCTP
 - víceproudový, multi-homed transport
 - <http://www.sctp.org/>
- DCCP
 - nezajištěný protokol (UDP) s řízením zahlcení kompatibilním s TCP
 - <http://www.ietf.org/html.charters/dccp-charter.html>
 - <http://www.icir.org/kohler/dcp/>

Jiné přístupy

- STP
 - založený na CTS/RTS
 - jednoduchý protokol pro snadnou implementaci v HW
 - bez sofistikovaného řízení zahlcení
 - <http://lwn.net/2001/features/OLS/pdf/pdf/stlinux.pdf>
- Reliable UDP
 - zajišťuje spolehlivé, in-order doručení (do maximálního počtu opakování retransmise)
 - RFC908 a RFC1151
 - původně vzniklo kvůli IP telefonii
 - konfigurace parametrů spojení per-spojení
 - <http://www.javvin.com/protocolRUDP.html>
- XTP (Xpress Transfer Protocol), ...

Závěrečné poznámky

- Současný stav
 - víceproudové TCP se intenzivně používá (např. Gridové aplikace)
 - hledání cesty, jak bezpečně (zpětně kompatibilně) zajistit vývoj/nasazení post-TCP protokolů
 - používání agresivních protokolů na privátních/dedikovaných sítích a okruzích (např. λ -sítě CzechLight/CESNET2, SurfNet, CaNET*4, ...)
 - implementace SCTP ve FreeBSD 7.0
 - implementace DCCP v Linuxu

Závěrečné poznámky

- Interakce s L3 (IP)
- Interakce se linkovou vrstvou
 - proměnné zpoždění a propustnost u bezdrátových sítí
 - optical burst switching
- Specifické per-flow stavy ve směrovačích
 - např. per-flow nastavení generovaných výpadků (→ E-TCP)
 - může pomoci krátkým tokům s vysokými přenosovými nároky (makro-bursty)
 - problém se škálovatelností a náklady :-)

Literatura



Jacobson V. "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM'88 (Stanford, CA, Aug. 1988), pp. 314–329.

<ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z>



Allman M., Paxson V., Stevens W. "TCP Congestion Control", RFC2581, Apr. 1999.

<http://www.rfc-editor.org/rfc/rfc2581.txt>



Brakmo L., Peterson L. "TCP Vegas: End to End Congestion Avoidance on a Global Internet", IEEE Journal of Selected Areas in Communication, Vol. 13, No. 8, pp. 1465–1480, Oct. 1995.

<ftp://ftp.cs.arizona.edu/xkernel/Papers/jsac.ps>



<http://www.web100.org>



Hacker T. J., Athey B. D., Sommerfield J. "Experiences Using Web100 for End-To-End Network Performance Tuning"

<http://www.web100.org/docs/ExperiencesUsingWeb100forHostTuning.pdf>

Literatura



Kelly T. "Scalable TCP: Improving Performance in Highspeed Wide Area Networks", PFLDnet 2003,
<http://datatag.web.cern.ch/datatag/pfldnet2003/papers/kelly.pdf>,
<http://www.lce.eng.cam.ac.uk/~ctk21/scalable/>



Floyd S. "HighSpeed TCP for Large Congestion Windows", 2003, <http://www.potaroo.net/ietf/all-ids/draft-floyd-tcp-highspeed-03.txt>



BIC-TCP, <http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/>



Floyd S., Allman M., Jain A., Sarolahti P. "Quick-Start for TCP and IP", 2006, <http://www.ietf.org/internet-drafts/draft-ietf-tsvwg-quickstart-02.txt>



Jin C., Wei D., Low S. H., Buhrmaster G., Bunn J., Choe D. H., Cottrell R. L. A., Doyle J. C., Newman H., Paganini F., Ravot S., Singh S. "FAST – Fast AQM Scalable TCP"
<http://netlab.caltech.edu/FAST/>
<http://netlab.caltech.edu/pub/papers/FAST-infocom2004.pdf>



tsunami, <http://www.anml.iu.edu/anmlresearch.html>



Zdroj: E. He, J. Leigh, O. Yu, T. A. DeFanti, "Reliable Blast UDP: Predictable High Performance Bulk Data Transfer," *IEEE Cluster Computing 2002*, Chicago, Illinois, Sept, 2002.

Další studijní materiály

- Workshopy PFLDnet 2003–2006
 - <http://datatag.web.cern.ch/datatag/pfldnet2003/program.html>
 - <http://www-didc.lbl.gov/PFLDnet2004/>
 - <http://www.ens-lyon.fr/LIP/RES0/pfldnet2005/>
 - <http://www.hpcc.jp/pfldnet2006/>
 - <http://wil.cs.caltech.edu/pfldnet2007/>
- Strány s příspěvky prof. Sally Floyd
 - <http://www.icir.org/floyd/papers.html>
- RFC3426 – “General Architectural and Policy Considerations”

http://www.hamilton.ie/net/eval/results_HI2005.pdf