An OpenFlow Safe Update Protocol

Rick McGeer,
HP Labs
August 13, 2012
The Safe Update Problem

–Correctness constraints:
  • Deterministic handling of each packet (each packet takes a well-defined route through the network)
  • Deterministic handling of each flow
    – Strong Consistency: every packet from a flow takes the same route through the network
    – Weak Consistency: Prefix of a flow takes route 1, suffix takes route 2 (Preferred in some applications, notably traffic steering)

–Reality constraint
  • Rule updates arrive at each switch asynchronously and unpredictably
Update Example

A

B

C

u

v

s

t

r

w

y

z

A

B

C

u

v

s

A

B

C

r

w

y

z
Approach I: Verification

- Formalization of problem: switch $j$ implements logic function $F_{j1}$ (before update) or $F_{j2}$ (after update)
  - Introduce new variable $x_j$
  - Switch function is $x_jF_{j2} + x_j'F_{j1}$
  - Composed in network exactly as in [McGeer2012a]
  - Verification property holds iff verification obligation is met for all values of the $x_j$s

- Probably boosts the problem by a complexity class

- Properties to be verified will not, in general, hold anyway for most updates...
  - The point of an update is to change packet handling....
Example Bad Update

Update Arrives at $u$ but not $w$
Approach II: Do Both At The Same Time

– Reitblatt, 2011
  • Updated version at SIGCOMM 2012
– Load each switch with the function $x_j F_{j2} + x_k F_{j1}$
– Use an unused bit in the header to mark each packet with which function should handle it
  • Plenty of unused header bits
– Consistency achieved by appropriate packet marking
  • Mark packets uniformly in flow for strong consistency
  • Mark prefix and suffix differently for weak consistency
Approach II

A

u
v
w
y
z

B

C
Approach II Features

😊 Deterministic packet-handling (?)

😊 Packet handling selected at edge

⚠️ Takes a lot of TCAM space
  - Each switch function roughly doubled in size
  - TCAMs are typically most expensive element of switch

⚠️ Need not simply a header bit for $x_j$, but one that can be matched in the fast path
  - May mean burning half the space in a common header field (uses VLAN tags)
Approach III: Use the Controller

– Assumption I: *Most* updates affect the handling of only a few packets
  • E.G. only *one* switch in our example changed packet handling
  • E.g. In Traffic Steering Application, packets from only *one* flow are redirected

– Assumption II: Packets can be sent safely to the controller at any time and held, released when safe
  • Optimization: for each switch, designate a packet refuge (need not be the controller)
Protocol

– Send function $F_{12}$ to each switch
  • $F_{12} = F_1$ if packet handling doesn’t change at that switch
  • $F_{12} = F_2$ if only new flows are added
  • $F_{12} = $ Send to controller otherwise. Controller holds packets

– When switch gets $F_{12}$, send completion signal to controller

– When all completion signals received
  • Send $F_2$ to each switch
  • Release held packets from controller to next switch on path 2
Example of the OpenFlow Safe Update Protocol

$F_1$ function

$F_{12}$ function
Example (cont)

$F_2$ function

Cleanup dead links
Approach III Features

😊 TCAM Space Conservation on switch
  • $\text{Max}(F_1, F_2)$

😊 No impact on flowspace, even for fast match

😊 Provably correct, no race conditions

⚠️ Increase in affected flow latency during transition
  • Minimum two LAN round-trips + rule load time

⚠️ Can only ensure weak consistency

⚠️ Increase in traffic to controller
  
  $(1.5 \times \text{LAN Round trip} + \text{rule load time}) \times \text{sum of bandwidth of affected flows}$
Conclusions and Further Work

– Approach II and Approach III represent different points in a trade space
  • Key positive: *both work*
  • Different costs and benefits
    – How much is switch flowspace and rulespace worth compared to latency and controller bandwidth penalty, and loss of strong flow consistency?
    – Likely dependent on specifics of LAN, switches, controller, application

– Not the only points in the trade space
  • Can use hybrid method

– One possible hybrid…
  • Use Reitblatt marking but send only hybrid function to changed switches
  • Topology and updates of this approach, marking of Reitblatt
  • Conserves TCAM space on most switches, synthesis techniques to conserve flowspace…
Thanks!