

TCP protokolo modeliavimas bevieliame tinkle

L. Pavilanskas

*Telekomunikacijų inžinerijos katedra, Vilniaus Gedimino technikos universitetas,
Naugarduko g. 41, LT-03227 Vilnius, Lietuva, tel. +370 5 274 49 77, el. p. lukas.pavilanskas@el.vtu.lt*

Ivadas

Didžiausia problema, su kuria susiduriama diegiant IEEE 802.11 standarto tinklus, yra atsitiktiniai paketų praradimai [1] dėl mažo signalo ir triukšmo santykio. Paketų praradimai daugiausia įtakos turi informacijos perdavimo TCP protokolu spartai [2].

TCP faktiškai yra patikimą ryšį užtikrinantis protokolas, kuris, naudodamas persipildymo lango, lėto starto ir persipildymo kontrolės mechanizmus [RFC 793], [RFC 2581], [RFC 2582], atlieka heterogeninių telekomunikacijų tinklų srautų kontrolę. Šie TCP protokolo mechanizmai yra išsamiai išanalizuoti, sudaryti ir ištirti jų modeliai, kai bevielio ryšio linijose vyrauja paketų praradimai [3], [4], [5], [6], [7].

Dažnai naudojamus TCP protokolo algoritmų modelius – OldTahoe, Tahoe, Reno ir NewReno – analitiniu būdu ištyrė A. Kumar [2]. Jis nustatė spartos funkcijos priklausomybę nuo prarandamų paketų tikimybės, kai bevielio ryšio kanaluose veikia nekoreliuoti atsitiktiniai paketų praradimai. Visų ištirtų TCP algoritmų analizės rezultatai rodo, kad, didėjant paketų praradimo tikimybei, mažėja TCP sesijos laidumas. Tai netaikoma TCP Vegas algoritmui [6], kuris, palyginti su Reno algoritmu, esant paketų praradimams, perdavimo spartą gali padidinti iki 70 %. Tačiau tai įmanoma, kai tinkle naudojamas tik TCP Vegas algoritmas [8]. Šio trūkumo neturi pagal Vegas modelį modifikuotas Reno algoritmas, vadinamas Veno algoritmu, kuris, esant 1 % atsitiktiniams paketų praradimams, spartą gali padidinti iki 80 % [9].

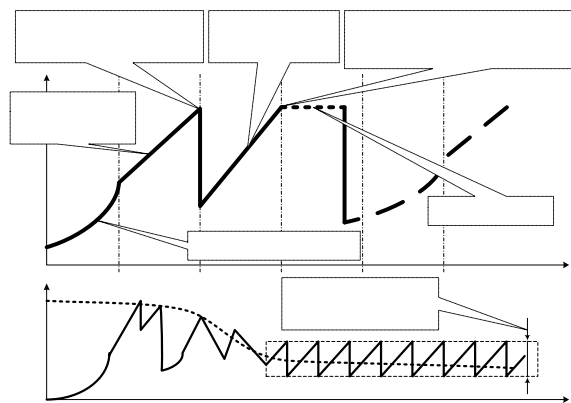
Siekiant išvengti atsitiktinių paketų praradimų bevielių tinklų linijose, naudojami žemesnio OSI lygmens patikimą ryšį užtikrinantys protokolai [10]. Šių protokolų vienas esminių bruožų yra tai, kad paketas siunčiamas, kol gaunamas MAC lygmens ACK paketas. Jei paketui nusiųsti reikia labai daug laiko, TCP algoritmas dėl TCP kritinės laukimo laiko ribos nustatys, kad paketas yra prarastas [3]. Ši problema ištyrinėta analizuojant palydovines ryšio linijas, kuriose nuolatinis paketų vėlinimas siekia iki 300 ms [11], [12], [13]. IEEE 802.11 standarto tinkluose, naudojant patikimą ryšį užtikrinančius protokolus, vėlinimo trukmė yra gerokai ilgesnė. Be to, vėlinami yra atsitiktiniai, pakartotinai siunčiami paketai.

Pateiktas susistemintas TCP protokolo Reno, Vegas ir Veno algoritmų aprašymas ir šių algoritmų modeliavimo IEEE 802.11b standarto ryšio linijoje rezultatai.

Analizuojamos šių algoritmų spartos priklausomybės nuo atsitiktinių paketų praradimų ir ilgų vėlinimų, kurie atsiranda naudojant patikimą ryšį užtikrinančius protokolus žemesniuose OSI lygmenyse. Modeliavimo rezultatais įrodyta, kad, esant dideliems paketų atsitiktiniams praradimams, naudotinas TCP Veno algoritmas. Tačiau ilgų atsitiktinių vėlinimų atveju bevieliuose IEEE 802.11 prieigos tinkluose tinkamesnis tampa TCP Reno. Modeliavimų tikslas – išanalizuoti TCP protokolų algoritmų savybes ir jų įtaką spartai IEEE 802.11b standarto tinkluose.

TCP Reno algoritmas

Internetas yra heterogeninė sistema, kurioje dėl skirtingų ryšio linijų charakteristikų prarandami paketai [9]. Norint to išvengti, naudojamas TCP protokolas, kuris, kontroliuodamas perduodamos informacijos srautą, užtikrina patikimą ryšį. TCP protokolo veikimo algoritmų, kurie atlieka srauto kontrolę, yra labai daug, tačiau dažniausiai naudojamas TCP Reno [2].



1 pav. TCP protokolo algoritmo veikimo schema

Reno gali nustatyti paketų praradimus tinkle, tačiau tokius įvykius traktuoja kaip tinklo persipildymą [3]. Bevielio ryšio linijose paketai atsitiktinai prarandami dėl triukšmo radijo ryšio kanaluose [14]. Dėl šios priežasties, naudojant TCP protokolą, mažėja informacijos perdavimo sparta [1], [9], [4].

TCP Reno algoritmas srauto kontrolę atlieka persipildymo lango, lėto starto ir persipildymo kontrolės

mechanizmais [RFC 2581], [RFC 2582]. Šio algoritmo persipildymo lango dydis normaliomis sąlygomis didinamas periodiškai, kol pasiekiamas maksimalus persipildymo lango dydis ($cwnd_{maks}$). TCP Reno naudoja du lango dinaminis režimus: lėto starto fazę ir persipildymo išvengimo fazę (1 pav.). Laiko momentu $t + t_A[s]$ persipildymo lango vertė $cwnd(t)$ kinta:

$$cwnd(t+t_A) = \begin{cases} \text{leto starto fazė} \\ cwnd(t)+1, \text{ jei } cwnd(t) < ssth(t); \\ \text{persipildymo išvengimo fazė} \\ cwnd(t) + \frac{1}{cwnd(t)}, \text{ jei } cwnd(t) \geq ssth(t); \end{cases} \quad (1)$$

čia $ssth(t)$ [paketais] – slenksčio vertė, kuriai esant TCP pereina iš lėto starto į persipildymo išvengimo fazę.

TCP laiko, kad paketas yra sėkmingai nusiųstas, kai gaunamas gavėjo išsiųstas patvirtinantis ACK paketas. Laikas, per kurį turi būti gautas ACK, vadinamas RTO:

$$RTO = \overline{RTT} + 4\sigma_{\overline{RTT}}; \quad (2)$$

čia \overline{RTT} – paketui nusiųsti ir gauti ACK reikalingo laiko vidurkis; $\sigma_{\overline{RTT}}$ – \overline{RTT} dispersija. Kai RTO pernelyg didelis, paketo praradimo nustatymas bus atidėtas, dėl to sumažės TCP sesijos laidumas [RFC 2581]. Jei per šį periodą dėl kanalo persipildymo negaunamas ACK, Reno išsiųstą paketą laiko prarastu (1 pav.). Persipildymo langas tampa minimalus, o slenksstis sumažėja:

$$cwnd(t) = 1; \quad (3)$$

$$ssth(t) = \frac{cwnd(t)}{2}. \quad (4)$$

Tačiau jei gaunamas kitą paketą patvirtinantis dubliuojantis ACK, suveikia greito atkūrimo mechanizmas [RFC 2581]. TCP pereina į persipildymo išvengimo fazę:

$$cwnd(t) = ssth(t); \quad (5)$$

$$ssth(t) = \frac{cwnd(t)}{2}. \quad (6)$$

Jei informacija perduodama bevielio ryšio kanalu, kuriame yra atsitiktinai prarandami paketai, nuolat nustatomas minimalus persipildymo lango dydis (3). Dėl to TCP sesijos laidumas sumažėja.

TCP Vegas ir Veno algoritmai

Vegas algoritmui būdingas efektyvesnis $cwnd(t)$ nustatymo mechanizmas, kadangi, nustatant $cwnd(t)$, yra įvertinama paketo perdavimo trukmė – RTT [8]. Šis algoritmas, siuntėjo pusėje matuodamas RTT, nustato numatomą ir faktinį sesijos laidumą (2 pav.) [5]:

$$C_l = cwnd(t) / RTT_{\min}; \quad (7)$$

$$C_f = cwnd(t) / RTT; \quad (8)$$

čia C_l – numatomas sesijos laidumas; C_f – faktinis sesijos laidumas; $cwnd(t)$ – TCP persipildymo lango dydis; RTT_{\min} – mažiausia nustatyta RTT vertė iki išsiunčiant paketą. Taigi perdavimo spartos skirtumas:

$$\Delta = C_l - C_f. \quad (9)$$

Jei $RTT > RTT_{\min}$, tai ryšio linija yra mažo laidumo, todėl perduodami paketai bus kaupiami (2 pav.) [6]. Dėl šios priežasties sumažės $cwnd(t)$. Jei šią kaupiamų paketų eilę pavadinsime N , tuomet

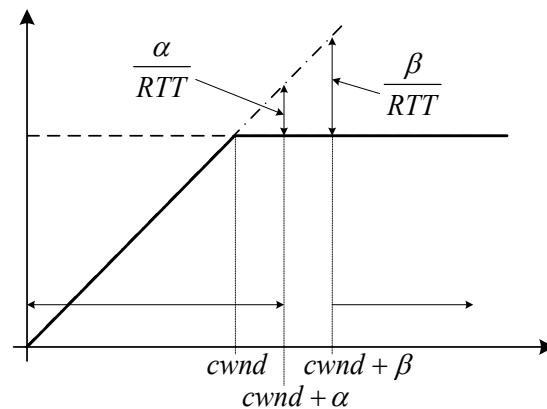
$$RTT = RTT_{\min} + \frac{N}{C_f}; \quad (10)$$

$$N = C_l(RTT - RTT_{\min}) = \Delta RTT_{\min}. \quad (11)$$

Taigi TCP Vegas sieks išlaikyti kuo mažesnę N , keisdamas TCP persipildymo lango dydį [5]. Be to, priešingai nei TCP Reno, nesvarbu, dėl kokių klaidų buvo prarastas paketas, jis iš karto nenusitais $cwnd(t) = 1$. Dėl to laidumas bus išnaudotas efektyviai. Naudojant Vegas algoritimą, lyginant su Reno, perdavimo sparta padidėja nuo 37 % iki 70 % [5]. Pagal suformuluotus teiginius Vegas algoritmo matematinė išraiška yra tokia:

$$cwnd(t) = \begin{cases} cwnd(t) + 1, & \text{jei } \Delta < \frac{\alpha}{RTT_{\min}}, \\ cwnd(t), & \frac{\alpha}{RTT_{\min}} \leq \Delta \leq \frac{\beta}{RTT_{\min}}, \\ cwnd(t) - 1, & \text{jei } \frac{\beta}{RTT_{\min}} < \Delta; \end{cases} \quad (12)$$

čia α – $cwnd(t)$ didinimo koeficientas: jei $cwnd_{maks} > cwnd(t) + \alpha$, tada $cwnd(t) + 1$; β – $cwnd(t)$ mažinimo koeficientas: jei $cwnd_{maks} < cwnd(t) + \beta$, tada $cwnd(t) - 1$.



2 pav. TCP Vegas protokolo persipildymo nustatymo veikimo schema

Šio algoritmo trūkumas tas, kad lango dydis nustatomas neefektyviai, kai ta pačia ryšio linija vyksta kitos TCP Reno algoritmo sesijos [8]. Suveikęs greito atkūrimo mechanizmas turi įtakos TCP Vegas algoritmo RTT laikui, nuo kurio priklauso TCP protokolo perduodamos informacijos sparta:

$$\theta = 0,93 \frac{MTU}{RTT \sqrt{\rho}}; \quad (13)$$

čia MTU – maksimalus paketo dydis; ρ – paketų praradimų tikimybė. Ši savybė leidžia teigti, kad Vegas algoritmas yra jautrus kintamam RTT laikui.

Siekiant pašalinti Vegas algoritmo trūkumus, Reno algoritmą galima pakeisti taip, kad, esant paketų praradimams, persipildymo lango dydis nebūtų staiga mažinamas. Tam reikia sujungti Reno ir Vegas algoritmų analizuojamu požiūriu palankias savybes. Tokius pakeitimus pasiūlė padaryti P. F. Cheng ir C. L. Soung [9]. Naujas algoritmas buvo pavadintas TCP Venó. Jo veikimas aprašomas matematinėmis išraiškėmis:

$$cwnd(t + t_A) = \begin{cases} cwnd(t) + 1, & \text{jei } N < \beta, \\ cwnd(t) + \frac{1}{cwnd(t)}, & \text{jei } N \geq \beta, \end{cases} \quad (14)$$

$$ssth(t + t_A) = \begin{cases} \frac{4cwnd}{5}, & \text{jei } N < \beta, \\ \frac{cwnd(t)}{2}, & \text{jei } N \geq \beta. \end{cases} \quad (15)$$

Nesvarbu, kokio tipo yra paketų praradimai – persipildymo ar atsitiktiniai, Venó algoritmas įvertina ryšio būvį. Naudojant šį algoritmą, IEEE 802.11 tinkluose, kuriuose vyrauja iki 1 % atsitiktinių paketų praradimai, lyginant su Reno algoritmu, pasiekama 80 % didesnė sparta [9].

Ilgo vėlinimo įtaka TCP algoritmams

Norint išvengti atsitiktinių paketų praradimų IEEE 802.11 ryšio linijose, naudojami žemesnio OSI lygmens patikimą ryšį užtikrinantys protokolai [16]. Vienas iš esminių tokių protokolų bruožų – tai, kad paketas siunčiamas tiek kartų, iki gaunamas kanalinio (MAC) lygmens ACK paketas [10]. Dėl to didėja aukštesnių OSI lygmenų protokolais perduodamų paketų vėlinimai. TCP atveju pailgėja RTT trukmė, kuri turi įtakos $cwnd(t)$ [11]. Kadangi patikimą ryšį užtikrinantys protokolai vėlina tik atsitiktinius paketus, todėl padidėja tik šių paketų RTT.

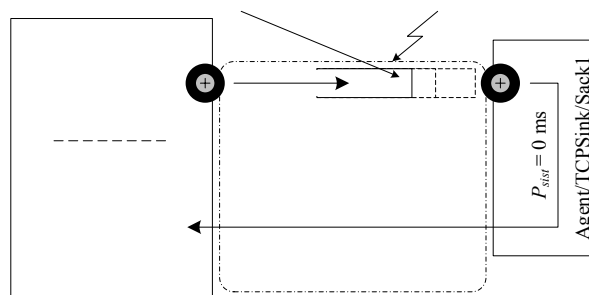
Naudojant TCP Reno algoritmą, trumpai vėlinamas paketas nustatys trumpą RTO (2) trukmę. Dėl to gali būti prarastas kitas, ilgesnio RTT laiko, paketas [RFC 2581]. Tačiau jei suvėlintas paketas nebus prarastas, jis pailgins RTO trukmę (2), dėl kurios, persipildžius TCP langui, gali būti atidėtas dingusio paketo nustatymas [RFC 793]. Be to, dėl ilgo RTO periodo TCP nustatys „lėto starto“ fazę. Akivaizdu, kad dėl šių savybių TCP Reno sesijos laidumas priklauso nuo atsitiktinių ilgai vėlinamų paketų procentinio skaičiaus [11].

TCP Vegas ir Venó atveju, jei $RTT \gg RTT_{\min}$, tai N (11) didės, dėl to sumažės $cwnd(t)$, kuris mažins TCP sesijos laidumą [8].

Atlikus įvairiose Lietuvos vietovėse esančių IEEE 802.11b linijų, kuriose naudojamas patikimą ryšį užtikrinantis „TurboCell“ [16] protokolas, analizę, nustatyta, kad vyraujanti RTT_{\min} trukmė yra 128 ms, o RTT_{\max} trukmė svyruoja nuo 1,2 iki 2 s. Analizuojamos linijose vidutinis prarastų paketų skaičius neviršijo 1 %. Nustatyta, kad kai kuriais atvejais RTT_{\max} gali siekti iki 5 s. Matavimai atlikti naudojantis „Karlnet Configurator“ programa [16].

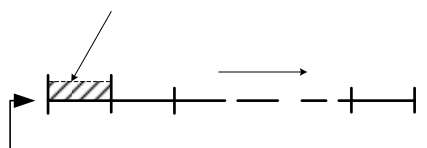
Modeliavimo metodika

Norint išsiaiškinti TCP protokolų, naudojamų IEEE 802.11 standarto tinkluose, TCP algoritmų reakcijas į atsitiktinių paketų praradimus bei priverstinį ilgą atsitiktinį vėlinimą, atliktas TCP protokolo algoritmų modeliavimas. Tam panaudotas „NS-2“ paketas [17].



3 pav. TCP algoritmų modeliavimo schema

Tyrimo metu atlikti trys modeliavimai, kurių metu tarp dviejų IEEE 802.11b standarto ryšio mazgų (3 pav.), naudojant Reno, Vegas ir Venó algoritmus, buvo perduvinėjama informacija. Naudotas 1494 baitų segmentų FTP srauto modelis [17].



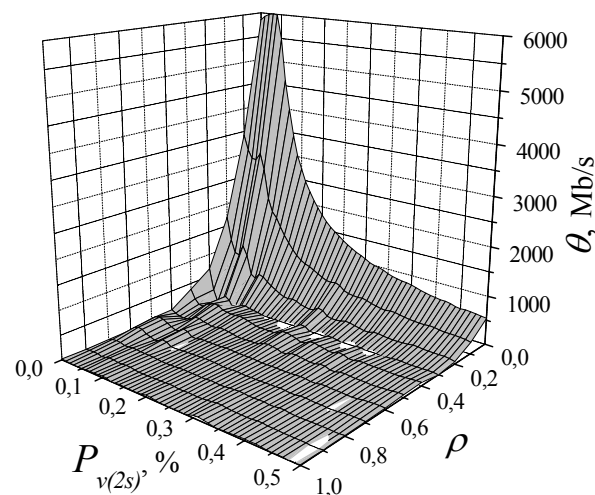
4 pav. Modeliavimo laikinė diagrama

Tiriant TCP algoritmus, nustatytas $45 \cdot MTU$ dydžio persipildymo langas ir 0 ms TCP sisteminis vėlinimas. TCP Vegas ir Venó algoritmų atveju nustatyti koeficientai $\beta = 3$ ir $\alpha = 1$ (2 pav.) [9]. Nustatyta įrangos fizinė sparta – 11 Mb/s. Tarp mazgų panaudoti „Free Space“ radijo ryšio linijos ir eksponentiniai atsitiktinių klaidų ir vėlinimų modeliai [17].

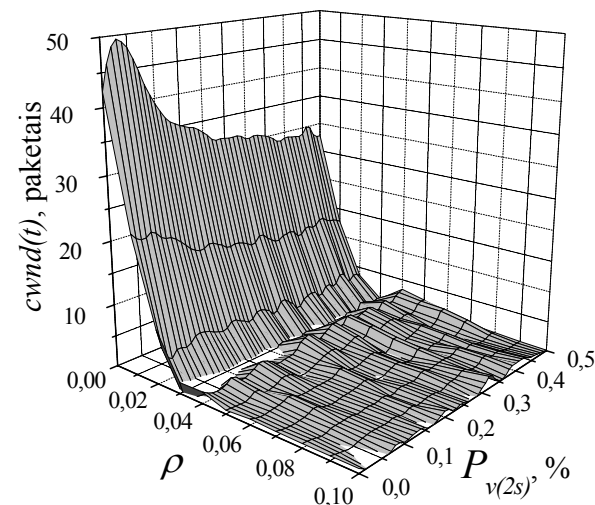
Modeliavimo metu kas 10 sekundžių 100 kartų keistas atsitiktinis ilgai vėlinamų paketų procentinis skaičius P_v (4 pav.). Kas 1000 sekundžių keista atsitiktinių paketų praradimų tikimybė ρ . Modeliavimo trukmė – 2 h 78 min (4 pav.).

TCP Reno algoritmo modeliavimas

Atlikus TCP Reno algoritmo modeliavimą IEEE 802.11b ryšio linijoje, nustatyta, kad perdavimo sparta priklauso nuo atsitiktinių paketų praradimų tikimybės ir vėlinamų paketų procento. Jeigu paketų praradimo tikimybė neviršija 10^{-2} , sparta sumažėja 40 % (iki 3,5 Mb/s). Minimumą ($\theta \approx 10 \text{ kb/s}$) ji pasiekia, kai paketų praradimo tikimybė yra $6 \cdot 10^{-2}$ (5 pav.). Tokiomis sąlygomis $cwnd(t)$ dažnai pereina į lėto starto fazę (3) (4), dėl kurios persipildymo langas tampa minimalus (6 pav.). Ryšys tampa nebeįmanomas, jei paketų praradimo tikimybė padidėja daugiau nei $6 \cdot 10^{-2}$. Gauti rezultatai atitinka realius eksperimentinių tyrimų rezultatus [9].



5 pav. TCP Reno spartos priklausomybė nuo atsitiktinių paketų praradimų ir ilgo atsitiktinio vėlinimo



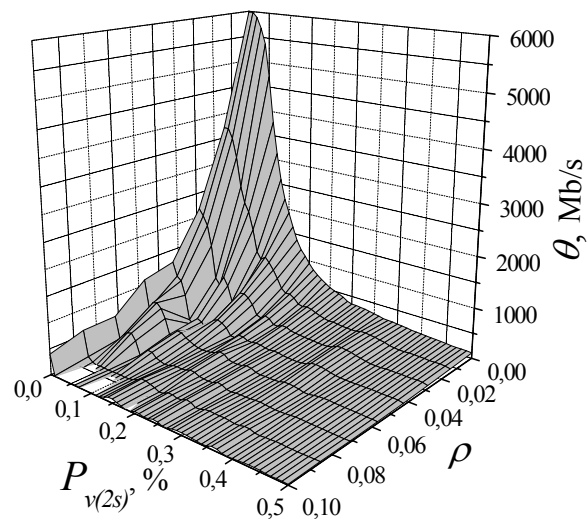
6 pav. TCP Reno lango dydžio priklausomybė nuo atsitiktinių paketų praradimų ir ilgo atsitiktinio vėlinimo

Kai atsitiktinai 2 s vėlinami 10 % paketų, naudojant Reno algoritmą, sparta sumažėja iki 50 % (~3 Mb/s).

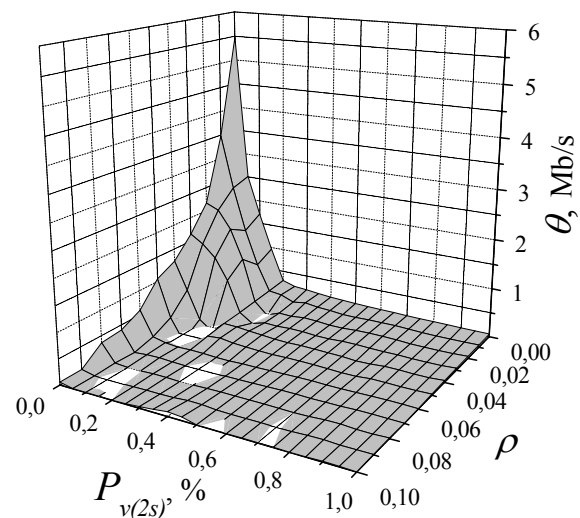
TCP Reno $cwnd(t)$ kitimas nepriklauso nuo RTT laiko (1), todėl ilgai vėlinant atsitiktinius paketus, $cwnd(t)$ priklausys nuo vėlinimo charakteristikos modeliavimo eksponentinio dėsnio (6 pav.).

TCP Vegas ir Venio algoritmų modeliavimas

Iš TCP Venio algoritmo modeliavimo rezultatų nustatyta, kad esant 10^{-2} paketų praradimams, sparta, kaip ir TCP Reno atveju, sumažėja 40 % (iki 3,5 Mb/s) (7 pav.). Tačiau, esant $3 \cdot 10^{-2}$ paketų praradimams, lyginant su Reno algoritmu (0,8 Mb/s), sparta 40 % didesnė (1,4 Mb/s) [9].



7 pav. TCP Venio spartos priklausomybė nuo atsitiktinių paketų praradimų ir ilgo atsitiktinio vėlinimo

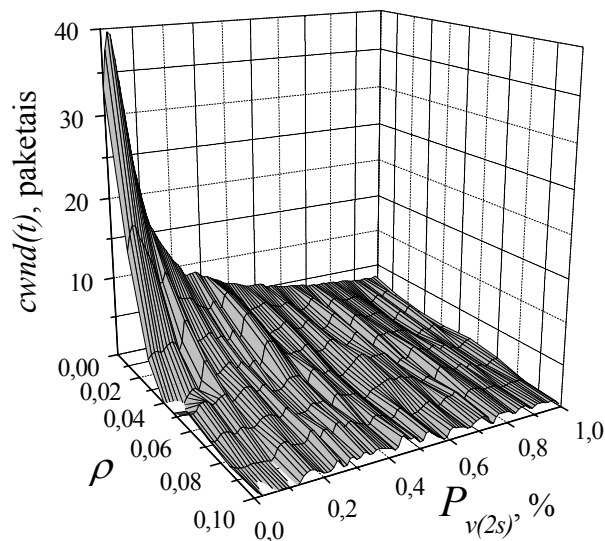


8 pav. TCP Vegas spartos priklausomybė nuo atsitiktinių paketų praradimų ir ilgo atsitiktinio vėlinimo

Perduodant informaciją TCP Venio algoritmu, kai yra atsitiktinai 2 s vėlinami paketai, spartos mažėjimo priklausomybė, lyginant su TCP Reno algoritmu, yra mažesnė. Kai 10 % atsitiktinių paketų vėlinami 2 s, sparta

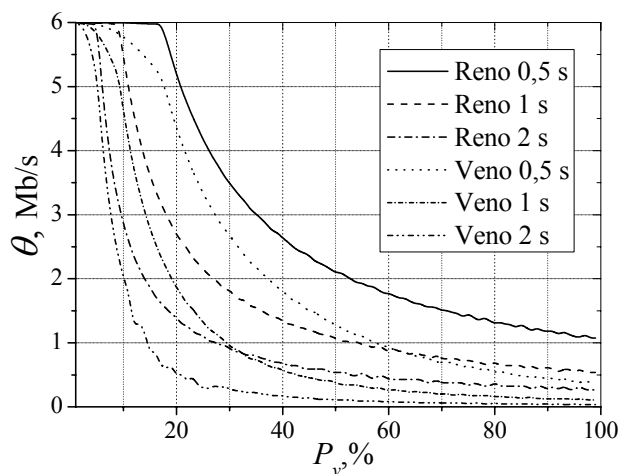
skiriasi nežymiai – 2,2 Mb/s, tačiau vėlinamų paketų skaičiui išaugus iki 30 %, TCP Reno algoritmu pasiekama 65 % didesnė sparta (1,5 Mb/s). Tai galima aiškinti tuo, kad TCP Veno algoritmas yra pagal TCP Vegas veikimo principą modifikuotas Reno algoritmas.

TCP Vegas atveju (8 pav.) spartos priklausomybė nuo paketų praradimų atitinka Veno charakteristiką. Šio algoritmo $cwnd(t)$ priklauso nuo RTT laiko (12), dėl to didelė šio laiko deviacija lems pastovų $cwnd(t)$ kitimą [5]. Tai atspindi gauti rezultatai – esant daugiau nei 20 % 2 s atsitiktinai vėlinamų paketų, informacijos perdavimo sparta sumažėja iki minimumo.



9 pav. TCP Veno lango dydžio priklausomybė nuo atsitiktinių paketų praradimų ir ilgo atsitiktinio vėlinimo

Naudojant TCP Veno algoritimą, informacijos perdavimo spartos charakteristikos proporcingai priklauso nuo persipildymo lango dydžio (9 pav.). $cwnd(t)$ kitimo charakteristika dėl TCP Veno algoritmo veikimo principo (14) (15) priklauso nuo modeliavimuose naudotos atsitiktinių paketų praradimo ir vėlinimo charakteristikų.



10 pav. TCP Reno ir Veno algoritmų spartos priklausomybės nuo 0,5, 1 ir 2 s atsitiktinio vėlinimo

Modeliavimo metu atlikti tyrimai su TCP Reno ir Veno algoritmais, naudojant maksimalius atsitiktinius 0,5 ir 1 s vėlinimus (10 pav.). Iš gautų rezultatų matyti, kad visais modeliavimo atvejais, esant ilgiems vėlinimams, naudojant TCP Veno algoritimą, sparta yra mažesnė už spartą, gautą naudojant Reno algoritimą.

Išvados

Atlikus TCP protokolų Reno, Vegas ir Veno algoritmų IEEE 802.11 standarto ryšio linijoje modeliavimą gauti spartos priklausomybės nuo atsitiktinių paketų praradimų bei atsitiktinių ilgų vėlinimų duomenys. Jais remiantis galima suformuluoti tokius teiginius:

1. Naudojant TCP Reno algoritimą, galimas ryšys, esant ne didesniems kaip $6 \cdot 10^{-2}$ paketų praradimams. Kai prarandama 10^{-2} paketų, sparta sumažėja 40 %. Tuo tarpu didėjant vėlinamų paketų skaičiui, vidutinis TCP lango dydis dėl kintančios RTT trukmės eksponentiškai mažėja.

2. TCP Vegas yra nepritaikytas dirbti tinkluose su didele vėlinimų deviacija, todėl, esant 20 ir daugiau procentų atsitiktinai 2s vėlinamų paketų, informacijos perdavimo sparta sumažėja net 95 %.

3. Naudojant TCP Veno algoritimą, kai prarandama 10^{-2} paketų, sparta yra tokia pat kaip naudojant Reno algoritimą. Tačiau paketų praradimams pasiekus $3 \cdot 10^{-2}$, sparta yra 40 % didesnė (1,4 Mb/s). Tuo tarpu atsitiktinai 2 s vėlinant paketus, spartos priklausomybė, lyginant su Reno, yra blogesnė. Vėlinamų paketų skaičiui išaugus iki 30 %, pasiekama 65 % mažesnė sparta.

4. Vėlinant 10 % paketų 2 s, sparta atitinka TCP Reno spartą tomis pačiomis sąlygomis, tačiau, kai vėlinamų paketų skaičius siekia 30 ir daugiau procentų, TCP Reno algoritmu informacija perduodama 75 % sparčiau (1Mb/s).

Reziumuojant galima teigti, kad naudojant bet kurį iš modeliūtų TCP algoritmų, atsitiktiniai paketų praradimai ir atsitiktiniai ilgi vėlinimai turės įtakos perdavimo spartai. Esant dideliems paketų atsitiktiniams praradimams, naudotinas TCP Veno algoritmas, tačiau ilgų atsitiktinių vėlinimų atveju tinkamesnis yra Reno algoritmas. Dėl to naudojant patikimą ryšį užtikrinančius protokolus IEEE 802.11b tinkluose, reikia įvertinti atsitiktinai prarandamų ir vėlinamų paketų santykį. Jam mažėjant patikimą ryšį užtikrinantys protokolai [10], [16] tampa neefektyvūs.

Literatūra

1. **Lakshman T. V., Madhow U.** The performance of TCP/IP for networks with high bandwidth-delay products and random loss // IEEE/ACM. – Transactions on Networking. – 1997. – Vol. 5, No 3. – P. 336-350.
2. **Kumar A.** Comparative performance analysis of versions of TCP in a local network with a lossy link // IEEE/ACM. – Transactions on networking. – 1998. – Vol. 6, No. 4.
3. **Jacobson V., Karels M. J.** Congestion Avoidance and Control // Uni. of California at Berkeley. – 1988.
4. **Balakrishnan H., Padmanabhan V. N., Seshan S., Katz R. H.** A comparison of mechanisms for improving TCP performance over wireless links // IEEE/ACM. – Transactions on Networking, (5). – 1997. – Vol. 6. – P. 756-769.

5. **Brakmo L. S., O'Malley S. W., Peterson L. L.** TCP Vegas: New Techniques for Congestion Detection and Avoidance // SIGCOMM. – 1994. – P. 24-35.
6. **Jeonghoon M., Richard L. L., Venkat A., Walrand J.** Analysis and comparison of TCP Reno and Vegas Networking // IEEE/ACM. – Transactions on networking. – 1999. – Vol. 6. – P. 1556-1563.
7. **Floyd S., Jacobson V.** Random early detection gateways for congestion avoidance Networking // IEEE/ACM. – Transactions on networking. – 1993. – Vol. 1, (4). – P. 397-413.
8. **Hengartner U., Bolliger J., Gross T.** TCP Vegas Revisited // INFOCOM'2000. – 2000. – Vol. 2. – P. 1546-1555.
9. **Fu C. P., Liew S. C.** TCP VenO: TCP Enhancement for Transmission Over Wireless Access Networks // IEEE. – Communications. – 2003. – Vol.2(21).
10. **Airhook** – reliable data delivery protocol // <http://airhook.org/protocol.html> [prieiga per internetą]
11. **Peng J., Andreadis P., Barbeau C. B. M.** Improving TCP Performance over Long Delay Satellite Links // OPNETWORK'2001. – Washington, 2001. – Vol. 1.
12. **Partridge C., Shepard T. J.** TCP/IP performance over Satellite Links // IEEE/ACM. – Transactions on networking. – 1997. – P. 44-49.
13. **Ghani N., Dixit S.** TCP/IP enhancements for satellite networks // IEEE. – Communications Magazine. – 1999. – Vol. 37. – P. 64-72.
14. **Šaltis A., Pavilanskas L.** Paketinio radijo ryšio 2,4 GHz dažnių ruože eksperimentiniai tyrimai // Elektronika ir elektrotechnika. – Kaunas: Technologija. 2003. – Nr. 46. – P. 48-56.
15. **Mo J., La R. J., Anantharam V., Walrand J. J.** Analysis and Comparison of TCP Reno and Vegas // INFOCOM'99. – 1999. – Vol. 3. – P. 1556-1563.
16. **Karl D., Monahan S., Whitman A.** KarlNet's TurboCell: Enhancing the Capabilities of Standard 802.11, 2004 // www.karlnet.com [prieiga per internetą].
17. **Fall K.** The ns Manual // UC Berkeley, LBL, USC/ISI, and Xerox PARC, <http://www.isi.edu/nsnam/ns> [prieiga per internetą].

Pateikta spaudai 2005 03 02

L. Pavilanskas. TCP protokolo modeliavimas bevielame tinkle // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2005. – Nr. 5(61). – P. 78–83.

Pateiktas susistemintas TCP protokolo Reno, Vegas ir VenO algoritmų aprašymas ir šių algoritmų modeliavimo rezultatai IEEE 802.11b standarto ryšio linijoje. Analizuojama šių algoritmų spartos priklausomybės nuo atsitiktinių paketų praradimų ir ilgų vėlinimų, kurie atsiranda naudojant patikimą ryšį užtikrinančius protokolus žemesniuose OSI lygmenyse. Modeliavimo rezultatais įrodyta, kad esant dideliems paketų atsitiktiniams praradimams, naudotinas TCP VenO algoritmas, tačiau ilgų atsitiktinių vėlinimų atveju bevieluose IEEE 802.11 priegijos tinkluose tinkamesnis tampa TCP Reno. Modeliavimų tikslas – išanalizuoti TCP protokolų algoritmų savybes ir jų įtaką spartai IEEE 802.11b standarto tinkluose. Il. 10, bibl. 17 (lietuvių kalba; santraukos lietuvių, anglų ir rusų k.).

L. Pavilanskas. TCP Modelling in the Wireless Network Link // Electronics and Electrical Engineering. – Kaunas: Technologija, 2005. – No. 5(61). – P. 78–83.

This paper deals with the TCP Reno, Vegas and VenO systematic analysis and modelling results of TCP Reno, Vegas and VenO algorithms in the IEEE 802.11b networks link. Bias of TCP session throughput against random packet losses and long random delay emerging when link layer realable protocols are used is analyzed. The modelling results demonstrate that if random packet losses are very heavy the TCP VenO algorithm should be used, and if random long delay becomes heavy the TCP Reno should be used. The modelling aim is – to sift features of TCP protocol algorithms and find it's dependences on throughput in the IEEE 802.11 networks. Ill. 10, bibl. 17 (in Lithuanian; summaries in Lithuanian, English, Russian).

Л. Павиланскас. Моделирование TCP протокола в канале беспроводных сетей // Электроника и электротехника. – Каунас: Технология, 2005. – № 5(61). – P. 78–83.

Представлен систематический анализ TCP Reno, Vegas и VenO алгоритмов, и их моделирование в беспроводных IEEE 802.11b сетях. Анализируются зависимости пропускной способности от вероятности случайных потерь пакетов и случайных длинных временных задержек, которые возникают, используя надёжную связь гарантирующих протоколов в нижних слоях OSI систему. Результатами моделирования доказано что при больших случайных потерях пакетов лучше использовать TCP VenO алгоритм, но при большом количестве долго задерживаемых пакетов лучше использовать TCP Reno. Цель моделирования – проанализировать свойства алгоритмов TCP протокола и разуть их влияние на пропускную способность в IEEE 802.11 сетях. Ил. 10, библи. 17 (на литовском языке; рефераты на литовском, английском и русском яз.).