Tema de casa

2014

Retele de calculatoare

*Combaterea congestiei in TCP cu ECN*

 STUDENTI: Teodor Chirica

 Badea Cristian Alexandru Victor

 GRUPA: 441A

**CUPRINS:**

**Capitolul 1**

1.Introducere generala – Cristian Badea

**Capitolul 2**

2.1 Introducerea gateway-urilor RED pentru imbunatatirea comportamentului TCP pe durata congestiilor – Teodor Chirica

2.2 Introducerea mecanismului ECN – Teodor Chirica

 2.2.1 Header-ul ECN-TCP

 2.2.2 Gateway-urile ECN-TCP

 2.2.3 Sursele ECN-TCP

 2.2.4 Receptorul ECN-TCP

**Capitolul 3**

 3. Concluzii – Cristian Badea

**Capitolul 4**

 4. Bibliografie – Cristian Badea

***Capitolul 1***

*Introducere generala*

 In aceasta lucrare este prezentat raspunsul protocolului TCP la mecanismele ECN ( Explicit Congestion Notification) si exploreaza efectele mecanismelor ECN asupra performantelor retelelor TCP/IP. De asemenea, sunt prezentate anumite implementari legate de mecanismele ECN.[1]

 In retelele TCP/IP, TCP-ul se bazeaza pe pachetele aruncate ca semn de aparitie a congestiei. O sursa de pachete (sursa TCP) detecteaza pachetele aruncate fie prin receptia a trei mesaje ACK duplicate, fie prin expirarea timpului de retransmitere si raspunde unei asemenea situatii prin micsorarea ferestrei de congestie.

 Mecanismele de control prezente in TCP, “slow start” si “congestion avoidance” se folosesc de pierderea de pachete sau de expirarea anumitor timpi pentru a descoperi congestia din retea. De obicei, pierderile apar cand buffer-ul prezent la gateway-ul retelei isi atinge capacitatea maxima de procesare de pachete, iar fiecare pachet care soseste este aruncat.[1] In TCP nu exista mijloace de a descoperi aparitia congestiei inainte ca pachetele sa se piarda. Aceasta detectare a pierderilor de pachete cauzeaza o problema globala de sincronizare a statiilor care transmit pachete spre acelasi gateway. Cand un gateway incepe sa arunce pachete, majoritatea statiilor observa cam in acelasi timp ca pachetele transmise catre gateway sunt aruncate, fapt care cauzeaza o utilizare slaba a latimii de banda si cauzeaza un delay considerabil.

***Capitolul 2***

*2.1 Introducere gateway-urilor RED pentru imbunatatirea comportamentului TCP pe durata congestiilor*

 Pentru a imbunatati comportamentul protocolului TCP pe durata congestiilor , S. Floyd si V. Jacobson au introdus gateway-urile RED ( Random Early Detection ). Aceste gateway-uri incearca sa detecteze congestiile inainte ca in gateway-uri sa existe “queue overflow”- ul.

Aceasta metoda noua de detectie ajuta gateway-ul sa previna pierderea mare de pachete si problema sincronizarii globale. Un gateway RED poate arunca un pachet sau il poate marca cu un indicator de congestive pentru sursa pachetului.[2]

 Decizia de a arunca sau a marca un pachet depinde de protocolul folosit de sursa pachetelor. In cazul TCP-ului , gateway-ul va arunca pachetele deoarce TCP-ul observa congestia dupa aruncarea pachetelor. Pachetele sunt aruncate aleator in functie de dimensiunea medie a marimii cozii. Algoritmul de aruncare al pachetelor este bazat pe dimensiunea medie a cozii si nu pe dimenziunea cozii intr-un anumit moment pentru a evita o partinire a traficului de tip “burst”. Prin aruncarea la aleatoare a pachetelor, gateway-urile RED limiteaza sincronizarea de multiple stream-uri de trafic prin gateway. Prin folosirea gateway-urilor RED este posibila izolarea procesului de detectie a congestiei de cel de notificare a congestiei.

*2.2 Introducere mecanismului ECN*

Pentru a imbunatati comportamentul protocolului TCP la congestii, S. Floyd si K. Ramakrishnan au propus mecanismul de Notificare Explicita a Congestiei (ECN). [1]

Scop-ul ECN-ului este de a evita delay-ul in transmisiuni cauzat de aruncarea inutila de pachete. Hostul sursa primeste un feedback de congestive din retea prin pachetele “marcate”. Un pachet este marcat de un gateway RED prin setarea unui bit in headerul TCP in loc sa fie aruncat. Marcajul este anuntat de receptorul TCP prin setarea unui “ECN echo flag” in headerul unui pachet ACK iar sursa reactioneaza la pachetul marcat la fel cum ar reactiona la un pachet ce trebuie aruncat. [3]

Avantajul acestui mechanism este ca nu depinde de timpii de retransmitere si de granularitatea timpilor TCP. Cu suportul ECN, TCP nu trebuie sa astepte expirarea timpilor pentru a reactiona la congestive. Pentru stream-uri de pachete cu Round-Trip Time (RTT) mic, granularitatea timpilor TCP poate intarzia detectia unui pachet aruncat, cu rezultatul surse in mod “idle” si a legaturii subutilizate. ECN in combinatie cu gateway-urile RED se axeaza pe rezolvarea problemelor de pachete aruncate inutil si a intarzierilor inutile in retea cauzate de pierderile de pachete mentionate mai sus, dar nu poate garanta impartirea egala a latimii de banda intre host-uri.



Fig.1: Bitii ECN din header-ul IP

TCP sugereaza ca o retea este congestionata doar prin pierderea de pachete care este indicate de expirarea timpilor de retransmitere sau a pachetelor ACK duplicate.[4]

 Acest mecanism poate fi o modalitate destul de “scumpa” pentru a detecta prezenta congestiei. La folosirea mecanismului ECN, pachetele sunt marcate in loc sa fie aruncate.[4]

 Acest marcaj informeaza sursa despre prezenta congestiei astfel incat sa nu mai fie nevoie sa astepte receptionarea ACK-urilor duplicate sau a expirarii diversilor timpi TCP. Din aceasta cauza, conexiunile mai sensibile la delay au de castigat de pe urma mecanismului ECN.[3]

 In general, TCP ar trebui sa raspunda unui pachet marcat ECN ca unui pachet pe care urmeaza sa il arunce. Asta inseamna ca TCP injumatateste fereastra de congestive (cwnd) si reduce pragul de “slow start” (sstresh).

*2.2.1. Header-ul ECN-TCP*

 ECN-TCP se foloseste de doi biti din header:

* Un bit arata ca o conexiune este capabila ECN (ECT-bit)
* Al doilea bit indica congestia (CE-bit)

Daca o conexiune foloseste ECN pentru controlul congestiei, bitul ECT este setat 1 in toate pachetele; altfel este setat 0.

Bitul CE este setat pe un router congestionat. ECN introduce doua flaguri noi in campul rezervat din headerul TCP. Primul flag este “ECN echo flag” care este setat in ACK de catre hostul care a primit pachetul daca un pachet ECN are bitul CE setat. Al doilea flag este “congestion window reduced” (CWR). Acest flag este setat de sursa dupa ce si-a redus dimensiunea ferestrei de congestie din motive diverse ( expirarea timpilor de retransmisie, ACK-uri duplicate sau ECN echo ACK).[5]

*2.2.2. Gateway-urile ECN-TCP*

ECN-TCP foloseste gateway-uri RED cu algoritm modificat pentru a seta bitul CE in headerul TCP. Diferenta este ca gateway-ul marcheaza pachetul in loc sa il arunce. [3]

Algoritmul RED incearca sa mentina lungimea medie a cozii intre doua praguri. Niciun pachet nu este marcat cat timp lungimea cozii este sub un prag minim ($min\_{th}$). Daca lungimea medie a cozii depaseste pragul minim, dar se afla sub pragul maxim ($max\_{th}$), pachetele care vin sunt marcate cu probabilitatea $p\_{drop}$, unde $p\_{drop}$ este calculate in functie de lungimea medie a cozii . Imediat ce lungimea medie a cozii depaseste pragul maxim, fiecare pachet care vine este aruncat indifferent daca este ECN sau nu. Lungimea medie a cozii este o functie ponderata exponentiala a dimensiunii cozii q cu ponderea $w\_{q}$. Algoritmul RED este prezentat mai jos :

for each packet arrival {

 calculate the average queue size $q\_{avg}$

 if $min\_{th}$<$q\_{avg}$<$max\_{th}$ {

 calculate probability $p\_{drop}$

 with probability $p\_{drop} $mark incoming packet

 }

 else if $max\_{th}$<=$q\_{avg}$

 drop the arriving packet

}

*2.2.3. Sursele ECN-TCP*

In faza de setare a conexiunii, sursa si destinatia trebuie sa negocieze capabilitatea de ECN. Acest fapt este realizat prin setarea CWR si ECN echo flag in pachetele SYN de catre sursa si de setarea ECN echo flag in pachetul SYN-ACK de catre receptor. [2]

In cazul in care doar sursa este capabila de ECN, in conexiune nu se vor folosi bitii ECT si CE. Altfel, bitul ECT terbuie setat de sursa in fiecare pachet si trebuie sa reactioneze la fiecare ECN echo ACK.[2]

Sursa trateaza un ECN echo ACK ca pe un pachet pierdut fara sa retransmita pachetul marcat. Cu toate ca acest ACK recunoaste un pachet de date, nu se creste dimensiunea ferestrei de congestie. Daca sursa primeste mai multi indicatori de congestie( timeout-uri, ACK-uri duplicate, ECN echo ACK), va reactiona o singura data la fiecare RTT la indicatorii de congestie. In cazul in care un pachet retransmis este marcat sau aruncat, sursa trebuie sa reactioneze la indicatorii de congestie din nou.

Nu se reduce pragul de “slow start” daca acesta a fost redus la ultimul RTT. Dup ace sursa raspunde la un indicator de congestie , seteaza flagul CWR in urmatorul pachet pe care il trimite dupa reducerea ferestrei.[5]

*2.2.4. Receptoarele ECN-TCP*

Receptorul trimite indicatorii de congestie a unui pachet cu bitul CE setat 1 inapoi la sursa. Dupa ce a primit un pachet cu bitul CE setat 1, seteaza ECN echo flag a pachetului urmator. Receptorul continua sa seteze echo flags pana cand primeste un pachet de date cu flagul CWR setat.

***Capitolul 3***

3.Conlcuzii

ECN reduce numarul de pachete aruncate de o conexiune TCP, care, prin evitarea retransmiterilor reduce latenta si fluctuatiile in special. Acest efect este foarte drastic cand o conexiune TCP are un singur segment si este capabila sa evite un timeout RTO; acesta este de obicei cazul pentru conexiunile interactive cum ar fi login-urile de la distanta si protocoalele tranzactionale(HTTP, faza conversationala a SMTP-ului sau cererile SQL).

Un alt avantaj al mecanismului ECN este faptul ca notifica intr-un timp mult mai scurt hosturile de aparitia congestiei decat in comportarea normala a protocolului TCP fara ECN.

Un dezavantaj al acestui mecanism apare atunci cand se pierd pachetele ECN in retea iar notificarile de congestie nu mai ajung la nodurile terminale. In aceasta situatie, gateway-urile RED continua sa marcheze aleator pachete ECN pe durata congestiei

***Capitolul 4***

4.Bibliografie

[1] - ***S.*** Floyd, “TCP and Explicit Congestion Notification,” ACM Computer Communication Review, October 1994.

[2] - **V.** Jacobson, “Modified TCP Congestion Avoidance Algorithm, April 1990

[3] - H. Krishnan, “Analyzing Explicit Congestion Notification (ECN) Benefits

for TCP,” 1998.

[4] - ***S.*** Floyd and K. Ramakrishnan, “A Proposal to add Explicit Congestion

Notification (ECN) to I**P,”**R FC 2481, January 1999.

[5] - V. Jacobson, “Congestion Avoidance and Control,” August 1998.