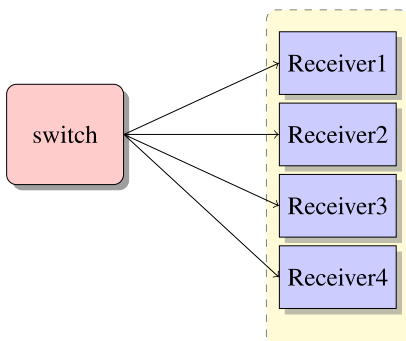
Collaborative Packet Classification with Overselection

Hao Wu[†*] Yunhong Xu[§*] Nick Duffield[§] Minlan Yu[‡] Bin Liu[†]
[†] Tsinghua University [§] Texas A&M University [‡] University of Southern California
wu-h11@mails.tsinghua.edu.cn, {yunhong,duffieldng}@tamu.edu, minlanyu@usc.edu, liub@mail.tsinghua.edu.cn

1. NETWORK-WIDE ALLOCATION

Single Switch.

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



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 $s_{r,t}$ in our design is the Blacklist space already allocated to the rules from receiver r, and a policy $a_{r,t}$ is to increase, decrease or maintain the allocation given on the current state. For example, policy 1 is to increase the allocation by 200 slots, and policy 2 is 100 slots decreament.

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 scalar, otherwise, it is a positive scalar. By performing the reward function, we aim to control the actual overselection rate under or close to ov_T.

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,

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

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

Multiple Switches.

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,

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

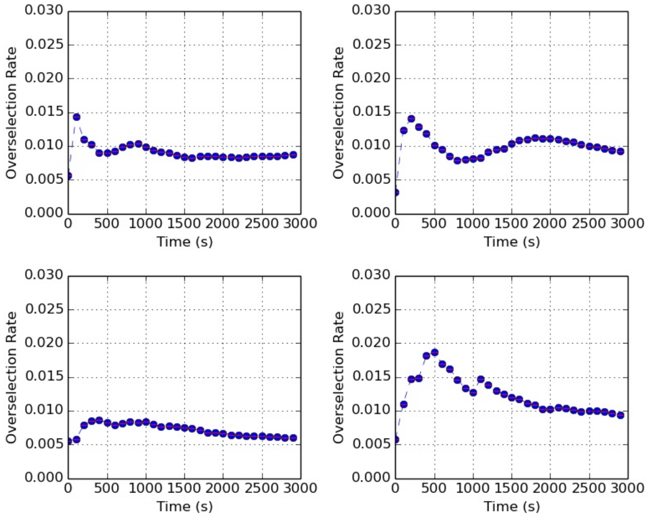
$\sum_{r=1}^{n} a_{s,r,t} \leq A_{c,s}$

And,

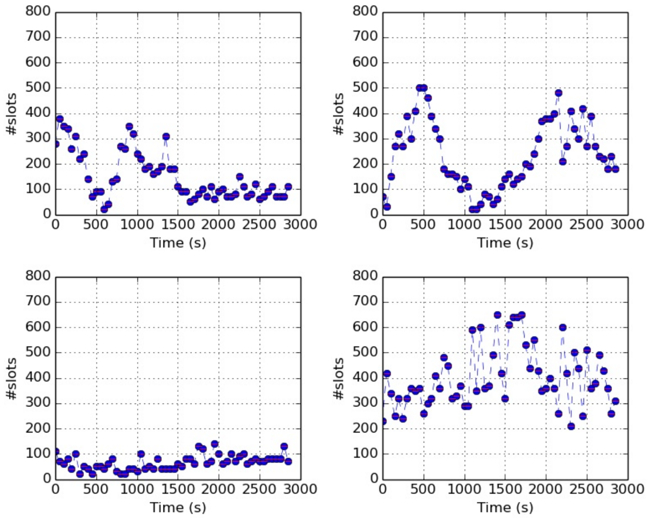
$U_{s,r}(a_{s,r,t}) = Q_r(s_{s,r,t}, a_{s,r,t})$

Where the subscript s indicates switch s

[*]Co-primary authors placed in alphabetic order.

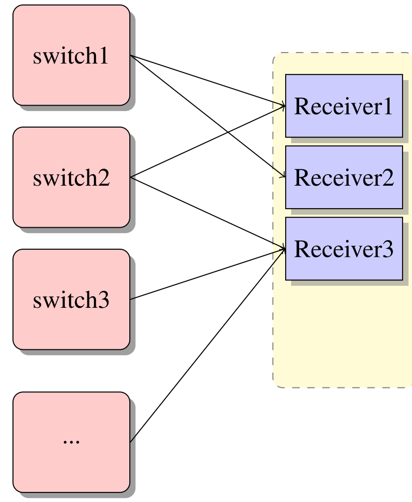


(a) The overselection rate



(b) The convergence time

Figure 1: Results for different updating intervals for both the BUA and the DCA



Experiment Settings.

Single Switch.

The receiver number $n = 4$ as shown in Figure,

Multiple Switches.

The switch number $m = 3$, and the receiver number $n = 3$

Simulation Results and Analysis.

2. REFERENCES