

Tema: Algoritmi de tratare a congestiei (partea 3/3)

Student : Caraivan George-Alexandru 441A

Tehnici RED de control al congestiei

Tehnicile de evitare a congestiei monitorizează debitul traficului prin rețea într-un efort de a anticipa și a preveni congestia în locurile cu gâtuiți. Evitarea congestiei se realizează prin aruncarea de pachete iar unul din cele mai cunoscute mecanisme este RED și este optim pentru rețelele tranzit de viteză mare.

Mecanismul RED a fost propus de Sally Floyd și Van Jacobson la începutul anilor 1990 pentru a rezolva problema congestiei într-o manieră mai mult responsabilă decât reactivă. Mecanismul RED se bazează pe faptul că majoritatea traficului folosește implementari la nivel de transport de date care sunt sensibile la pierderi și care vor încetini temporar când o parte din trafic este aruncat. TCP de exemplu răspunde într-un mod robust la acest lucru prin încetinirea transmisiei folosind mecanismul RED pe post semnal declanșator.

Deoarece TCP este cel mai folosit protocol de nivel transport RED oferă un mecanism foarte efektiv de evitare a congestiei. Nu se recomandă folosirea RED atunci când o porțiune a traficului nu este robustă precum cele care folosesc procoale de transport Novell sau Apple.

Algoritmi RED de control al congestiei

Algoritmul **RED** încearcă să controleze lungimea medie a cozii indicând surselor de trafic când ar trebui să își încetinească temporar transmisia pachetelor. Folosind mecanismele de control al congestiei ale TCP prin aruncarea pachetelor aleator înaintea momentelor în care se estimează congestia RED transmite sursei să încetinească transmisia pachetelor.

Probabilitatea aruncării unui pachet este definită de 2 praguri : min,max și de fracția de pachete aruncate atunci când lungimea medie a cozii este la pragul maxim. Când lungimea medie a cozii depășește pragul minim încep să fie aruncate pachete și rata de aruncare a lor crește liniar cu creșterea lungimii cozii până când se ajunge peste pragul maxim caz în care toate pachetele sunt aruncate.

Pragul de minim ar trebui setat suficient de sus astfel încât să fie maximizată folosirea legăturii, dacă el este setat prea jos pachetele pot fi aruncate deși nu ar trebui iar legătură nu va fi folosită eficient.

Diferența dintre pragul maxim și cel minim trebuie să fie suficient de mare astfel încât să poată fi evitată sincronizarea globală TCP. Dacă diferența este prea mică ar putea exista posibilitatea aruncării multor pachete la un moment dat fapt ce ar duce la o sincronizare globală.

Algoritmul RED este următorul :

```
inițializare : avg = 0, count = -1 ;
pentru fiecare pachet ce ajunge
    calculează lungimea medie nouă a cozii avg:
        dacă coada nu este goală
             $avg = (1 - w_q) * avg + w_q * q$ 
        altfel
             $m = f(\text{time} - q\_time)$ 
             $avg = (1 - w_q)^m * avg$ 
    dacă  $\text{min}_{th} \leq avg < \text{max}_{th}$ 
        incrementează count
        calculează probabilitatea  $p_b$ :
             $p_b = \text{max}_p * (avg - \text{min}_{th}) / (\text{max}_{th} - \text{min}_{th})$ 
             $p_a = p_b / (1 - \text{count} * p_b)$ 
        cu probabilitatea  $p_G$ :
            marchează pachetul ce ajunge
            count = 0
    altfel dacă  $\text{max}_{th} \leq avg$ 
        marchează pachetul ce ajunge
        count = 0
    altfel count = -1
cînd coada este goală
q_time = time
```

Unde:

avg: lungimea medie a cozii

q_time: începutul timpul de așteptare a cozii

count: pachete de la ultimul pachet marcat

w_q : lățimea cozii

min_{th} : pragul minim al cozii

max_{th} : pragul maxim al cozii

max_p : valoare maximă pentru p_b

p_a : probabilitatea curentă de marcare a pachetului

q: lungimea cozii curente

time: timpul curent

f(t): funcție liniară cu variabila de timp t

Deoarece lungimea medie a cozii este foarte sensibilă la nivelul de congestie și la parametrii RED înseamnă că ea nu poate fi estimată în avans. Cum întârzierea este o mare componentă a calității serviciului, operatorii rețelei ar să aibă o estimare apriorică a întârzierilor medii în routerele congestionate iar pentru a atinge aceste lucruri cu RED ar fi necesară o modificare constantă a parametrilor pentru a fi ajustați conform condițiilor de trafic curente. Prin **ARED** se dorește eliminarea sensibilității la parametrii care afectează performanța RED și se poate obține o lungime medie a cozii specifică într-o mare varietate de scenarii ale traficului.

Algoritmul ARED are ca scop adaptarea probabilității maxime (\max_p) de aruncare a pachetelor astfel încât lungimea medie a cozii să fie ținută între pragul minim (\min_{th}) și maxim (\max_{th}). Probabilitate de aruncare a pachetelor este adaptată încet cu pași mici și este constrânsă să fie între 0.01 și 0.5. În loc de creșterea și descreșterea multiplicativă a \max_p se va folosi politica AIMD.

Algoritmul ARED este următorul :

pentru orice interval :

dacă ($\text{avg} > \text{target}$ și $\max_p \leq 0.5$)
 $\max_p = \max_p + \alpha$;
altfel dacă ($\text{avg} < \text{target}$ și $\max_p \geq 0.01$)
 $\max_p = \max_p * \beta$;

Unde:

avg: lungimea medie a cozii

interval: 0.5 secunde

target: [$\min_{th} + 0.4 * (\max_{th} - \min_{th})$, $\min_{th} + 0.6 * (\max_{th} - \min_{th})$]

α : $\min(0.01, \max_p/4)$

β : 0.9

Robustețea ARED survine din ajustarea înceată și ocazională a \max_p iar prețul plătit pentru aceasta este că după o modificare abruptă a nivelului congestiei va dura 10-20 secunde până când \max_p se va adapta.

Din acest motiv există constrângerea de interval care face ca în această perioadă de tranziție performanțele generale să fie încă acceptabile deși întârzierea medie și debitul vor suferi puțin.

WRED combină posibilitățile algoritmului RED cu precedența ip pentru a dirija traficul preferențial al pachetelor cu prioritate ridicată. WRED poate arunca în mod selectiv traficul cu prioritate

scăzută când interfața începe să devină congestionată și oferă performanțe diferite pentru diferite clase ale serviciilor.

WRED este folosit în ruterele nucleu are rețelei și oferă praguri și ponderi diferite pentru precedente IP diferite putând asigura calități diferite ale serviciilor în ceea ce privește aruncarea de pachete pentru diferite tipuri de trafic.

WRED aruncă pachetele în mod selectiv bazat pe precedența ip. Pachetele cu o precedență ip mare au o probabilitate mult mai mică de a fi aruncate decât cele cu o precedență ip mică. În acest fel cu cât este mai mare prioritatea unui pachet cu atât este mai mare probabilitatea ca el să fie livrat la destinație.

Sursele care generează cel mai mult trafic vor fi sursele cel mai probabile de a fi încetinite. WRED tratează traficul non-ip cu cea mai mică precedență deci traficul non-ip este mult mai probabil să fie aruncat decât traficul ip.

Ruterul calculează automat parametrii care sunt folosiți în calculele WRED. Lungimea medie a cozii depinde de lungimea medie anterioară și de lungimea curentă a cozii după formula :

$$\text{lungimeamedie} = (\text{lungimeamedieanterioară} * (1 - 2^{-n})) + (\text{actualalungime} * 2^{-n})$$

Unde 'n' este factorul pondere exponențial care poate fi configurat de utilizator.

Pentru valori mari ale lui n lungimea medie anterioară devine importantă. Un factor mare netezește valorile de maxim și minim ale lungimii cozii. Procesul WRED va începe cu greu aruncarea pachetelor și va continua să arunce pachete și pentru un interval de timp după ce lungimea cozii scade sub pragul minim.

Pentru valori mici ale lui 'n' lungimea medie a cozii va fi apropiată de lungimea curentă a cozii, în acest caz procesul WRED răspunde imediat la cozi lungi iar odată ce lungimea cozii scade sub pragul de minim aruncarea pachetelor se va opri. Dacă valoarea lui 'n' este prea mică, WRED va suprareacționa la exploziile temporare de trafic și va arunca trafic care nu ar trebui aruncat.

RRED este o variantă RED destinată scopului de a îmbunătăți capacitatea TCP împotriva atacurilor LDoS. Principiul pe care se bazează este acela de a detecta și de a filtra pachetele atac înainte de a aplica algoritmul RED normal.

Implementarea RRED constă în inserarea înaintea blocului RED pe un router a unui bloc de detecție și filtrare. Diferențierea între pachetele atac și pachetele tcp normale se face prin faptul că dacă un pachet este trimis la scurt timp după ce un alt pachet a

fost aruncat atunci acel pachet este suspectat a fi pachet de atac.

Criterii de evaluare RED

Pentru evaluarea RED se folosesc următoarele criterii:

Evitarea congestiei: Dacă pachetele sunt aruncate cum ajung atunci când lungimea maximă a cozii atinge pragul maxim RED garantează că lungimea medie calculată a cozii nu depășește pragul maxim. Dacă parametrul w_q este setat în mod corespunzător în acest caz RED controlează chiar lungimea medie a cozii actuale.

Nici o sincronizare globală: Rata la care RED marchează pachetele depinde de nivelul de congestie. În timpul unei congestii scăzute probabilitate de a marca fiecare pachet este mică iar ea crește odată cu creșterea congestiei. RED evită sincronizarea globală prin marcarea pachetelor cu o rată cât mai mică posibil.

Simplitate: Algoritmul RED ar putea fi implementat cu un overhead moderat în rețele curente.

Maximizarea puterii globale: RED controlează în mod explicit lungimea medie a cozii și se poate observa că puterea globală este mare decât în cazul Drop Tail.

Corectitudinea: Un scop al mecanismului de evitare a congestiei este chiar corectitudinea dar ea nu este bine definită. Routerelor care folosesc RED nu fac diferență între conexiuni particulare sau clase de conexiuni. Pentru RED fracția de pachete marcate pentru fiecare conexiune este proporțională cu cât de mult acea conexiune ocupă banda.

Corespunzător pentru o scară largă de medii : Mecanismul random de marcarea a pachetelor este corespunzător pentru rețele cu diferite domenii de valori ale timpului dus întors și al capacității de transfer precum și pentru diferite domenii de valori ale numărului de conexiuni la un timp dat.

Sensibilitatea parametrilor pentru implementarea eficientă a RED

Mecanismul de evitare a congestiei ar trebui să aibă sensibilitate mică la parametrii iar ei ar trebui să fie aplicabili rețelelor cu diferite valori ale lățimii de bandă.

Parametrii w_q , \min_{th} , \max_{th} sunt necesari pentru luarea de decizii conștiente asupra lungimii medii dorite a cozii, asupra dimensiunii și duratei a exploziilor de trafic permise. Parametrul \max_p poate fi

ales dintr-un domeniu foarte larg el fiind doar limita superioară a probabilității de marcarea a pachetului actual p_b . Dacă congestia este suficient de mare încât nu se mai poate controla lungimea medie a cozii prin marcarea fracției \max_p din pachete atunci lungimea medie a cozii va depăși pragul de maxim și vor începe să fie marcate toate pachetele pînă cînd congestia este sub control.

Pentru o performanță bună este bine să se asigure calcule adecvate a lungimii medii a cozii prin setarea $w_q \geq 0.001$. Lungimea medie a cozii este limitată de \max_{th} atît timp cît lungimea medie calculată a cozii avg este o reflexie de acuratețe a lungimii medii a cozii actuale. W_q nu ar trebui setat prea jos pentru a se evita ca lungimea medie calculată a cozii să întîrzie prea mult reflectarea unei creșteri a lungimii actuale a cozii. \max_{th} și \min_{th} trebuie să fie suficient de mari astfel încât să se maximizeze puterea rețelei iar diferența dintre ele suficient de mare cît să se evite sincronizarea globală. În general e bine ca \max_{th} să fie de 2 ori mai mare decît \min_{th} .

Bibliografie

- Adaptive RED : An algorithm for increasing the robustness of RED's active queue management by Sally Floyd, Ramakrishna Gummadi and Scott Shenker
- RRED: Robust RED algorithm to counter low-rate denial-of-service attacks by Changwang Zhang, Jianping Yin, Zhiping Cai, Weifeng Chen
- Random early detection gateways for congestion avoidance by Sally Floyd and Van Jacobson
- http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfconavps1835_TSD_Products_Configuration_Guide_Chapter.html#wp1003370
- http://en.wikipedia.org/wiki/Robust_random_early_detection