

Členění protokolů pro řízení zahlcení

▶ router-based

- ▶ strategie zvolena na základě explicitní zpětné vazby z routerů
- ▶ přidání podpory do existujících routerů ve světě je nepraktické
- ▶ **protokoly:** *XCP, VCP, ..*

▶ end-to-end

▶ loss-based CC

- ▶ ztráta paketu jako indikátor zahlcení
- ▶ zvětšení okna po každém *ACK*, zmenšení při ztrátě paketu
- ▶ **protokoly:** *HSTCP, STCP, BIC, CUBIC, HTCP, ..*

▶ delay-based CC

- ▶ zpoždění ve frontách routerů / *RTT* jako indikátor zahlcení
- ▶ vytěžování front routerů
- ▶ **protokoly:** *FAST, Vegas*

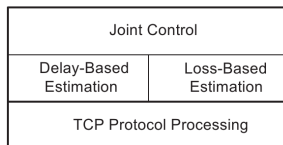
▶ loss-delay-based CC

- ▶ snaha o kombinaci předchozích přístupů
- ▶ někdy jeden přístup primární a druhý jako *fallback*
- ▶ **protokoly:** *CTCP, Illinois, EEFast, HCC*



Architektura

Z architektury znázorněné na obrázku 2 je patrné, že se protokol v podstatě skládá ze 3 komponent - komponenta odhadu na bázi zpoždění (**delay-based estimation**), komponenta odhadu na bázi ztrátovosti (**loss-based estimation**) a komponenta společné kontroly (**joint control**).



Obrázek 2: Architektura HCC

Značení

- ▶ α_i - počet paketů v rovnovážném stavu ve frontách routerů
- ▶ $w_i, w_i^a(t), \Delta w_i$ - velikost okna, resp. předpokl. velikost okna zdroje i v čase t , resp. rozdíl mezi aktuální a předpokl. velikostí okna
- ▶ D_i^f, d_i - dopředné jednocestné resp. propagační zpoždění zdroje i
- ▶ $baseD_i^f$ - minimální doposud pozorované D_i^f
- ▶ Q_i^f, Q_i^r - dopředné resp. zpětné zpoždění ve frontách zdroje i ,
 $Q_i^f = D_i^f - baseD_i^f$
- ▶ $avgQ_i^f(t), avgD_i(t)$ - průměrné hodnoty $Q_i^f(t)$ resp. RTT
- ▶ I_{min}, I_{max} - minimální resp. maximální změna velikosti okna, $I_{max} = \gamma\alpha_i$

Komponenty I.

Komponenta delay-based estimation

- ▶ využívá mechanismů *EEFAST* pro odhad zahlcení v síti
- ▶ rozšíření o nové kontrolní postupy řídicí velikost okna
- ▶ v rovnovážném stavu by mělo být ve frontách routerů α_i paketů bez negativního efektu provozu na zpětné cestě
- ▶ velikost okna lze zapsat jako $w_i = \frac{\alpha_i}{Q_i^f} D_i = \frac{\alpha_i}{Q_i^f} (Q_i^f + Q_i^r + d_i)$
- ▶ předpokládaná velikost okna v čase t odpovídá $w_i^a(t) = \frac{\alpha_i}{Q_i^f(t)} D_i(t) = \alpha_i \left(1 + \frac{Q_i^r(t) + d_i}{Q_i^f(t)} \right)$
- ▶ v iniciální fázi obvykle Q_i^f velmi malé a $w_i^a(t)$ naopak velmi vysoké (než $w_i^c(t)$)
- ▶ s rostoucí $w_i^c(t)$ roste také $D_i^f \Rightarrow$ se $w_i^a(t)$ snižuje



Komponenty II.

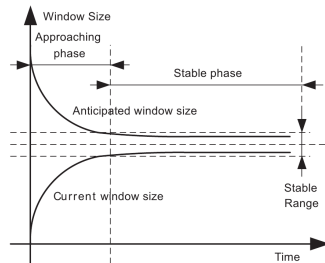
Pokud se vzdálenost mezi předpokl. a aktuální velikostí okna nachází v oblasti *stable range* (viz obr. 3), můžeme říct, že je plně využito dostupné kapacity sítě.

Každý zdroj si spočítá předpokl. velikost okna

$$w_i^a(t) = \frac{\alpha_i}{\text{avg}Q_i^f(t)} \text{avg}D_i(t).$$

Poté rozdíl Δw_i mezi předpokl. a aktuální velikostí okna $\Delta w_i(t) = w_i^a(t) - w_i^c(t)$.

Pokud je $\Delta w_i > 0$, pak by se mělo aktuální okno zvětšit. V opačném případě by mělo dojít ke zmenšení, které povede k přiblížení se k předpokl. oknu.



Obrázek 3: Změna velikosti okna na základě *delay-based* strategie

Komponenty III.

Před změnou velikosti okna musí dojít ke kontrole, zda změna velikost o Δw_i není větší než I_{max} . $|\Delta w_i| \geq I_{max} ? I_{max} : |\Delta w_i|$. Příliš velké změny by mohly zapříčinit výkyvy, které by měly za následek degradaci výkonu protokolu. Z předešlých výzkumů v oblasti protokolu *FAST TCP* bylo zjištěno, že ideální hodnota $I_{max} = \gamma \alpha_j$.

- ▶ protokol aktualizuje velikost okna jednou za *RTT*
- ▶ pozvolné přibližování k $w_i^a(t)$, velikost se mění pouze o $\Delta w_i/2$



Komponenty IV.

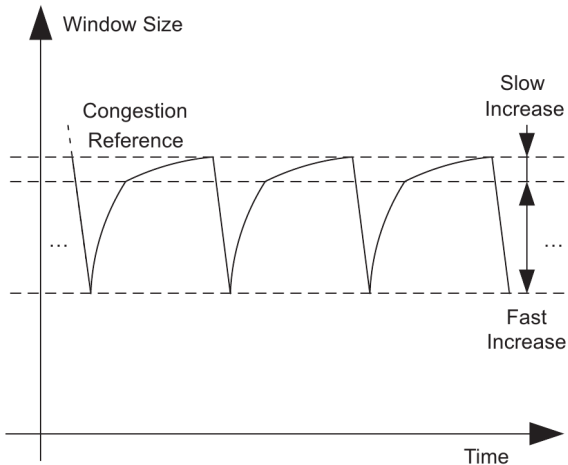
Komponenta loss-based estimation

Delay-based algoritmy vyžadují ve frontách routerů určený počet paketů, aby mohly udržovat průměrnou propustnost okolo maximálního možné hodnoty. Proto by velikost bufferů na routerech měla být větší než tento určený počet paketů. V opačném případě může docházet k výrazným ztrátám paketů v síti.

- ▶ rychlé získání volné kapacity sítě a prodloužení doby mezi po sobě následujícími zahlceními díky změnám velikosti okna v prim. komponentě (lineární → logaritmický růst - viz obr. 4)
- ▶ změna velikosti okna vzhledem k parametrům I_{max} a I_{min} stejně jako v primární komponentě
- ▶ bere také pouze polovina vypočítané hodnoty, která by měla určovat změnu okna



Komponenty V.



Obrázek 4: Lineárně → logaritmičticky se zvyšující schéma

Komponenty VI.

- ▶ při detekci ztraceného paketu dojde k zaznamenání aktuální velikosti okna (*loss-reference*) a k nastavení okna na polovinu
- ▶ poté fáze zotavení, kde $w_i^{max} = \text{loss-reference}$, $w_i^{min} =$ aktuální velikost okna a cílové zvětšení okna (w_i^{tar}) bude $w_i^{tar} = (w_i^{max} - w_i^{min}) / 2$
- ▶ není-li po navýšení okna ve fázi zotavení detekována ztráta, nastaví se w_i^{min} na akt. velikost okna a celý proces se opakuje
- ▶ je-li detekována ztráta, dojde k zaznamenání aktuální velikosti okna na *loss-reference* a w_i^{max}

Implementace

- ▶ při startu protokolu se pro odhad zahlcení na dopředné cestě nastaví *threshold* hodnota *mi_threshold*
- ▶ pokud $avgQ_i^f(t) < mi_threshold$, pak indikuje, že dopředné zpoždění ve frontách je nízké - lze použít schéma multiplikativního navýšení (MI)
- ▶ v opačném případě se předá řízení zahlcení primární komponentě
- ▶ dále se nastaví *loss-reference* na svou maximální (implicitní) hodnotu (dokud není detekována ztráta)



Variables:

W Current window size;

W_{old} Previous window size;

$w_i(t)$ Window size of source i , at time t ;

α_i Packet numbers that should be maintained at routers for source i ;

$mi_threshold$ Threshold value for forward congestion estimation.

Startup ():

Set *loss-reference* to be a default maximum value;

Start to calculate average queuing delay: $avgQ_i^f(t)$;

Start to calculate average RTT: $avgD_i(t)$;

If $avgQ_i^f(t) < mi_threshold$, then $W = 2 \cdot W_{old}$ per RTT;

If $avgQ_i^f(t) \geq mi_threshold$, then turn to **delay-based congestion control function ()**;

Delay-based congestion control function ():

Estimate anticipated window size (*delay-reference*):

$$w_i^d(t) = \alpha_i \cdot avgD_i(t) / avgQ_i^f(t);$$

Calculate the change for current window size: $\Delta w_i(t) = w_i^d(t) - w_i^c(t)$;

If $\Delta w_i(t) > I_{max}$, then $\Delta w_i(t) = I_{max}$;

If $\Delta w_i(t) < -I_{max}$, then $\Delta w_i(t) = -I_{max}$;

Update congestion window: $W = W_{old} + (\Delta w_i/2)$ per RTT;

Upon detected loss packet:

update *loss-reference*;

back off to $W = W/2$;

turn to **loss-based congestion control function ()**;

Loss-based congestion control function ():

Calculate increment for current window size:

$$w_i^{tar} = (w_i^{max} - w_i^{min}) / 2;$$

If $w_i^{tar} > I_{max}$, then $w_i^{tar} = I_{max}$;

If $w_i^{tar} < I_{min}$, then $w_i^{tar} = I_{min}$;

Update congestion window: $W = W_{old} + (w_i^{tar}/2)$ per RTT;

If $W > loss_reference$ & no packet loss is detected,

reset *loss-reference*;

turn to **delay-based congestion control function ()**;

Upon detected loss packet:

update *loss-reference*;

back off to $W = W/2$;

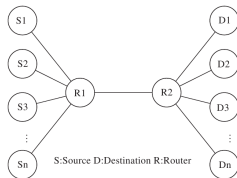
(a) část 1.

(b) část 2.

Obrázek 5: Algoritmus - pseudokód

Nastavení testovací prostředí

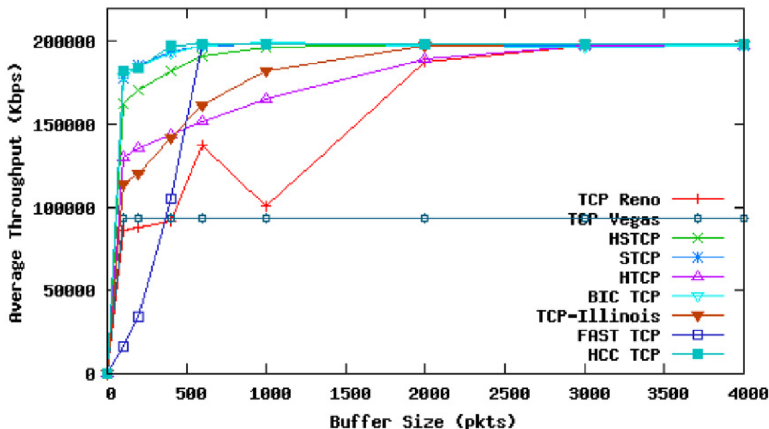
- ▶ simulace prostřednictvím *The Network Simulator (ns-2)*
- ▶ cílem porovnat a vyhodnotit vlastnosti *HCC TCP* s dalšími protokoly²
- ▶ zvolena topologie na obr. 6, jedna sdílená *200 Mbps* linka ($d_i = 20ms$)
- ▶ routery R1 i R2 využívají FIFO fronty, velikost datových paketů 1000 bajtů
- ▶ parametrizované protokoly (*FAST*, *HCC*) implicitně $\alpha = 400, \gamma = 0.5$
- ▶ standardní doba měření každé simulace je *300 s*



Obrázek 6: Síťová topologie zvolená k testování

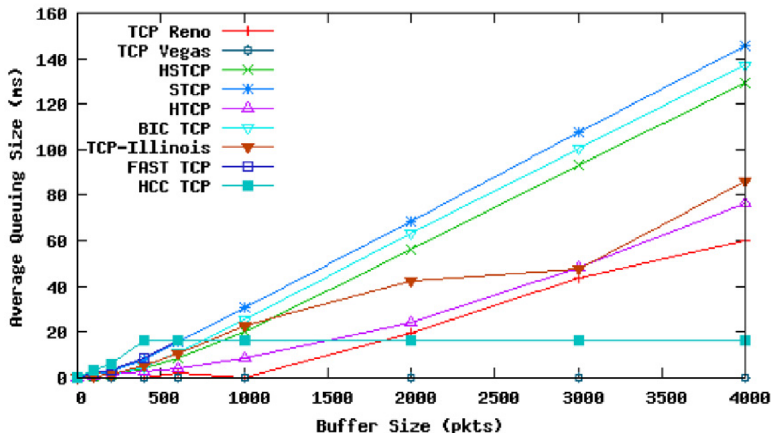
² *TCP Reno, TCP Vegas, HSTCP, STCP, HTCP, BIC-TCP, TCP-Illinois a FAST TCP*

Účinnost jednoho proudu



Obrázek 7: Průměrná propustnost vs. velikost bufferů

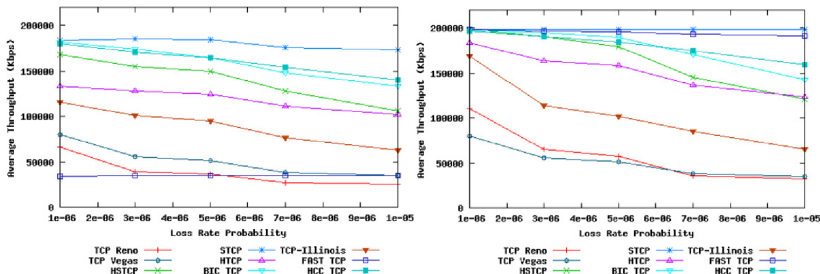
Účinnost jednoho proudu II.



Obrázek 8: Průměrná frontovací velikost vs. velikost bufferů



Účinnost jednoho proudu III.

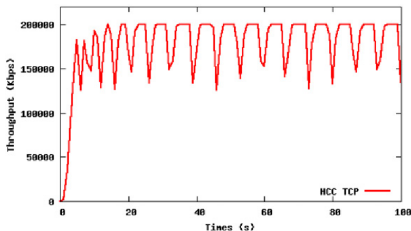


(a) velikost bufferu 200 paketů

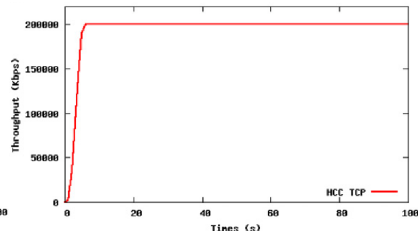
(b) velikost bufferu 2000 paketů

Obrázek 9: Průměrná propustnost vs. pravděpodobnost ztráty paketu

Účinnost jednoho proudu IV.



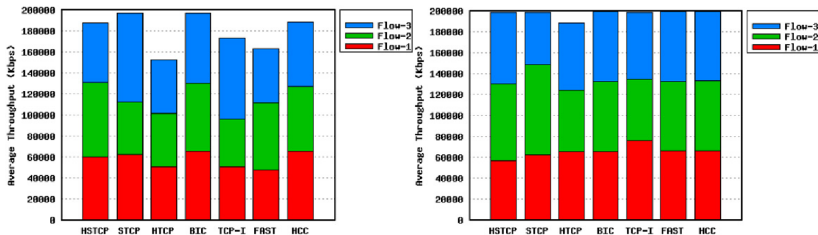
(a) velikost bufferu 200 paketů



(b) velikost bufferu 600 paketů

Obrázek 10: Dynamičnost *HCC TCP*

Fairness



(a) velikost bufferu 600 paketů

(b) velikost bufferu 2000 paketů

Obrázek 11: Průměrná propustnost 3 homogenních RTT (120 ms) přenosů

Jain's fairness index (**FI**), počet paralelních proudů n , průměrná prop. proudu $i = \bar{x}_i$

$$FI = \frac{(\sum_{i=1}^n \bar{x}_i)^2}{n \sum_{i=1}^n \bar{x}_i^2}$$

Fairness II.

Efficiency index (EI) odpovídá součtu propustností všech paralelních proudů

RTT ratio	1.5				2				6			
	T1	T2	EI	FI	T1	T2	EI	FI	T1	T2	EI	FI
HSTCP	197.91	1.68	199.59	0.5085	196.75	2.76	199.51	0.5140	198.57	0.66	199.23	0.5033
STCP	199.53	0.08	199.61	0.5004	199.24	0.29	199.53	0.5015	199.54	0.02	199.56	0.5001
HTCP	175.96	9.38	185.34	0.5532	184.64	6.09	190.73	0.5329	166.89	9.31	176.20	0.5556
BIC TCP	199.02	0.62	199.64	0.5031	199.00	0.62	199.62	0.5031	199.31	0.34	199.65	0.5017
TCP-Illinois	156.56	42.28	198.84	0.7517	173.75	23.15	196.90	0.6309	195.81	2.93	198.74	0.5150
FAST TCP	187.13	12.43	199.56	0.5661	190.11	9.44	199.55	0.5495	186.76	6.21	192.97	0.5332
HCC TCP	177.03	22.45	199.48	0.6248	179.19	20.26	199.45	0.6116	182.33	16.18	198.51	0.5880

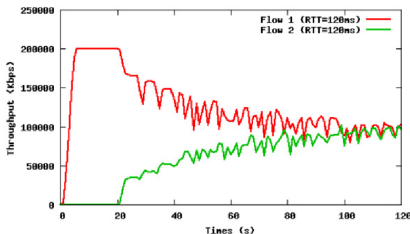
(a) velikost bufferu 600 paketů

RTT ratio	1.5				2				6			
	T1	T2	EI	FI	T1	T2	EI	FI	T1	T2	EI	FI
HSTCP	196.82	2.84	199.66	0.5144	199.35	0.23	199.58	0.5012	199.01	0.32	199.33	0.5016
STCP	198.74	0.93	199.67	0.5047	199.45	0.12	199.57	0.5006	197.69	1.26	198.95	0.5064
HTCP	189.87	9.79	199.66	0.5514	189.55	10.00	199.55	0.5526	168.39	30.28	198.67	0.6742
BIC TCP	199.33	0.31	199.64	0.5016	198.17	1.33	199.50	0.5067	199.03	0.63	199.66	0.5032
TCP-Illinois	130.19	69.27	199.46	0.9147	161.90	37.56	199.46	0.7201	168.30	31.13	199.43	0.6788
FAST TCP	100.18	99.40	199.58	1.0000	100.35	99.22	199.57	1.0000	102.77	96.77	199.54	0.9991
HCC TCP	99.94	99.55	199.49	1.0000	100.13	99.33	199.46	1.0000	103.31	96.11	199.42	0.9987

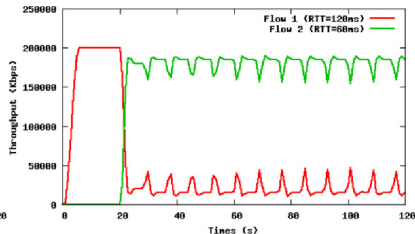
(b) velikost bufferu 2000 paketů

Obrázek 12: Výsledky měření indexů EI a FI

Fairness III.

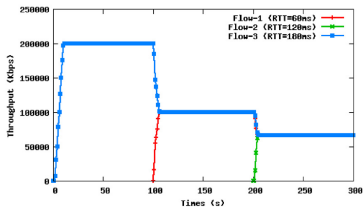


(a) 2 homogenní RTT přenosy

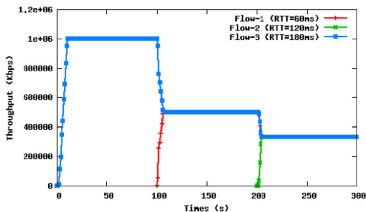


(b) 2 heterogenní RTT přenosy

Obrázek 13: Dynamičnost *HCC TCP* s velikostí bufferu 600 paketů

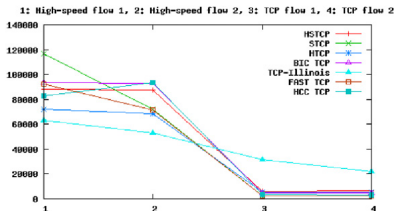


Obrázek 14: Dynamičnost *HCC TCP* s velikostí bufferu 2000 paketů

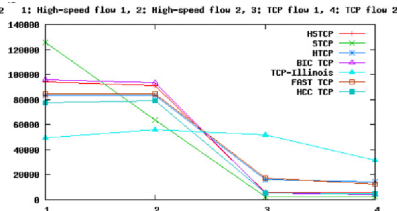


Obrázek 15: Dynamičnost *HCC TCP* na 1 Gbps síti

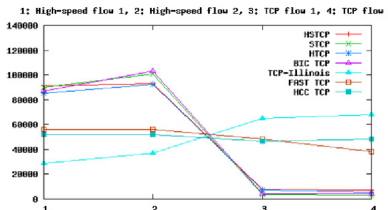
Výsledky testování



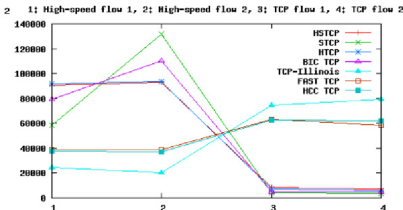
(a) velikost buffer 600 paketů



(b) velikost buffer 1000 paketů



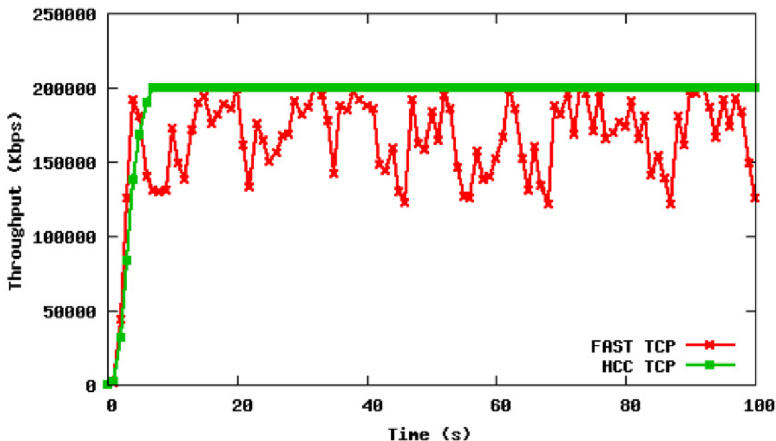
(c) velikost buffer 2000 paketů



(d) velikost buffer 3000 paketů

Obrázek 16: Průměrná prop. 2 high-speed TCP variant a 2 TCP Reno

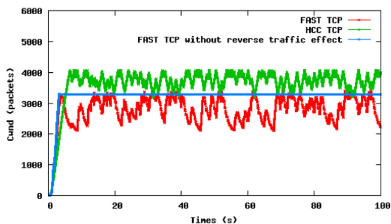
Robustness I.



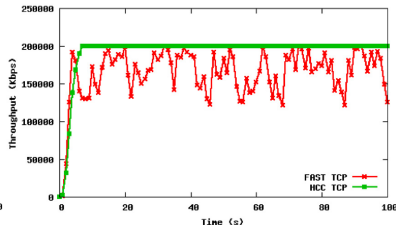
Obrázek 17: Průměrná prop. vs. velikost bufferu s přítomností provozu na zpětné cestě



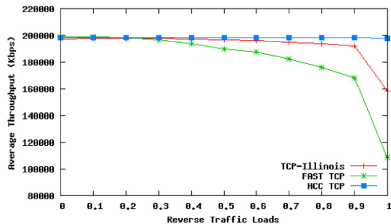
Výsledky testování



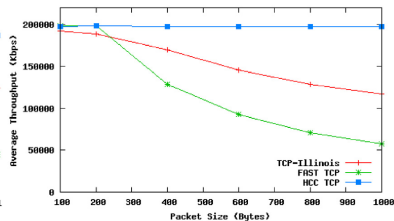
(a) porovnání křivek oken



(b) s provozem na zpětné cestě



(c) průměrná prop. vs. proměnná zátěž na zpětné cestě



(d) průměrná prop. vs. proměnná velikost paketů na zpětné cestě

