

TCP CONGESTION WINDOW (CWND) NING VAQT BO'YICHA O'ZGARISHINI MODELLASHTIRISH

Mirzohid Berdimuradov

berdimurodov.mirzohid@gmail.com

1 University of Management and Future Technologies universiteti
assistenti;

Annotatsiya: Mazkur maqolada TCP (Transmission Control Protocol) protokolining tiqilichni boshqarish mexanizmi doirasida congestion window (CWND) parametrining vaqt bo'yicha o'zgarishini modellashtirish masalasi ko'rib chiqiladi. CWND TCP oqimining asosiy boshqaruv kattaligi bo'lib, tarmoqda bir vaqtning o'zida "yo'lda" bo'lishi mumkin bo'lgan tasdiqlanmagan ma'lumotlar hajmini chegaralaydi. Shuning uchun uning dinamikasini tahlil qilish tarmoqning o'tkazuvchanligi, barqarorligi va tiqilinchga moyilligini baholashda muhim ahamiyatga ega.

Kalit so'zlar: TCP, SYN, SYN-ACK, ACK, CWND, Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery.

TCP (Transmission Control Protocol) - Internetning asosiy transport protokollaridan biri bo'lib, ishonchli, tartibli va ulanishga asoslangan ma'lumot uzatishni ta'minlaydi. U OSI modelining transport pog'onasida ishlaydi va ko'plab internet xizmatlarining (HTTP/HTTPS, SSH, FTP, SMTP) funkcionalligini ta'minlovchi asosiy mexanizm hisoblanadi[1].

TCP ning asosiy maqsadi - ikkita qurilma orasida xatolarsiz, yo'qotilmasdan va ketma-ketlik bo'yicha buzilmasdan ma'lumot yetkazib berishdir.

TCP ning asosiy xususiyatlari

Ulanishga asoslangan protokol (Connection-oriented)

TCP ma'lumot uzatishni boshlashdan oldin muloqotni o'rnatadi. Bu 3-bosqichli qo'l siqish (3-Way Handshake) orqali amalga oshadi[2]:

1. SYN (signal)
2. SYN-ACK (tasdiq)
3. ACK (yakuniy tasdiq)

Ishonchlilik (Reliability)

TCP har bir yuborilgan segmentni kuzatib boradi va quyidagilar orqali ishonchlilikni ta'minlaydi[3]:

1. ACK (Acknowledgment) mexanizmi
2. Paketlardan biri yo'qolsa — qayta yuborish (Retransmission)
3. Raqamlash (Sequence number) orqali ketma-ketlikni boshqarish
4. Paketlarni qayta yig'ish (Reassembly) mexanizmi

Oqimni boshqarish (Flow Control)

TCP har ikki tomondagi bufer to'lishining oldini olish uchun Receiver Window (RWND) deb ataladigan mexanizmni qo'llaydi.

Maqsad — tez yuborayotgan jo'natuvchi sust qabul qiluvchini "bo'g'ib qo'ymaslik".

Tiqilinchni boshqarish (Congestion Control)

Tarmoq yuklanishini kamaytirish uchun CWND orqali:

1. Slow Start
2. Congestion Avoidance
3. Fast Retransmit
4. Fast Recovery

algoritmari ishlatiladi.

TCP protokoli tarmoqdagi ortiqcha yuklanish (congestion)ni oldini olish va mavjud o'tkazuvchanlikni (bandwidth) samarali ishlatish uchun tiqilinchni boshqarish (congestion control) mexanizmlaridan foydalanadi. Shularning eng asosiy parametrlardan biri congestion window – CWND bo'lib,

u ma'lum bir vaqtda tarmoqda "yo'lda" bo'lishi mumkin bo'lgan tasdiqlanmagan (ACK olinmagan) baytlar yoki segmentlar sonini cheklaydi[4].

CWND qiymati qanchalik katta bo'lsa, jo'natilayotgan trafik shunchalik ko'p bo'ladi, ya'ni TCP oqimi tarmoqqa ko'proq yuk beradi. Tiqilinch aniqlanganda yoki paket yo'qotilishi yuz berganda CWND keskin qisqartiriladi, tarmoq barqaror bo'lsa, CWND asta-sekin oshiriladi.

CWND ning asosiy rejimlari: slow start va congestion avoidance

TCP Reno/klassik TCP uchun CWND vaqt bo'yicha o'zgarishi odatda ikki bosqichda qoraladi:

1. Slow start rejimi

TCPning boshlang'ich bosqichi bo'lgan slow start mexanizmi uzatish oynasining qiymatini geometrik tarzda oshirishga asoslanadi. Bu davrda har bir ACK signali tarmoqning hozircha yuklanishsiz ishlayotganini bildiruvchi musbat signal sifatida qabul qilinadi. Shuning uchun har bir tasdiq olinganda oynaning bir segmentga kengayishi tarmoqning real imkoniyatlari haqida tezkor ma'lumot beradi. Aynan shu sabab tufayli slow start vaqtida CWND qiymati RTT davomida bir marta emas, balki oynadagi segmentlar soni bo'yicha ko'paytiriladi[2].

Boshlanishida CWND odatda 1 MSS (Maximum Segment Size) yoki bir necha MSS qilib olinadi. Har bir ACK olinganda CWND oshib boradi. Har bir RTT davomida CWND taxminan ikki barobarga oshadi (eksponensial o'sish).

Agar CWND qiymati ssthresh (slow start threshold) deb nomlanuvchi chegaraga yetgach, slow start tugaydi va congestion avoidance bosqichi boshlanadi.

Matematik tarzda soddalashtirilgan ifoda (Slow start RTT bo'yicha diskret model) quyidagi 1-formulada keltirilgan[3]:

$$CWND_{k+1} = 2 \times CWND_k \quad (CWND_k < ssthresh) \quad (1)$$

bu yerda:

k – RTT bosqich raqami.

Formulaning $CWND_{k+1} = 2 \times CWND_k$ ko'rinishda bo'lishi TCPning har bir segmentdan qaytadigan ACKni mustaqil kengaytiruvchi faktor sifatida qabul qilishidan kelib chiqadi. Agar oynada 4 ta segment bo'lsa, 4 ta ACK qaytadi, 8 ta segment bo'lsa — 8 ta ACK qaytadi. Shu tariqa oynaning kengayishi qo'shilish orqali emas, balki segmentlar soni bilan proporsional ravishda yuz beradi. Natijada CWND qiymati klassik geometrik progressiya ko'rinishida 1, 2, 4, 8, 16 kabi eksponensial o'sishga ega bo'ladi. Bu TCPga juda qisqa vaqt ichida tarmoqning yuqori qismigacha ko'tarilib borish imkonini beradi.

Slow start mexanizmining nazariy maqsadi tarmoqning haqiqiy sig'imi haqida tezkor va minimal kechikishli baho olishdir. TCP ulanish o'rnatilgan paytda tarmoqning qaysi darajada yuklanganini, qancha ma'lumotni ko'tara olishini yoki qachon tiqilinch yuzaga kelishini bilmaydi. Shuning uchun dastlab ehtiyotkor qilib kichik CWND bilan boshlab, so'ngra imkoniyat mavjud bo'lsa juda tez sur'atda oshirib boradi. Eksponensial o'sish orqali tarmoqning sig'imi "qayerda to'xtashi kerak" ligini tizimning o'zi sstresh chegarasi yoki paket yo'qotilishi orqali bildiradi. Bu yondashuv TCPga yangi ulanishni samarali boshlash imkonini beradi[5].

Shuni ta'kidlash kerakki, slow start hech qachon cheksiz davom etmaydi, u faqat tarmoqning boshlang'ich bo'sh fazosini sinab ko'rish uchun mo'ljallangan. CWND sstresh qiymatiga yaqinlashgan sari tizim eksponensial o'sishdan voz kechib, yanada ehtiyotkor bo'lgan chiziqli o'sish bosqichiga o'tadi. Bu bosqich tarmoqning optimal yuklanish nuqtasini topish va ortiqcha yuklanish sodir bo'lmasligini ta'minlaydi. Eksponensial o'sishning qisqa, ammo kuchli dinamikasi slow startning eng asosiy nazariy qiymatini tashkil etadi. Tarmoq sig'imini aniqlashni sekinlashtirmasdan, lekin haddan tashqari yuklamasdan boshqarish[4].

Formulada berilgan ikki baravar oshish mexanizmi TCPni boshqa transport protokollaridan ajratib turadigan muhim mexanizmdir. U tizimning adaptivligini, tarmoq sharoitlariga tez moslasha olishini va minimal vaqt ichida maksimal samaradorlikka chiqishini ta'minlaydi. Slow startning aynan eksponensial xususiyati TCPning yillar davomida samarali ishlayotgan asosiy sabablaridan biri bo'lib, tarmoq texnologiyalarining kengayishi bilan hanuzgacha dolzarbligini yo'qotmagan[5].

2. Congestion avoidance (AIMD – Additive Increase, Multiplicative Decrease)

Bu bosqichda CWND eksponensial emas, chiziqli o'sadi. Har bir RTT davomida CWND taxminan 1 MSS ga ortadi (agar yo'qotish bo'lmasa). Bu additive increase qismi.

Soddalashtirilgan formulasi (RTT bo'yicha) Congestion avoidance quyidagi 2-formulada keltirilgan[4]:

$$CWND_{k+1} = CWND_k + 1 \quad (2) \quad (CWND_k \geq ssthresh, \text{yo'qotishsiz})$$

Paket yo'qotilishi aniqlanganda (masalan, timeout yoki 3 dupACK holatlarida) multiplicative decrease qo'llanadi:

$$CWND \leftarrow \frac{CWND}{2} \qquad ssthresh \leftarrow \frac{CWND}{2}$$

Yangi qiymat bilan odatda congestion avoidancedan yoki qayta slow startdan davom etiladi (variantga qarab).

TCP ning congestion avoidance bosqichida oynaning kengaytirilishi eksponensial emas, balki juda ehtiyotkor **additiv** shaklda amalga oshiriladi. Bu model TCPning tarmoqdagi yuklanishni ortiqcha oshirmasdan, mavjud o'tkazuvchanlikdan foydalanish darajasini asta-sekin kengaytirishga intilishini ko'rsatadi. Asta-sekinlik bu yerda muhim, chunki tarmoq real sig'imiga yaqinlashgan sari kichik o'zgarishlarning o'zi ham muvozanatni buzishi va tiqilinchga olib kelishi mumkin[5].

Additiv oshirishning nazariy mohiyati shundan iboratki, CWND qiymati endi ACKlar soniga bog'lanmaydi. Bu bosqichga kelib TCP tarmoqni

yetarli darajada sinab bo'lgan deb hisoblaydi va har bir RTT uchun oynani bir birlikka kengaytiradi. Ana shunda tarmoqning haqiqiy barqaror nuqtasi sekin va barqaror qadamlar bilan sinovdan o'tkaziladi. Eksponensial o'sishdan voz kechilganligi TCPning ehtiyotkor yondashuvini aks ettiradi: tizim endi "tarmoqni topish" emas, balki uni "qo'llab turish" jarayoniga o'tadi.

AIMD modelidagi additiv qism aynan shu formulaga asoslangan bo'lib, tarmoq sig'imini topishi bilan uni optimallashtirilgan tezlikda kengaytirib boradi. Bunda tarmoqqa har RTTda bittadan segment qo'shish mavjud sig'imning haddan tashqari oshib ketishiga yo'l qo'ymaydi, shu bilan birga tarmoqda bo'sh joy mavjud bo'lsa, asta-sekin foydalanishga imkon beradi. Bu yondashuv TCPni nozik balans holatiga yaqinlashtiradi, ya'ni paket yo'qotilishi sodir bo'lmasdan barqaror ishlashni davom ettiradi.

Formulaning matematik ahamiyati ham mavjud: CWND chiziqli o'sadi, RTT esa o'zgarmagan deb qaralsa, TCPning o'rtacha throughput qiymati ham chiziqli ravishda oshib boradi. Biroq bu o'sish tiqilinchga olib kelmaydigan eng xavfsiz tempda kechadi. Tarmoqdagi minimal buzilish belgilari (kechikishning ortishi, yo'qotish) sodir bo'lishi bilanoq bu formula to'xtaydi va tizim multiplicative decrease rejimiga qaytadi[6].

Additiv oshirish TCPning barqaror ishlashining asosi hisoblanadi. Agar bu mexanizm bo'lmaganida, tarmoq sig'imi yaqinida turib eksponensial o'sish davom etsa, tarmoq tez orada ortiqcha yuklanishga kirib borar edi. Shu sababli congestion avoidance bosqichidagi $CWND_{k+1}$ yondashuvi TCPni global Internet kabi ulkan tarmoqlarda ham o'z-o'zini muvozanatda ushlab tura oladigan mexanizmga aylantiradi.

CWND va o'rtacha tezlik o'rtasidagi bog'liqlik

TCP ning o'rtacha oqim tezligini juda soddalashtirilgan holda quyidagi 3-formulada keltirilgan[6]:

$$\text{Throughput} = \frac{CWND}{RTT} \quad (3)$$

ya'ni CWND qanchalik katta bo'lsa va RTT qanchalik kichik bo'lsa, TCP oqimi tezligi shunchalik yuqori bo'ladi. TCP oqimining o'rtacha tezligini ifodalashda qo'llaniladigan formula munosabati transport pog'onasidagi uzatish jarayonining eng fundamental xususiyatidan kelib chiqadi.

TCP har bir RTT davomida oynada ruxsat etilgan miqdordagi ma'lumotni - ya'ni CWND ga teng bo'lgan segmentlar to'plamini - tarmoqqa yuborishi va ularning tasdiqlanishini kutishi mumkin. Ulanishning ishlash mexanizmi shundan iboratki, yangi ma'lumot faqat avvalgi ma'lumotlar bo'yicha ACK olingandan keyin yuborilishi mumkin, demak ma'lumot oqimining tezligi aynan shu ikkita parametr - oynaning kengligi va tasdiqlanish siklining davomiyligi bilan chegaralanadi[5].

Bu tenglik TCPning ishlash jarayonini vaqt bo'yicha diskret bo'linmalarga ajratish orqali tushuntiriladi. Agar tarmoqning RTT qiymati ma'lum bo'lsa, TCP yaqinlashib kelayotgan ACKlar yordamida ma'lumot uzatish siklini har bir RTTda bir marta yangilaydi. Agar oynaning kengligi CWND birlik bo'lsa, har bir to'liq RTT davomida aynan shu miqdordagi ma'lumot uzatilishi mumkin. Shunday qilib, bir sikl davomida uzatilgan ma'lumot miqdori CWNDga, siklning davomiyligi esa RTTga teng bo'ladi. Natijada sikl boshiga to'g'ri keladigan o'rtacha tarmoqqa uzatilayotgan ma'lumot miqdori CWND/RTT nisbatida aniqlanadi.

Bu tenglikning nazariy ahamiyati shundaki, TCPning tezligini oshirish uchun faqat ikkita yo'l mavjudligini ko'rsatadi: CWND ni kattalashtirish yoki RTTni kamaytirish. CWND ning o'sishi congestion control algoritmlariga bog'liq bo'lsa, RTT odatda tarmoqning topologik xususiyatlari bilan belgilanadi. Shu sababli TCPning yuqori tezliklarda ishlashiga erishish ko'p hollarda aynan congestion control mexanizmlarini takomillashtirish orqali amalga oshiriladi. Buning tasdig'i sifatida yillar davomida Reno, CUBIC, BBR kabi algoritmlar aynan CWND yoki RTT bilan bog'liq modelni optimallashtirishga qaratilgan.

Ushbu tenglik TCP oqimining muvozanat holatini ham tushuntiradi. CWND oshgan sari throughput ortadi, biroq CWND juda katta bo'lganda tiqilinch ehtimoli kuchayadi va RTT ortib boradi. Natijada CWND va RTT o'rtasida tabiiy muvozanat nuqtasi vujudga keladi. TCPning barcha tiqilinchni boshqarish algoritmlari aynan shu muvozanatni topishga xizmat qiladi. Shunday qilib, ushbu modelning nazariy mazmuni TCPning ishlash tamoyillarini soddalashtirilgan, ammo chuqur va aniq shaklda aks ettirishga asoslanadi[5].

Rasmiyrog matematik modellarda yo'qotish ehtimoli p , MSS va RTT kiritilgan holda quyidagiga o'xshash ifodalar uchraydi (4-formula)[7]:

$$\text{Throughput} \approx \frac{\text{MSS}}{\text{RTT} \times \sqrt{p}} \quad (4)$$

Biroq bizning holatda maqsad – CWND ning vaqt bo'yicha diskret o'zgarishini simulyatsiya qilish, shuning uchun soddalashtirilgan AIMD modelining o'zi yetarli.

Soddalashtirilgan matematik model

Diskret vaqt modeli sifatida har bir qadamni "bitta RTT" deb qaraymiz. Keling, quyidagi parametrlarni olamiz:

- CWND(1) – boshlang'ich CWND (MSS birliklarida), masalan 1.
- ssthresh – slow start thresholddan congestion avoidance ga o'tish nuqtasi, masalan 16.
- N – umumiy RTT bosqichlar soni.
- p_loss – yo'qotish ehtimolini namoyish qilish uchun oddiy ehtimollik parametri (har bir RTTda yo'qotish bo'lish ehtimoli).

Model quyidagicha ishlaydi:

1. Agar $\text{CWND}(k) < \text{ssthresh}$ va yo'qotish yo'q bo'lsa:
 $\text{CWND}(k+1) = 2 * \text{CWND}(k)$ (slow start).
2. Agar $\text{CWND}(k) \geq \text{ssthresh}$ va yo'qotish yo'q bo'lsa:
 $\text{CWND}(k+1) = \text{CWND}(k) + 1$ (congestion avoidance).

3. Agar yo'qotish sodir bo'lsa (masalan, $\text{random} < p_{\text{loss}}$):
 $\text{CWND}(k+1) = \text{floor}(\text{CWND}(k)/2)$ va $\text{ssthresh} = \text{CWND}(k+1)$ qilib yangilanadi.

Bu model juda soddalashtirilgan, ammo CWND ning klassik “tishsimon” ko'rinishini, slow start va congestion avoidance davrlarini yaxshi ko'rsatadi[8].

Quyidagi Octave dasturida TCP Reno algoritmining nazariy formulalariga asoslanib CWND ning vaqt bo'yicha o'zgarishi modellashtiriladi.

Bosh parametrlarni aniqlash

clear all; close all; clc;

N = 200; % RTT bosqichlari soni

CWND = zeros(1,N); % CWND qiymatlari massivi

time = 1:N; % vaqt o'qi (RTT bo'yicha)

MSS = 1; % MSS birlik sifatida olindi

*CWND(1) = 1 * MSS; % boshlang'ich CWND = 1 MSS*

*ssthresh = 16 * MSS; % slow start threshold = 16 MSS*

p_loss = 0.05; % har bir RTTdagi yo'qotish ehtimoli = 5%

Bu qismda model uchun asosiy o'zgaruvchilar belgilanadi. CWND bosh qiymati an'anaviy tarzda 1 MSS dan boshlanadi. ssthresh TCP Reno modelida slow startdan congestion avoidance ga o'tish nuqtasini aniqlaydi. p_{loss} — yo'qotilish ehtimoli bo'lib, CWND ning keyingi sikldagi o'zgarish turini belgilash uchun ishlatiladi.

Asosiy sikl: CWND ni yangilash

for k = 1:N-1

loss = (rand() < p_loss); % Yo'qotish sodir bo'ldimi?

if loss

CWND(k+1) = floor(CWND(k) / 2); % Multiplicative decrease

if CWND(k+1) < 1

```

        CWND(k+1) = 1;
    end
    ssthresh = CWND(k+1); % yangi threshold
else
    if CWND(k) < ssthresh
        CWND(k+1) = 2 * CWND(k);    % Slow start (eksponensial
o'sish)
    else
        CWND(k+1) = CWND(k) + 1*MSS; % Congestion avoidance
(additive increase)
    end
end
end
end

```

Bu qism TCP Reno algoritmining yuragi bo'lib, unda CWND har bir RTT uchun yangilanadi:

- Yo'qotish bo'lganda:
 CWND ikkiga bo'linadi (multiplicative decrease) - bu AIMD modelining "MD" qismi.

- Yo'qotish bo'lmasa:
- Agar $CWND < ssthresh \rightarrow$ Slow Start rejimi ishlaydi (2 baravar o'sish).
- Agar $CWND \geq ssthresh \rightarrow$ Congestion Avoidance rejimi ishlaydi (har RTTda +1 MSS).

Bu mexanizm TCPni tarmoq sig'imiga moslashishga majbur qiladi.

Natijani grafikda chiqarish

figure;

plot(time, CWND, "-o");

grid on;

title("TCP Congestion Window (CWND) ning vaqt bo'yicha o'zgarishi");

xlabel("Vaqt (RTT bosqichlari)");

ylabel("CWND (MSS birliklarida)");

Bu qism CWND ning vaqt bo'yicha qanday o'zgarishini grafikda ko'rsatadi. Natija TCP Reno algoritmiga xos tishsimon (sawtooth) shaklga ega bo'ladi. Bu TCPning AIMD mexanizmi barqaror ishlab turganini bildiradi.

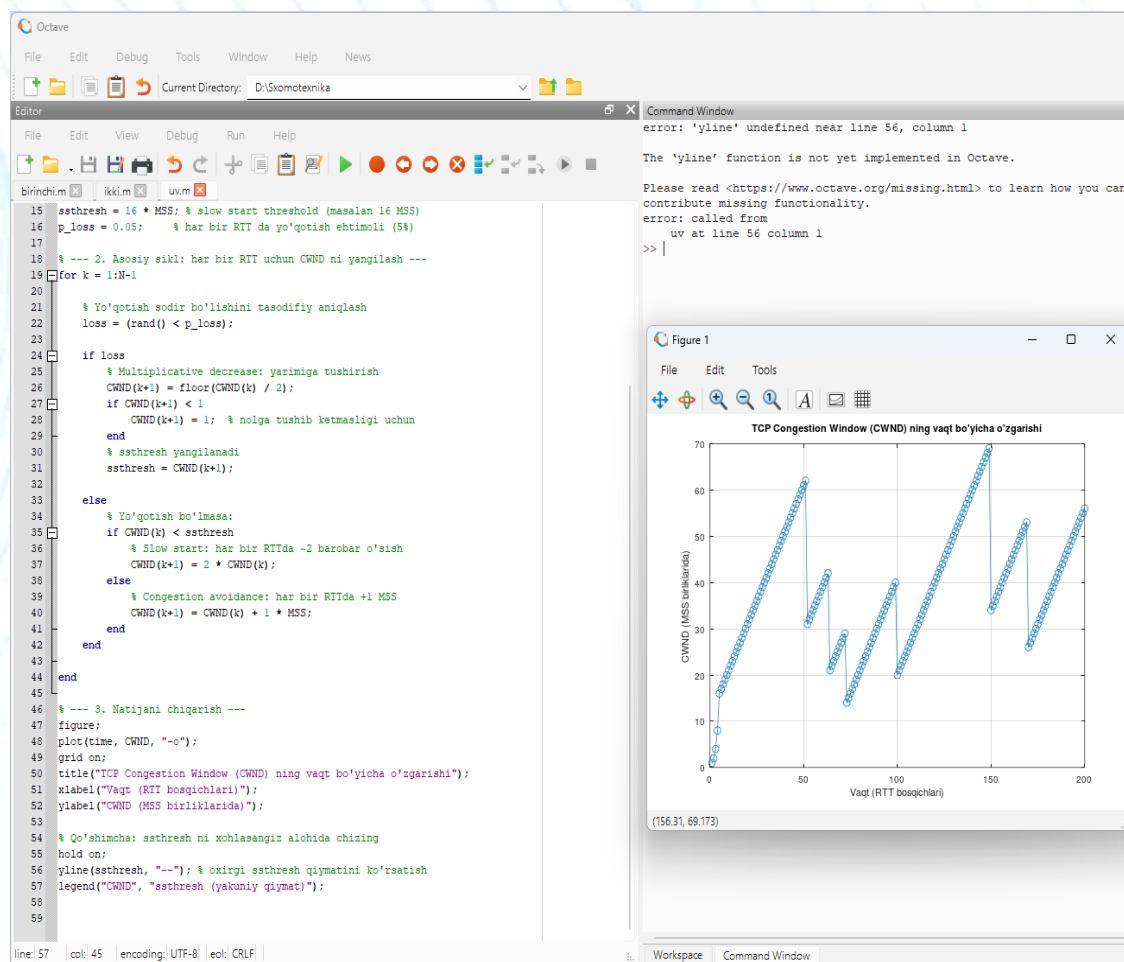
ssthresh chizig'ini ko'rsatish (ixtiyoriy element)

hold on;

yline(ssthresh, "--");

legend("CWND", "ssthresh (yakuniy qiymat)");

yline() funksiyasi yordamida threshold (ssthresh) qiymati grafikda ko'rsatiladi. Bu threshold TCP qaysi nuqtada slow startdan congestion avoidance ga o'tganini aniqlash imkonini beradi (1-rasm).



1-rasm. TCP Reno modelida CWND ning vaqt bo'yicha o'zgarish grafigi

Xulosa

Ushbu maqola davomida TCP Reno algoritmidagi qo'llaniladigan congestion window (CWND) parametrining vaqt bo'yicha o'zgarish qonuniyatlari nazariy formulalar va GNU Octave muhitida bajarilgan simulyatsiya asosida o'rganildi. Slow start bosqichida CWND ning eksponensial o'sishi $CWND_{k+1} = 2 \times CWND_k$ modeli orqali tasdiqlandi va bu bosqich TCPning tarmoq sig'imini tezkor baholashi hamda mavjud bo'sh o'tkazuvchanlikdan samarali foydalanishini ta'minlashi amaliy ko'rinishda namoyon bo'ldi. CWND ssthresh chegarasiga yetgach, tizim chiziqli o'sish bosqichiga o'tib, $CWND_{k+1} = CWND_k + 1$ formulasi asosida ehtiyotkor, barqaror o'sish amalga oshishi grafikda yaqqol ko'rindi.

Simulyatsiya jarayonida tasodifiy paket yo'qotilishi modellashtirishda CWND ning multiplicative decrease mexanizmi ishga tushib, qiymatning keskin kamayishi ($CWND/2$) kuzatildi. Bu TCP Reno algoritmining AIMD (Additive Increase, Multiplicative Decrease) xususiyati tarmoq tiqilinchining oldini olishda naqadar muhim rol o'ynashini amaliy jihatdan tasdiqladi. CWND grafigining tishsimon shaklga ega bo'lishi congestion control mexanizmlarining tabiiy oqibatlari ekanligi ko'rsatildi.

Octave orqali olingan natijalar TCPning real tarmoqlarda moslashuvchan va o'z-o'zini boshqaruvchi tizim sifatida ishlashini tahlil qilishga yordam berdi. CWND ning vaqt bo'yicha o'zgarishi orqali tizimning qaysi nuqtada tezkor kengayishi, qaysi holatlarda barqaror ishlashi va qachon keskin pasayish sodir bo'lishi ilmiy asosda izohlandi. Ushbu jarayon $\text{throughput} \approx CWND/RTT$ munosabati bilan bog'lanib, TCPning o'rtacha o'tkazuvchanligi qanday shakllanishi amaliy ravishda tushuntirildi.

Umuman olganda, maqola TCP protokolining congestion control mexanizmlarini chuqurroq anglash, CWND parametrining tarmoq barqarorligiga ta'sirini baholash va modellashtirish asosida olingan natijalarni tahlil qilish ko'nikmalarini shakllantirdi. Bu tajriba transport protokollarining ishlash tamoyillarini chuqur tushunishga zamin yaratadi va kelgusida yuqori samarali tarmoq algoritmlarini loyihalashda nazariy asos bo'lib xizmat qiladi.

FOYDALANILGAN ADABIYOTLAR

1. Van Jacobson, "Congestion Avoidance and Control," *SIGCOMM*, 1988 — TCP Reno algoritmining nazariy asoslarini birinchi bo'lib taklif qilgan klassik ilmiy maqola.
2. Kurose, J. & Ross, K., *Computer Networking: A Top-Down Approach*, Addison-Wesley, 2021 — TCP protokoli, congestion control va CWND dinamikasi haqida batafsil tavsif.

3. Tanenbaum, A. & Wetherall, D., *Computer Networks*, Pearson, 2019 — transport protokollari, TCP ishlash mexanizmlari va AIMD modeli bo'yicha asosiy nazariy manba.
4. Mathis, M., Semke, J., Mahdavi, J., "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *Computer Communication Review*, 1997 — TCP throughputning matematik modeli haqida ilmiy asos.
5. RFC 5681 — TCP Congestion Control — TCP Reno, Tahoe va congestion control algoritmlarining rasmiy spetsifikatsiyasi.
6. GNU Octave Documentation, "Plotting and Simulation Functions" — modellashtirish va grafik hosil qilish bo'yicha texnik manba.
7. Stevens, W., *TCP/IP Illustrated, Volume 1*, Addison-Wesley — TCP segmentlari, ACK mexanizmi, CWND xatti-harakati va RTT asoslari bo'yicha fundamental qo'llanma.
8. Comer, D., *Internetworking with TCP/IP*, Prentice Hall — TCP ulanishi, oqimni boshqarish va paket yo'qotilishi ta'siri bo'yicha nazariy tushuntirishlar.