

Proiect Congestion Control PCom

Proiectul propune implementarea unui algoritm de control al congestiei, menit să maximizeze performanța și fairness-ul, și să minimizeze latența per pachet. Proiectul se desfășoară pe echipe; fiecare echipă poate conține între 2 și 4 persoane. Puteți dezvolta propriul vostru algoritm sau utiliza un algoritm existent cu condiția ca algoritmul **să nu fie implementat de mai mult de 4 echipe de la seria voastră** (first come first served). Puteți alege atât algoritmi ce se pot implementa doar la endhosts, cât și cu sprijin adițional în rețea, prin [Explicit Congestion Notification \(ECN\)](#). **Va exista un clasament live al implementărilor.**

Implementarea algoritmului va fi testată într-o topologie de tip [dumbbell](#), în care avem un număr variabil de transmițători și receptori (sink), care împart o legătură de date ce va limita performanța (adică va constitui un punct de restricție, bottleneck).

Implementarea o vom face în simulatorul **HTsim**. Vom lucra în fișierul **cc.cpp**, unde se găsește o implementare de sender/receiver. În mare vom lucra în funcțiile `CCSrc::processAck()` și `CCSrc::processNack()`. Mai multe detalii sunt prezente sub forma de comentarii în schelet. Scheletul vine cu o implementare a algoritmului [TCP NewReno](#) și o variantă simplistă a [explicit congestion notification](#). Trebuie să înlocuiți acest cod cu implementarea algoritmului ales de voi.

Un overview al NewReno poate fi găsit în [notitele de curs de la seria CB](#) și înregistrările cursului:

- Curs 1: [partea 1](#) [partea 2](#)
- [Curs 2.](#)

Alte materiale relevante pentru proiect sunt [Network Congestion - pana la 17 \(20 min\)](#) și [Congestion control](#).

Odată implementat, algoritmul va fi rulat cu diferite viteze pentru legătura de date și cu diferite latențe. Dimensiunea cozii va fi setată la un multiplu al [bandwidth-delay product](#) (BDP), iar pachetele vor fi marcate cu **ECN** atunci când coada depășește **BDP**-ul.

Pentru a vizualiza comportamentul algoritmului, vom folosi o serie de scripturi Python într-un Jupyter Notebook.

Aveți la dispoziție următoarele resurse:

- Schelet tema + notebook: <https://gitlab.cs.pub.ro/pcom/congestion-control-project>
- Website clasament: adresa IP la care veți putea vedea un clasament live este pe Moodle

Înscriere Echipa

Înscrierea echipei se va face de către coordonator pe forum conform instrucțiunilor din thread-ul creat pentru înscrieri.

Fiecare echipă trebuie să selecteze un algoritm de control al congestiei pe care dorește să îl implementeze. Puteți alege orice algoritm doriți să implementați din lista de mai jos, cu condiția ca algoritmul să nu fie implementat de mai mult de patru echipe din seria voastră. Alocările de algoritmi se vor face în ordinea sosirii (first come, first served).

O lista posibila de astfel de algoritmi, împreuna cu descrierea lor detaliată, este următoarea:

- [Cubic](#)
- [Scalable TCP](#)
- [Swift](#)
- [Vegas](#)
- [BBR](#)
- [Compound TCP](#)
- [DCTCP](#)
- [NDP](#)

PS: Oricine poate participa, dar doar cei de la seriile CB si CD vor fi punctati pentru activitatea de la curs.

Trimiterea și notarea proiectului

Coordonatorul echipei va încărca proiectul pe platforma Moodle, unde acesta va fi rulat și înscris în clasament. Arhiva trebuie să conțină și un document de prezentare în format PDF care să expună soluția voastră și rezultatele sub formă de grafice și interpretări. Nota finală va fi calculată în funcție de poziția în clasament, prezentarea finală și de rezultate. Evaluarea va avea loc în cadrul ultimului curs, unde fiecare echipă va face o scurtă prezentare de 5-10 minute a implementării.