March 2, 2016

1 Single SW

We first extend CO2 to the case with single switch and multiple receivers. The topology of the switch and receivers is shown in Figure . The switch holds rules with multiple action tag, and each action represents forwording the traffic to the receiver with the same action tag. The receivers feedback the top overselected IPs with regard to bytes volume to controller, and the controller decides which one to put into the Blacklist in the switch. The challenge is that the Blacklist memory is a bottleneck. We propose a Q-Learning (RL) based Approach (QLA) in the controller to allocate Blacklist space to the rules with different actions.

background foreground background,main,foreground

sensor=[draw, fill=blue!20, text width=4em, text centered, minimum height=2.5em,drop shadow] ann = [above, text width=5em, text centered] wa = [sensor, text width=4em, fill=red!20, minimum height=4em, rounded corners, drop shadow] aggr = [sensor, text width=6em, fill=red!20, minimum height=4em, rounded corners, drop shadow] sc = [sensor, text width=13em, fill=red!20, minimum height=10em, rounded corners, drop shadow]

1! (wa) [wa] switch;

(wa.east)+(3,1) node (vote) [sensor] Receiver1; (wa.east)+(3,0) node (vote1) [sensor] Receiver2; (wa.east)+(3,-1) node (vote2) [sensor] Receiver3; (wa.east)+(3,-2) node (vote3) [sensor] ...;

[draw, -¿] (wa.east) – node [above] (vote.west); [draw, -¿] (wa.east) – node [above] (vote1.west); [draw, -¿] (wa.east) – node [above] (vote2.west); [draw, -¿] (wa.east) – node [above] (vote3.west);

background (vote.north)+(1,0.3) node (ab) ; (vote.south)+(-1,-3.5) node (cb) ;

[fill=yellow!20,rounded corners, draw=black!50, dashed] (ab) rectangle (cb);

(wa.north)+(-3.7,0.2) node (ms) ;

Assume the system contains 1 switch and n receivers, and we define a utility function U_i for receiver r, we aim to maximize the following objective:

$max\ \sum_{r=1}^{n} U_r(a_{r,t})\ st.,$

$\sum_{r=1}^{n} a_{r,t} \leq A_c$

Where $a_{r,t}$ is the Blacklist space allocated to the rules from receiver r at time t, and A_c is the Blacklist space capacity.

We use reinforcement learning to compute the utility function for each receiver, so,

$U_r(a_{r,t}) = Q_r(s_{r,t}, a_{r,t})$

where function Q_r is the value function in the reinforcement learning system, and $s_{r,t}$ is the state at time t.

Reinforcement learning is learning what to do-how to map situations to ations-so as to maximize a numerical reward signal[Reinforcement book]. Beyond the agent and the environment, one can identify four main subelements of a reinforcement learning system: a policy, a reward function, a value function, and optionally, a model of the environment[Reinforcement book].

A policy defines the learning agent's way of behaving at a given time. A state $s_{r,t}$ is the Blacklist space already allocated to the rules from receiver r, and a policy $a_{r,t}$ is to increase or decrease the allocation given on the current state.

A reward function r_t defines the goal in a reinforcement learning problem. We set a target overselection rate ov_T, and if the actual overselection rate ov_a is larger than ov_T, the reward function is a negative const value, otherwise, it is a positive const value. By performing the reward function, we aim to control the actual overselection rate under the target value.

We use Q-learning in our design, and the one-step Q-learning is defined by,
$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\{r_{t+1}) + \gamma\ max_a\ Q(s_{t+1}, a) - Q(s_t, a_t)\}$

The $Q_r(s_{r,t}, a_{r,t})$ value is kept at a table called Q table. There are n tables for the n reinforcement learning systems, respetively.

The QLA is shown in procedural form in Algorithm.

Initialize Q tables, policy and states.
Repeat (for each step of episode):
foreach $s, s < m$
foreach $r, r < n$
Q-learning to update $Q_r(s_{r,t}, a_{r,t})$
forend
forend
Repeat end
Compute the maximum allocation from the n tables by,
$max\ \sum_{r=1}^{n} U_r(a_{r,t})\ st.,$
$\sum_{r=1}^{n} a_{r,t} \leq A_c$

2 Multiple SW

We extend our design to a network-wide case, with multiple switches and multiple receivers, and the topology of the switch and receivers is shown in Figure .. Assume the system contains m switches, we define the objective function by,
$max\ \sum_{s=1}^{m} \sum_{r=1}^{n} U_{s,r}(a_{s,r,t})\ st.,$
$\sum_{r=1}^{n} a_{s,r,t} \leq A_{c,s}$
And,
$U_{s,r}(a_{s,r,t}) = Q_r(s_{s,r,t}, a_{s,r,t})$
Where the subscript s indicates switch s

3 experiment settings

4 simulation analysis