TCP VEGAS ALGORITHM FOR CONTROLLING THROUGHPUT AT THE TRANSPORT LAYER

Michal Pánek
Master Degree Programme (2), FEEC BUT
E-mail: xpanek01@feec.vutbr.cz

Supervised by: Jaroslav Koton
E-mail: koton@feec.vutbr.cz

Abstract: In order to easily send data between two end elements without congestion, methods that suitably control flow of date and evaluate possible overload state are necessary. One such method is to control the data flow directly on the transport layer. This layer offers a range of mechanisms dedicated to deal with this issue. The aim of this paper is focused on TCP Vegas for its ability to shape the data flow and the advantages and disadvantages of its use in the network. The paper also shows the behavior of two parallel data streams using TCP Vegas.

Keywords: TCP Vegas, transport layer, congestion control

1 ÚVOD

Uvádzaná práca sa zaoberá problematikou zahltenia na úrovni transportnej vrstvy za použitia mechanizmu TCP Vegas, ako hlavný prvok pri riadení dátového toku. TCP Vegas využíva momentálnych nameraných hodnôt dátového toku, podľa nich je schopný modifikovať okno zahltenia CWND (Congestion Window) [1]. Toto okno slúži na odosielanie zatiaľ nepotvrdených segmentov, a tak sa dá TCP Vegas na rozdiel od ostatných mechanizmov považovať za dynamický. V prípade vzniku zahltenia dôjde ku strate segmentov, tieto je potrebné znovu odoslať, a preto TCP Vegas rozšíril možnosti znovu odosielania stratených jednotiek. Práca sa preto snaží analyzovať TCP Vegas a zistiť jeho pozitívne a negatívne prínosy pri vzájomnej komunikácií dvoch prvkov.

2 MECHANIZMUS TCP VEGAS

Mechanizmus TCP Vegas (ďalej TCP-V) je rozsiahlou modifikáciou mechanizmu TCP Reno [2, 3]. Je založený na princípe aktívneho merania doby odozvy (RTT), ktoré je efektívnejšie pri identifikácií nadchádzajúceho sa zahltenia. Pomocou tohoto princípu TCP-V nepotrebuje vyčkávať na duplicitné potvrdenia aby zahájilo opatrenia pri vzniknutom zahltení. Kde Reno muselo počkať na duplikáty, sieť sa už nachádzala v stave zahltenia, TCP-V je schopné predísť zahlteniu skôr než k tomu dôjde.

2.1 POMALÝ ŠTART

Slúži ako základný mechanizmus pre nadviazanie spojenia a vlastný prenos dát. V počiatočnej fázy nemá TCP žiadne znalosti o parametroch siete, čiže nevie aká môže byť veľkosť CWND. Teoreticky by mohol TCP posielať na maxime čo by však viedlo k zahlteniu siete. Aby sa predišlo podobným problémom, používa sa mechanizmus pomalého štartu. TCP-V svoje okno CWND zvyšuje exponenciálne na každý ďalší RTT počas pomalého štartu, pričom pomalý štart opustí len v prípade kedy CWND prekročí limit siete, tu označovaný ako prah. Počas dvoch po sebe idúcich RTT sa okno CWND nemení, ostáva fixné, vďaka čomu je možné zmerať rozdiel medzi Očakávanou (Expected) a Aktuálnou (Actual) hodnotou priepustnosti [2].

2.2 MECHANIZMUS PRE OPÄTOVNÉ ODOSIELANIE

Mechanizmy ktoré používa TCP opätovne odosielajú segmenty v prípade vypršaní časovača alebo po prijatí troch duplicitných potvrdení. TCP-V tuto myšlienku prijal a rozšíril, vďaka tomu že je schopný zaznamenávať jednotlivé časy pre každý segment, individuálne pristupuje k potvrdeniam následovne [1, 2]:

- V prípade **duplicitného ACK**, TCP-V overí či je rozdiel, medzi aktuálnym časom a časom zaznamenaným pri odoslaní, väčší než je hodnota časového limitu. Pokiaľ áno, TCP-V opätovne odošle daný segment bez toho aby musel čakať na tri duplicitné potvrdenia.
- V prípade **neduplicitného ACK**, pokiaľ sa jedná o prvý alebo druhý prijatý ACK po opätovnom odoslaní, TCP-V znovu kontroluje či hodnota zaznamenaná pri odoslaní neprevyšuje hodnotu časového limitu. Pokiaľ áno, TCP-V znovu odošle príslušný segment. Tento spôsob poslúži v prípade ak by došlo ku strate aj počas samotného opätovného odosielania.

2.3 ALGORITMUS TCP VEGAS KU ZAHLTENIU

Prístup TCP-V k zahlteniu závisí na zmenách **Očakávanej** (Expected) hodnoty priepustnosti a **Aktuálnej** (Actual) hodnoty priepustnosti v určitom časovom úseku, pracuje následovným spôsobom [2]:

1. Je určená *BaseRTT* [s] ako minimálna nameraná hodnota RTT, ta sa zvolí na základe šírenia oneskorenia (ide o RTT segmentu, ktorý sa nenachádzal v preplnenom spojení). Následovne je vypočítaná **Očakávaná** (Expected) priepustnosť $[\mathrm{seg}\cdot\mathrm{s}^{-1}]$ a **Aktuálnu** (Actual) priepustnosť $[\mathrm{seg}\cdot\mathrm{s}^{-1}]$, ktorá sa počíta pre každý RTT:

$$Expected = \frac{cwnd}{BaseRTT}, \qquad (1) \qquad\qquad Actual = \frac{cwnd}{RTT}, \qquad (2)$$

   kde *cwnd* je momentálna hodnota okna zahltenia [seg] a RTT [s] je najnovšia odmeraná RTT.
2. Následuje porovnávanie **Očakávanej** (Expected) hodnoty priepustnosti a **Aktuálnej** (Actual) hodnoty priepustnosti označené tiež ako *diff* $[\mathrm{seg}\cdot\mathrm{s}^{-1}]$:

$$diff = Actual - Expected = \left(\frac{cwnd}{BaseRTT} - \frac{cwnd}{RTT}\right) BaseRTT = cwnd\left(1 - \frac{BaseRTT}{RTT}\right), \quad (3)$$

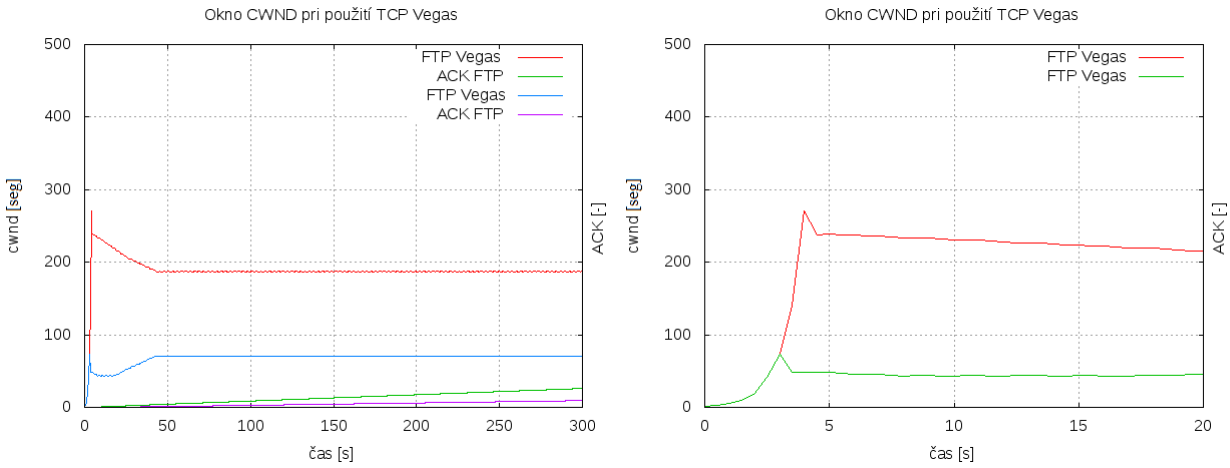
   ktorej hodnota je vždy nezáporná, a slúži k nastaveniu okna CWND.
3. TCP-V ma implementované dve konštanty α a β, podľa ktorých riadi CWND:

$$cwnd = \begin{cases} cwnd + 1 & diff < \alpha, \quad (4a) \\ cwnd & \alpha \leq diff \leq \beta, \quad (4b) \\ cwdn - 1 & diff > \beta. \quad (4c) \end{cases}$$

   Pokiaľ *diff*<α, TCP-V zväčší CWND lineárne počas ďalšieho RTT.
   Pokiaľ *diff*>β, TCP-V zníži CWND lineárne počas ďalšieho RTT.
   Pokiaľ $\alpha \leq diff \leq \beta$, TCP-V ponecháva CWND nezmenené.

3 POUŽITIE TCP VEGAS PRI PRENOSE PARALELNÝCH DÁTOVÝCH TOKOV

Pomocou metódy TCP Vegas, boli simulované dva dátové toky na šírke pásma o veľkosti 10 Mbit/s. V Obr. 1(a) je možné vidieť ovládanie okna CWND dvoch dátových tokov pomocou metody TCP Vegas. Pri počiatočnom prenose ktorý je možný vidieť na Obr. 1(b) v čase 0 s až 4 s, nastáva fáza pomalého štartu, kedy pri každom následujúcom RTT sa hodnota okna CWND zvyšuje exponenciálne do doby kým sa neprekročí prah.Obr. 1(a) zároveň ukazuje správanie sa okna CWND počas prenosu.



(a) (b)

Obrázek 1: (a) Graf okna CWND TCP Vegas, (b) graf okna CWND počiatočných 20 s

Dvojica okien sa správa rozdielne, miesto chaotického správania su rozdelené a ich hodnota okna CWND sa dá považovať za konštantnú. Tento spôsob úpravy okien vychádza z algoritmu metódy, kedy sa nepoužíva pre indikáciu zahltenia duplicitné potvrdenia, ale používa sa predikcia zahltenia pomocou vypočítanej hodnoty *diff*. Pretože je RTT rozdielny pre oba dátové toky výsledná hodnota *diff* bude rozdielna a tak okno CWND rôzne veľké. Aj keď sa hodnota RTT mení pre oba dátové toky, nieje kritická aby dochádzalo ku zmenam veľkosti okna a od času 47 s je splnená podmienka $\alpha \leq diff \leq \beta$ pre oba dátove toky podľa vzorca 4b, hodnota CWND nemení, nedochádza ku stratám a dané TCP spojenie je maximálne efektívne. Graf následovne ukazuje neefektívne rozdelenie dostupného zdroja kde je jeden dátový tok zvýhodnený a jeho hodnota okna CWND je vyššia.

4 ZÁVER

Práca popisuje správanie sa TCP Vegas, z uvedených výsledkov je možné vidieť, že sa samotné TCP Vegas snaží o minimálnu strátavosť a to však na úkor agresívnej správy okna CWND, čo ma za následok kedy sa dátový tok drží na nižšej úrovni a nemôže ísť k maximálnej hranici, ktorá je dovolená sieťou. Podstatnou nevýhodou TCP-V je neefektívne rozdelenie poskytnutých zdrojov, nedochádza k spravodlivému rozdeleniu prenosovej kapacity a to po celu dobu samotného prenosu.

POĎAKOVANIE

Tento príspevok vznikol za činnosti podporené vývojovým zázemím výskumného centra SIX.

REFERENCE

[1] BRAKMO, L.S., O'MALLEY, S.W., PETERSON, L.L., TCP Vegas: new techniques for congestion detection and avoidance, in *Proc. conf. Commun. arch., prot. and apps.*, London, United Kingdom, p.24-35, 1994.

[2] SHIHADA, B., ZHANG, Q., HO, P.-H., JUE, J.P., A Novel Implementation of TCP Vegas for Optical Burst Switched Networks, *Optical Switching and Networking*, vol. 7, no. 3, pp. 115-126, 2010.

[3] Doporučenie RFC 2582: *The NewReno Modification to TCP's Fast Recovery Algorithm* [online]. April 1999. dostupné online: `https://tools.ietf.org/html/rfc2582`