

Nagysebességű TCP protokollok együttműködésének modellezése

SIMON BOGLÁRKA, SONKOLY BALÁZS, MOLNÁR SÁNDOR

BME Távközlési és Médiainformatikai Tanszék, HSNLab
simonbogi@gmail.com, molnar@tmit.bme.hu

Lektorált

Kulcsszavak: HSTCP, STCP, transzport protokoll modellezés, fairness

A hagyományos TCP (Transmission Control Protocol) torlódásvezérlésében jelentkező problémák miatt nagysebességű és nagykiterjedésű hálózati környezethez a közelmúltban több új, nagysebességű TCP verziót fejlesztettek ki. Ilyen többek között a HighSpeed TCP és a Scalable TCP. Bár több kutatás foglalkozott teljesítményanalízissel, számos nyitott kérdés maradt még a működésükkel kapcsolatban. Nagyon fontos kérdés az, hogy a különböző protokollok mennyire képesek igazságos együttműködésre (fairness). Ebben a cikkben szabályozástechnikai modellezés alapú eredményeket ismertetünk. A hálózatot egy visszacsatolt rendszerként értelmezve vizsgáltuk a nagysebességű protokollok együttműködését. Megadtuk a hálózat elemeinek – TCP források, szűk sáv szélességű link és RED algoritmus – folyadékmodelljét. Az analitikusan nehezen kezelhető, bonyolult differenciálegyenlet-rendszerek numerikus approximációval történő megoldására terveztünk és implementáltunk egy MATLAB/Simulink környezetet és ebben vizsgáltuk a különböző folyamatok egymásra hatását. Eredményeinket Ns-2 szimulációkkal validáltuk. Ennek eredményeképp pontosabb és részletesebb tudásbázist hoztunk létre a nagysebességű protokollok alapvető tulajdonságairól, előnyeiről és hátrányairól.

1. Bevezetés

A TCP napjainkban a leggyakrabban használt végpontok közötti szállítási protokoll az Interneten, melynek robbanásszerű terjedése és gyorsan növekvő kihasználtsága miatt a hálózaton már a kezdeti időkben szükségessé vált a torlódások és a csomagvesztések elkerülése. Az általunk hagyományos TCP-nek tekintett TCP Reno protokoll torlódás megelőzési fázisában AIMD (Additive Increase Multiplicative Decrease) algoritmust használ. Ez az algoritmus eggyel növeli az ablakméretet (W) minden beérkező nyugta esetén és felezi azt, ha csomagvesztés történik. Ez a torlódásvezérlési algoritmus napjainkra nem nyújt hatékony működést nagysebességű, nagykiterjedésű hálózati környezetben, mert ablaknövelése túl lassú, míg az ablak méretének felezése torlódás esetén túl drasztikus megoldás. A hatékony működés érdekében a közelmúltban számos javaslat született, amelyek főként azt próbálják elérni, hogy a TCP torlódásvezérlési mechanizmusa rugalmasabb, dinamikusabb legyen és minél jobb hatásfokkal kihasználja a rendelkezésre álló kapacitást, sáv szélességet. Ezek közül az egyik a csomagvesztés alapú algoritmusok fejlesztése, mint például a HighSpeed TCP (HSTCP) [1] vagy a Scalable TCP [2], egy másik a késleltetés alapú algoritmusok csoportja, ilyen a FAST TCP.

Az új protokollok elterjedéséhez alapvető fontosságú a más protokollokkal való igazságos együttműködés. Erősen vitatott kérdés, hogy a kezdeti késleltetés változtatása milyen hatással lehet a versenyző nagysebességű protokollok hosszútávú együttműködésére [6]. Munkánk során nagysebességű TCP protokollok együttműködését vizsgáltuk a torlódásszabályozási algoritmusok modellezése alapján.

Vizsgálatainkat aktív sorkezelési mechanizmust alkalmazó hálózatokra végeztük. Fontos hangsúlyozni, hogy a vizsgálatok során a RED (Random Early Detection) algoritmust használtuk, nem az úgynevezett Drop Tail-t, így eredményeink eltérnek a Drop Tail-nél tapasztaltaktól. Szabályozásméleti terminológiát követve egy aktív sorkezelési mechanizmust alkalmazó TCP hálózat egyes komponensei egy visszacsatolt szabályozási kör egyes blokkjaival azonosíthatók [3-5].

Az egyes elemek működése, a használt algoritmusok és az egymásra hatások jól leírhatók analitikusan differenciálegyenlet-rendszerrel. A visszacsatolt rendszert leíró meglehetősen bonyolult differenciálegyenlet-rendszereket – melyek változó idejű késleltetéseket tartalmaznak bizonyos argumentumokban, illetve rekurzív összefüggéseket írnak le – implementáltuk egy általunk kialakított MATLAB/Simulink környezetben. Modelljeinket különböző hálózati elrendezések és beállítások mellett csomag szintű szimulációs vizsgálatokkal validáltuk (Ns-2) [7].

2. A HSTCP és Scalable TCP protokollok torlódásvezérlése

A HSTCP és a Scalable TCP újabb verziójú csomagvesztés alapú protokollok, melyek a rossz ablakdinamikán javítandó módosított AIMD-t használnak, melynek segítségével az ablakméret (W) növelése gyorsabb, míg torlódás esetén való csökkentése kevésbé drasztikus.

A Scalable TCP [2] módosított AIMD algoritmus az MIMD (Multiplicative Increase Multiplicative Decrease) jóval dinamikusabb működést eredményez, így az átvitel hamarabb felgyorsul, a linkek kihasználtsága pedig rövid időn belül eléri a maximális értéket. Az algoritmus

aW -vel növeli az ablakméretet 1 helyett (ahol a egy paraméter és W a torlódási ablak mérete) egy-egy nyugtára és nem felezi azt csomagvesztés esetén, hanem csupán az $1/8$ -ával csökkenti. Ez a legagresszívabb, legrobustusabb TCP protokoll. A Scalable TCP működését összevetve a hagyományos TCP-vel azt tapasztalhatjuk, hogy a Scalable TCP nagy sebességű hálózatokon jobb helyreállítási időket produkál és dinamikusabban képes kihasználni a hálózatok kapacitását. A protokollt úgy tervezték meg, hogy biztosítsa az erőforrások megosztását a linken, és mindezek mellett stabil és rugalmas legyen a hálózati körülményekkel szemben.

A HighSpeed TCP [1] szintén a TCP Reno torlódásvezérlési algoritmusára épül, csak jóval dinamikusabbra lett tervezve és ennek érdekében módosították az általa használt AIMD algoritmust. A protokoll egy bizonyos ablakméret (low_W) alatt a hagyományos TCP-nek megfelelően működik, tehát minden nyugtára eggyel növeli az ablakméretet és minden csomagdobásra felezi azt, low_W felett pedig két új paramétert (a -t és b -t) vezet be, ahol a az ablaknövelő, b az ablakcsökkentő paraméter. A hagyományos TCP-nél ez a két paraméter $a=1$ és $b=0,5$ volt. Az említett változók a HighSpeed TCP esetében több új paramétertől és a W aktuális értéktől függenek. Ennek köszönhetően a torlódási események gyakoriságának csökkenésével a HSTCP átviteli sebessége nagyobb mértékben növekszik, így adott átviteli sebességet több torlódási esemény mellett is el tud érni, szemben a hagyományos TCP-vel.

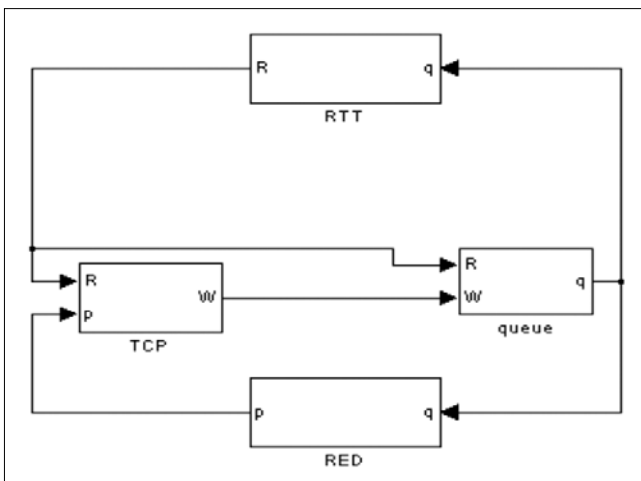
3. Vizsgálati környezet

Ebben a fejezetben bemutatjuk a folyamszintű vizsgálatokhoz kialakított Matlab/Simulink környezetet és ismertetjük a csomagszintű validáláshoz használt Ns-2 szimulációs környezetet.

a) MATLAB/Simulink környezet bemutatása

Az 1. ábra mutatja be az általunk kialakított MATLAB/Simulink környezet felső szintjét.

1. ábra A MATLAB/Simulink környezet felső szintje



A TCP működését vizsgáló folyadékmodell folyamszintű ("flow-level") vizsgálatokat tesz lehetővé. Az egyes jellemzők, változók dinamikus viselkedését a várható értékükkel írjuk le. A Simulink modell csak a torlódás megelőzési fázist modellezi, nem tartalmazza a kezdeti (slow start) fázist, mert az a protokoll működésében a mi vizsgálataink szempontjából elhanyagolható, hiszen nagyon rövid időt tesz ki.

A Scalable TCP (1-2) és a HighSpeed TCP (3-4) működését leíró differenciálegyenletek a következők:

$$\frac{dW}{dt} = \frac{aW(t)}{R(t)} - \frac{bW(t)W(t-R(t))}{R(t-R(t))} p(t-R(t)) \quad (1)$$

$$\frac{dq}{dt} = \frac{W(t)}{R(t)} N(t) - C \quad (2)$$

$$\frac{dW}{dt} = \frac{a(W(t))}{R(t)} - \frac{b(W(t))W(t)W(t-R(t))}{R(t-R(t))} p(t-R(t)) \quad (3)$$

$$\frac{dq}{dt} = \frac{W(t)}{R(t)} N(t) - C \quad (4)$$

ahol:

$W(t)$: az aktuális ablakméret

$q(t)$: aktuális sorhossz

$R(t)$: körfordulási idő (RTT)

C : link kapacitás

a : Scalable TCP ablaknövelő paramétere

b : Scalable TCP ablakcsökkentő paramétere

$a(W(t))$: HSTCP $W(t)$ -től függő

ablaknövelő paramétere

$b(W(t))$: HSTCP $W(t)$ -től függő

ablakcsökkentő paramétere

$N(t)$: TCP folyamok száma

$p(t)$: csomagelölési valószínűség

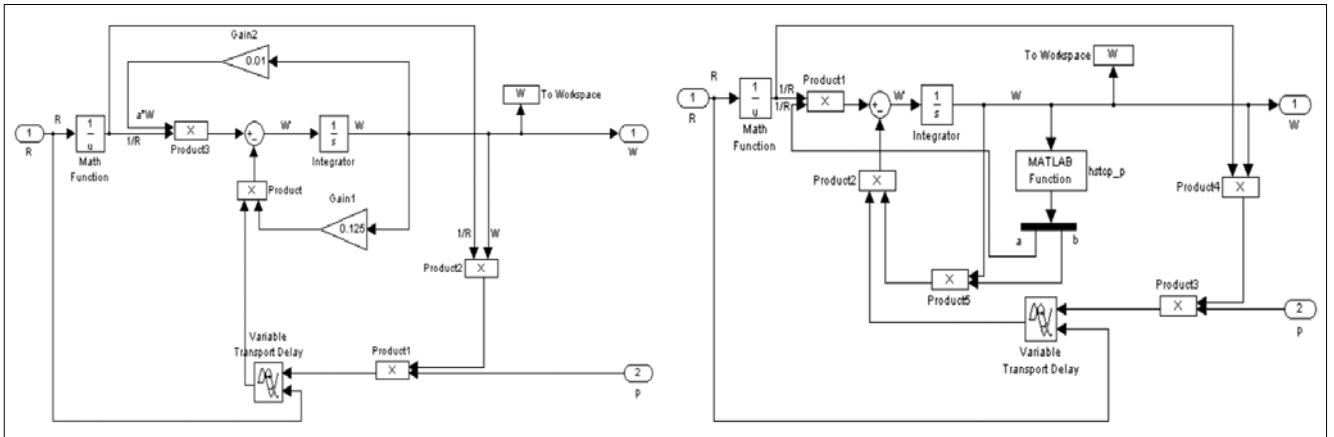
Az egyenletek közül az ablakméret szabályozását leíró egyenlet felépítése a következő. Az első rész valósítja meg az additív, illetve multiplikatív növekedési részt, mely növeli W -t minden RTT alatt a pozitív nyugták érkezési rátájával arányosan. A második (mínuszos) rész a multiplikatív csökkentési rész, mely a negatív nyugták (megjelölt csomagok) érkezési rátájával arányosan csökkenti a W -t.

A negatív nyugták érkezési rátája:

$$\frac{W(t-R(t))}{R(t-R(t))} p(t-R(t)).$$

A várakozási sor dinamikáját leíró egyenlet felépítése a következő. $W(t)/R(t)$ mutatja a sorba érkező csomagok rátáját, mely N folyam esetén természetesen N -szeres lesz. A kiszolgálás C kapacitás szerint történik.

Mivel az egyenletek rekurzív összefüggéseket és változó késleltetéseket tartalmaznak, analitikusan nehezen kezelhetők, ezért implementáltuk őket a MATLAB/Simulink környezetben, ahol megoldhatók különböző kezdeti feltételek és késleltetések mellett.



2. ábra Scalable TCP és HSTCP modelljének ablakméret-szabályozó blokkja

A fenti differenciálegyenletek közül az ablakméret-szabályozást leíró MATLAB/Simulink modelleket a 2. ábra mutatja.

b) Ns-2 környezet bemutatása

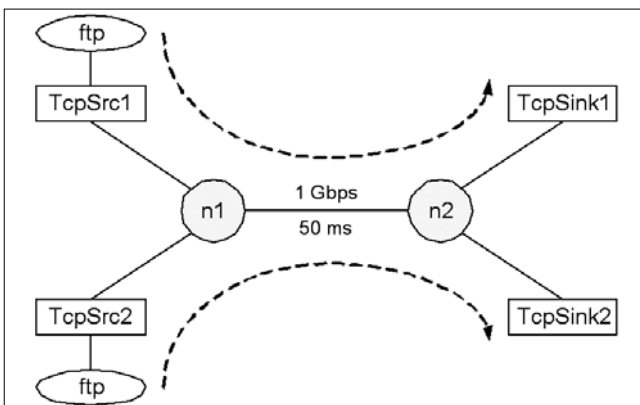
A Matlab/Simulink környezetben végzett szimulációk eredményeit az Ns-2 [7] csomag szintű szimulátorral validáltuk. Ez abban különbözik a folyamszintű vizsgálatoktól, hogy pontosabb eredményeket szolgáltat, de jóval lassabb lefutású. Ebben a környezetben különböző TCP folyamatok együttműködése vizsgálható az egy linket és várakozási sort tartalmazó dumb-bell topológiával. A topológiát a 3. ábra ismerteti.

A hálózat egy szűk sáv szélességű linkből áll, amire forrásokat és nyelőket kötünk. A hálózat forgalmát a vizsgált TCP protokollok által generált adatcsomagok adják. Természetesen a TCP kapcsolat kétirányú, de a nyelő-forrás irányban csak nyugták haladnak, ebben az értelemben tehát a forgalom egyirányúnak tekinthető.

4. Vizsgálati eredmények

A fairness vizsgálatoknál az összehasonlítandó protokollok ablak szabályozó blokkját egymás mellett alkalmazzuk és a forgalmakat összegezzük. Ekkor megfigyelhetjük a két protokoll ablakméretének változását és ezen keresztül az együttes viselkedésüket.

3. ábra Ns-2 dumb-bell környezet egyszerűsített rajza



A következőkben először megvizsgáljuk azt az esetet, amikor két egyforma protokoll verseng, majd pedig azt, amikor két nagysebességű TCP verzió kényszerül osztozni a hálózaton.

4.1. Intraprotokoll működés

A jelen esetben egy-egy Scalable TCP folyamat versengését tanulmányoztuk 50 másodperces indulási késleltetés mellett. A 4. ábra mutatja a MATLAB/Simulink és az Ns-2 által szolgáltatott eredményt.

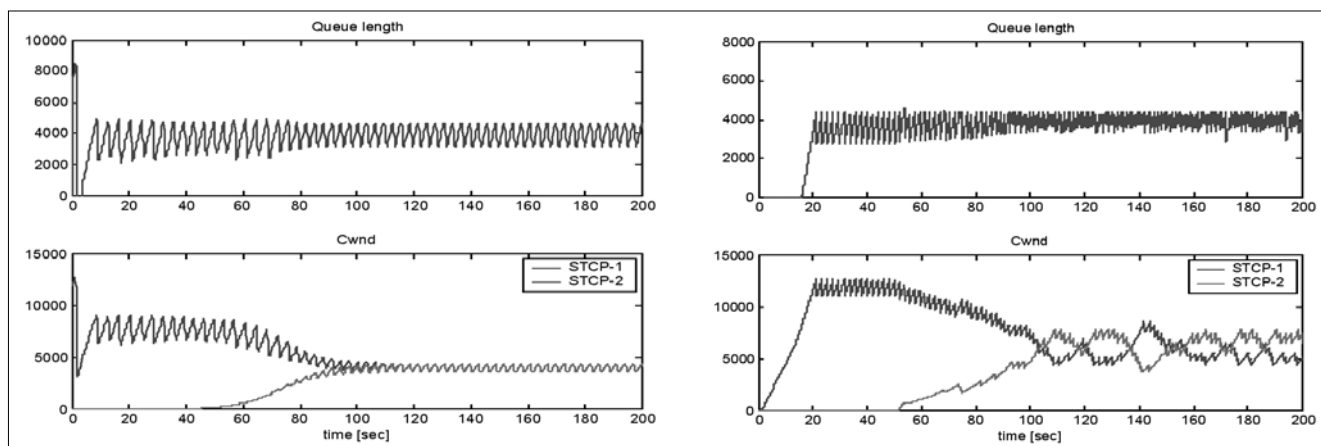
A modell és a szimuláció megfelelően működik, hiszen a különböző vizsgálati módszerekkel jellegre hasonló eredményt kaptunk. A különbségek a két program közötti eltérésekből adódnak. Az Ns-2-es szimulátor például figyelembe veszi a kezdeti (slow start) fázist, amit a Simulinkben nem modelleztünk, ebből ered a görbék elején tapasztalható eltérés. Ezenkívül míg az Ns-2 csomag szintű szimulációt végez, a Matlab/Simulinknél csupán várható értéket látunk. Az intraprotokoll működés mind a Scalable TCP, mind a HighSpeed TCP esetén hasonló eredményeket adott. Azt tapasztaltuk, hogy a protokollok mind egy, mind több folyamat esetén késleltetéssel és anélkül is teljesen fair módon működtek együtt.

Az intraprotokoll együttműködések vizsgálatokor minden protokoll esetén azt tapasztaltuk, hogy mind késleltetéssel, mind anélkül, függetlenül a folyamszámtól fair módon osztoztak a hálózaton a protokollok. Ez abban az esetben igaz, amikor RED mellett vizsgáltuk az együttműködést. Drop Tail-lel az eredmények különbözhetnek.

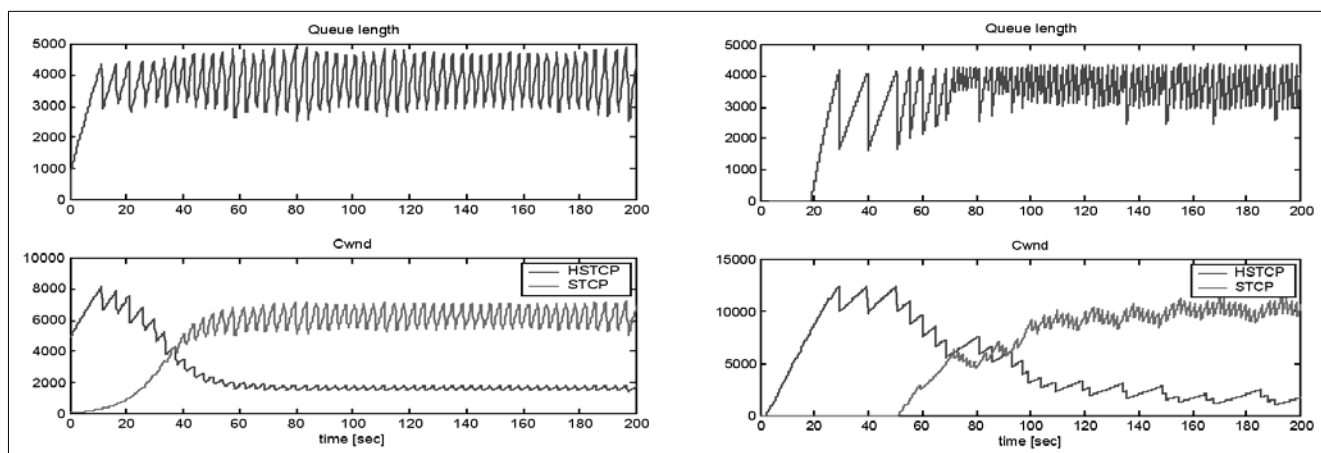
4.2. Interprotokoll működés

Ebben az esetben azt vizsgáltuk, hogy a két nagysebességű protokoll hogyan osztozik a hálózaton egyidejű illetve késleltetett indulás esetén. A 5. ábra bal oldalán a MATLAB/Simulink által szolgáltatott eredmény, míg a jobb oldali képen az Ns-2 szimuláció eredménye.

A Scalable TCP késleltetésétől függetlenül dinamikusabban működött és megszerezte csaknem a teljes sáv szélességet még 50 másodperccel későbbi indulás esetén is. Jelen esetben is láthatóak kisebb különbségek a modell és a szimuláció eredményei között, de ezek okai a már korábban említett eltérések a két vizsgálati módszer között.



4. ábra 1-1 Scalable TCP folyam indítása késleltetéssel (modell és szimuláció)



5. ábra 1-1 HS és Scalable TCP folyam indítása 50 s késleltetéssel (modell és szimuláció)

4.3. Következtetések

A nagysebességű TCP protokollok együttműködésének vizsgálatánál is az elvártaknak megfeleltek a kapott eredmények. A Scalable TCP MIMD algoritmusával dinamikusabb működést biztosít, jobban kihasználja a rendelkezésre álló sávszélességet. A dinamikusabb, agresszívebb protokoll minden esetben elnyomta a nála lassabbat. Jelen esetben az alkalmazott MIMD algoritmus dinamikusabb volt a HSTCP módosított AIMD algoritmusánál, így ez tudta nagyobb határfokkal kihasználni a rendelkezésre álló sávszélességet. Fontos kihangsúlyozni, hogy az eredményeket RED használata mellett kaptuk.

5. Összefoglalás

Cikkünkben nagysebességű TCP protokollok modelljeit vizsgáltuk olyan hálózaton, melynek torlódásvezérlése a RED algoritmust használja. A különböző nagysebességű verziók együttműködését ezen modelleken alapulva vizsgáltuk, majd eredményeinket csomagszintű szimulátorral validáltuk. Jövőbeli terveink között szerepel az eddigi vizsgálati környezet kibővítése további nagysebességű verziókkal, mint például az eddig még csak részben vizsgált késleltetés alapú protokollokkal, valamint a visszacsatolt rendszer stabilitásának vizsgálata.

Irodalom

- [1] S. Floyd, HighSpeed TCP for Large Congestion Window, RFC 3649, December 2003.
- [2] T. Kelly, Scalable TCP: Improving Performance in Highspeed Wide Area Networks, ACM SIGCOMM Computer Communication Review, 33(2), April 2003.
- [3] V. Misra, W. B. Gong, D. Towsley, Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED, in Proc. of ACM SIGCOMM, Stockholm, Sweden, 2000.
- [4] C. V. Hollot, V. Misra, D. Towsley, W. B. Gong, A Control Theoretic Analysis of RED, in Proc. of IEEE INFOCOM, Anchorage, USA, 2001.
- [5] B. Sonkoly, T. A. Trinh, S. Molnár, Understanding HighSpeed TCP: A Control-Theoretic Perspective, IASTED CCN 2005, Marina del Rey, CA, USA, October 24-26, 2005.
- [6] M.C. Weigle, P. Sharma, J. Freeman, Performance of Competing High-Speed TCP Flows, in Proc. of IFIP Networking, May 2006.
- [7] Ns-2 Network Simulator, <http://www.isi.edu/nsnam/ns>