

PROIECT CONGESTION CONTROL

-2024-

Nume echipă: $O(n^3)$

Membri: Chira Vlad-Andrei 324CB ~ cod de bază, script optimizare parametri

Vlădulescu Denis-Marian 324CB ~ implementare timeout, actualizare `tcp_friendliness`

Savin Ana-Bianca 324CB ~ analizarea și realizarea documentației

Marcu Diana 324CB ~ analiza rezultatelor obținute

Algoritm ales: Cubic

Descrierea implementării:

Implementarea echipei noastre se bazează pe algoritmul standard CUBIC[1][2], căruia i-au fost aduse câteva modificări astfel încât să se potrivească scenariului propus în cadrul proiectului.

În primul rând, în mod empiric, am modificat parametri β (*constanta de descreștere a ferestrei*) și C (*parametrul CUBIC*), atribuindu-le acestora valorile de 0.5, respectiv 0.2 (valorile generale indicate fiind 0.2 respectiv 0.4).

Coeficientul β ne indică procentul cu care se va reduce fereastra în momentul când detectăm pierdere de pachete, $cwnd = cwnd * (1 - \beta)$, ceea ce va conduce către o reducere de 50% cu parametrul curent (și de 80% considerând parametrul indicat în documentație).

Coeficientul C este un coeficient al funcției cubice care determină creșterea dimensiunii ferestrei. Am decis să-l reducem întrucât în valoarea sa inițială inducea o creștere mult prea rapidă care, de cele mai multe ori, ducea, în primii pași, chiar la reducerea dimensiunii ferestrei în loc să aducă o creștere.

Având acești parametri stabiliți, flow-ul algoritmului se folosește de **slow start**, care scade în viteza de creștere cu cât se apropie mai mult de punctul în care s-au înregistrat pierderi ultima dată. Dacă se ajunge în acel punct și nu se pierd pachete, se realizează o creștere modelată cu o funcție de grad 3, cubică, mai exact $(t - K)^3$.

Pe parcursul executării algoritmului, se actualizează în timp real RTT-ul pachetelor, stocându-se, pe parcursul a 250 de ACK-uri (echivalentul unei runde). În cazul în care un RTT depășește un anumit prag ($base_{rtt} * 1.1075$), atunci considerăm apariția unui timeout și se apelează funcția de resetare a parametrilor cubic la valorile de bază.

Tot pe parcursul executării algoritmului, se stochează numărul de pachete trimise și numărul de pachete neprimite (pentru care nu s-a recepționat ACK), iar în funcție de acești parametri se activează/dezactivează valoarea booleană `tcp_friendliness` pe baza unui

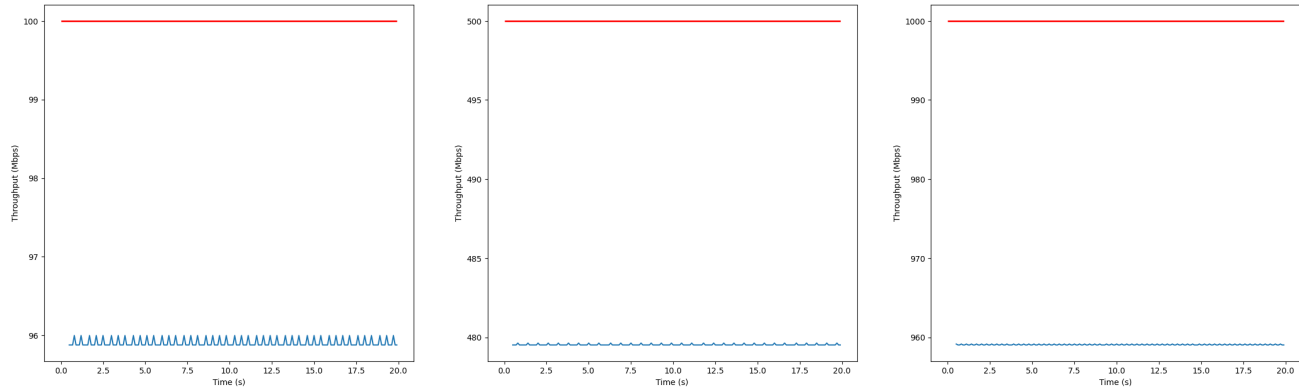
$$loss_rate = \frac{total_lost_packets}{total_sent_packets}$$

Pe lângă ajustarea parametrilor, am adăugat, în cadrul funcției de procesarea a unui ACK, o condiție care verifică dacă pachetul primit este marcat ECN, caz în care se va reduce dimensiunea ferestrei, dar cu o reducere mai puțin agresivă decât în cazul unui loss (în cazul în care pachetul este marcat ECN, scăderea va fi dată de ecuația $cwnd = cwnd * (1 - \frac{\beta}{2})$). Dacă pachetul nu este marcat ECN, se încearcă creșterea ferestrei de congestie :

- dacă încă suntem sub ultimul punct de loss, creștem exponențial
- dacă am ajuns să depășim, determinăm cu cât vom crește, ieșind din starea de slow start.

Rezultatele obținute:

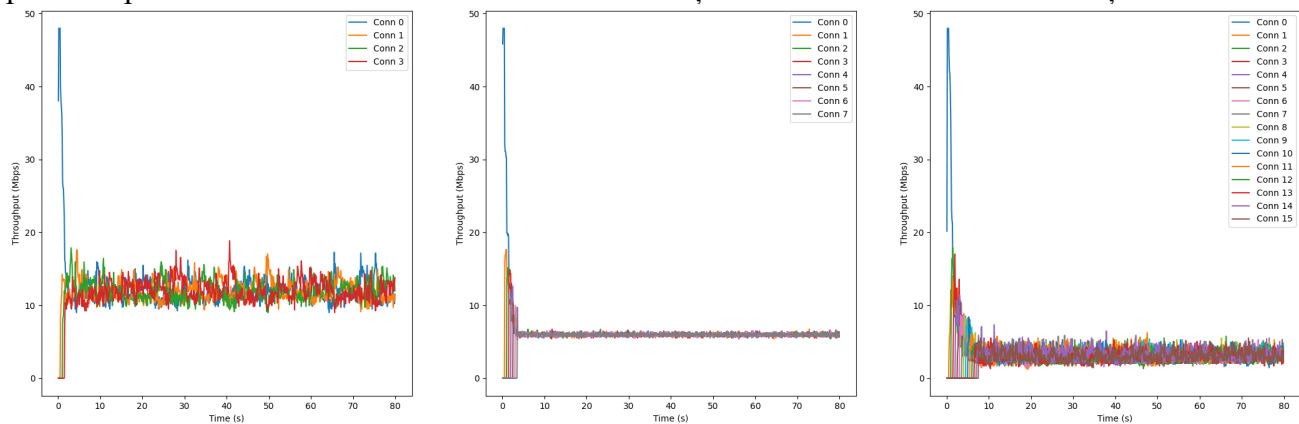
I. Viteza de transmisie(THROUGHPUT)



În cadrul testului de throughput se observă cum se păstrează, în toate cele 3 scenarii, o viteză egală cu aproximativ 95% din target-ul impus.

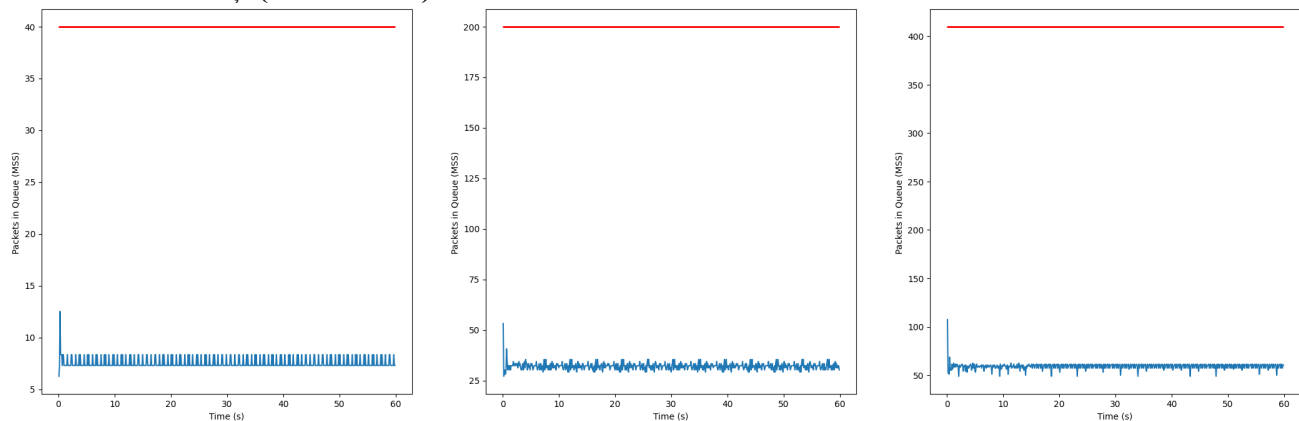
II. Corectitudinea(FAIRNESS)

În cadrul acestui test, se verifică dacă modul în care algoritmul împarte lățimea de bandă este pe cât se poate de corect atunci când intervin mai mulți utilizatori activi simultan în rețea.



Se observă cum la început, când este doar un client activ, throughput-ul este maxim, iar ulterior, când intervin și alți clienți, acesta se reduce astfel încât aceștia să împartă în mod egal lățimea de bandă.

III. Latență(LATENCY)



Se poate observa cum se menține un număr pseudoconstant de pachete în coadă egal cu aproximativ 20% din limita superioară a cozii de pachete.

DOCUMENTAȚIE:

[1] [https://github.com/govindjeevan/Cubic-TCP/blob/master/CUBIC TCP.pdf](https://github.com/govindjeevan/Cubic-TCP/blob/master/CUBIC%20TCP.pdf)

[2] <https://web.archive.org/web/20150726051512/http://www4.ncsu.edu/%7Erhee/export/bitcp/cubic-paper.pdf>