



UNC CHARLOTTE

OF-RHM: Transparent Moving Target Defense using Software Defined Networking

Haadi Jafarian, Qi Duan and Ehab Al-Shaer

ACM SIGCOMM HotSDN Workshop

August 2012

Helsinki, Finland



- Static assignment of IP addresses gives adversaries significant advantage
 - Host scanning and Network reconnaissance
 - Intelligent worm propagation
 - Attack planning
- The goal of IP Mutation moving target defense is Distort, Deceive or Deter attack reconnaissance and planning.



Requirements/Challenges for IP Mutation

- Highly unpredictable
- Fast
- Operationally Safe
- Transparent
 - No interruption for active session
 - Deployable with no major network changes



- Incorporation IP Mutation on traditional networks is disruptive and costly
 - Application/host Transparent → Network level
 - Global optimization and control
 - Real-time distributed reconfiguration
 - Management synchronization
- Software-defined networking (SDN) provides flexible infrastructure for developing and managing random IP mutation



Approach Overview

- *The goal of OpenFlow Random Host Mutation (OF-RHM) is to mutate IP addresses of end-hosts randomly, frequently and quickly.*
- Each MT host is assigned a new virtual IP (vIP) at regular intervals (called *Mutation interval – T*).
- vIPs are selected from unused address space of the network
- Real IP address (rIP) of the hosts remains unchanged
- vIPs are translated to rIPs right before the host.
- vIP are the only routable addresses.



Unused Address Range Construction

- We have a set of n hosts h_1, \dots, h_n
- Each host has a required mutation rate R_i
 - Sensitive hosts must have higher mutation rates
- Each host belongs to one subnet in the set $\{s_1, \dots, s_z\}$
 - a decision variable $c_{i,k}$ shows if host h_i belongs to subnet s_k
- Unused address ranges of network by Boolean operations
 - $\{r_1, \dots, r_m\} = A \wedge \neg(A_1 \vee \dots \vee A_u)$
 - A represents full network address space
 - A_1, \dots, A_u are used address ranges
- Large ranges are broken into smaller ones



Problem Definition

- **Main Objective:** maximize both mutation unpredictability and mutation rate.
- **Range Allocation Problem:** Given the IP addresses of MT hosts (h_i) located in subnets (s_k), and the required mutation rate for each host (R_i), how to allocate/assign ranges of unused IP addresses to hosts/subnets such that

- Allocate the largest possible unused address space as contiguous ranges
- Assigned ranges have enough IP addresses to satisfy the required *mutation rate* of all hosts in that subnet during a mutation interval T
- A subnet can be assigned multiple mutation ranges
- Ranges are assigned based on their sizes and proportional to the mutation requirement of each subnet.
- One range can only route to one subnet s_k

Unpredictability
Constraints

Mutation Rate
Constraint

Routing
Constraint



Range Allocation Complexity & Formulation

- Assigning address ranges to subnets (Range Allocation Problem) is an NP-hard Problem
 - Generalization of knapsack problem
 - A subnet may be assigned *with multiple ranges*
 - different mutation requirements
 - unequal range sizes.
- We formulate this problem as a constraint satisfaction problem using SMT (Satisfiability Modulo Theories)
 - We use SMT to find values for decision variables $b_{j,k}$



Range Allocation Constraints

- **Mutation Rate Constraint**

- The total number of mutated vIPs of all hosts in subnet s_k during T must be less than the aggregate size of all ranges assigned

- $\forall k, \left(\sum_{1 \leq i \leq n} c_{i,k} R_i \right) * T \leq \sum_{1 \leq j \leq m} b_{j,k} |r_j|$

- **Range Allocation Constraint**

- Each range must be assigned to exactly one subnet

- $\forall j, \sum_{1 \leq k \leq z} b_{j,k} = 1$



- **Range Distribution (Unpredictability) Constraint**

- ranges must be assigned to subnets proportionate to their total required mutation rate
- We define P_k as total required mutation of subnet s_k during T on total size of ranges allocated to it

- $$\forall k, P_k = \frac{T * \sum_{1 \leq i \leq n} c_{i,k} R_i}{\sum_{1 \leq j \leq m} b_{j,k} |r_j|}$$

- We define P_a as total required mutation of all hosts on total size of unused address space

- $$P_a = \frac{\sum_{1 \leq i \leq n} R_i}{\sum_{1 \leq j \leq m} |r_j|}$$

This constraint can be denoted as: $\forall k, P_k \approx P_a$



IP Mutation Problem

- **IP Mutation within allocated ranges in each subnet:**
 - Each host must be associated with a new vIP after each mutation interval according to R_i
 - Any vIP will NOT be assigned more than once for number of consecutive T mutation intervals
 - vIPs must be chosen randomly from ranges assigned to subnet with No collision with hosts in the same subnet
- The new vIP is chosen randomly in two ways:
 - Blind Random (uniform) Mutation
 - Weighted Random Mutation (based on feedback)

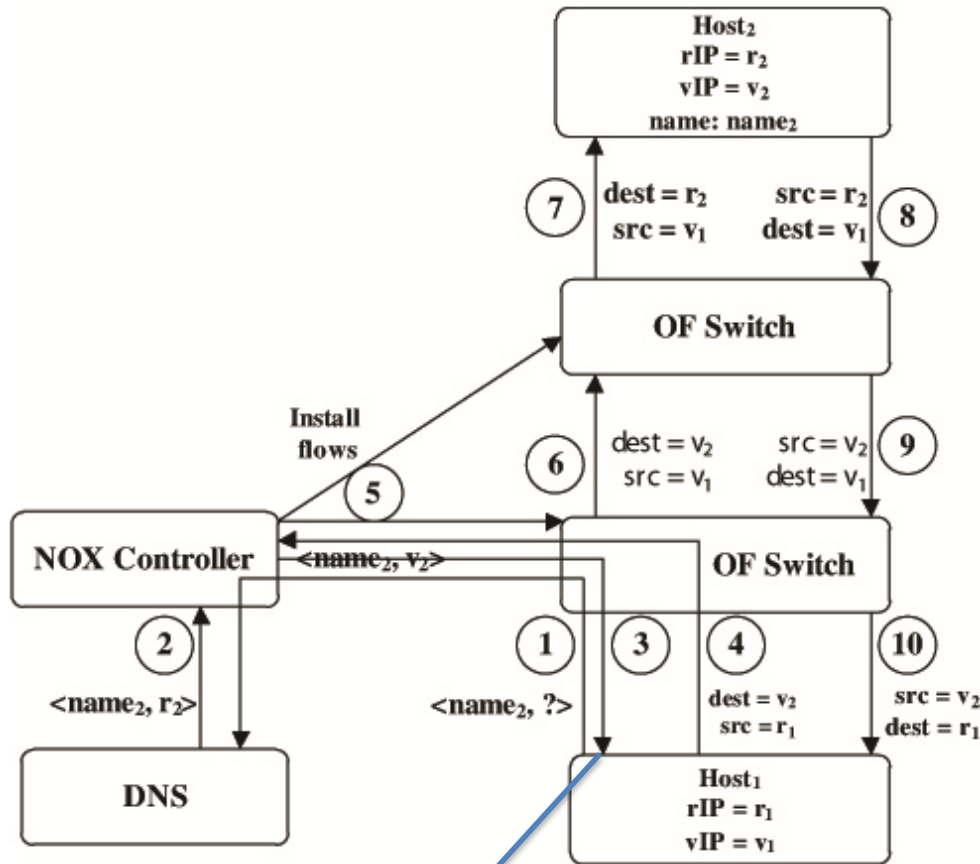


UNC CHARLOTTE

Protocol, Architecture, Algorithms



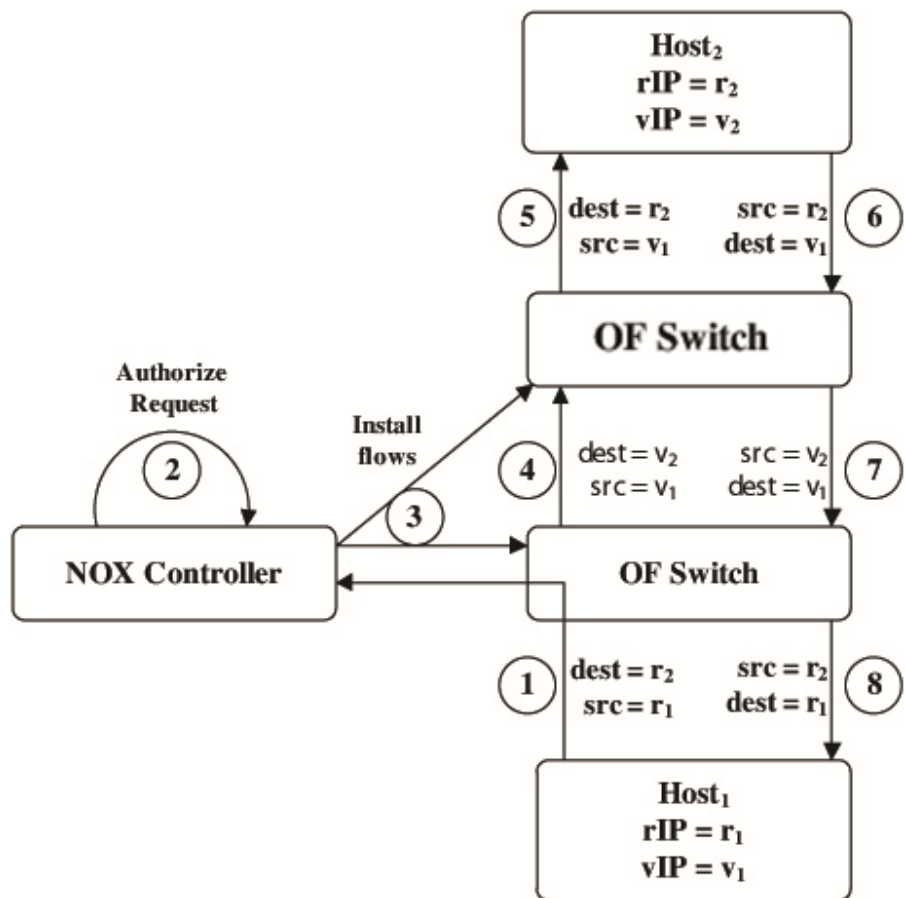
Communication via Host Name



TTL set according to mutation rate



Communication via rIP

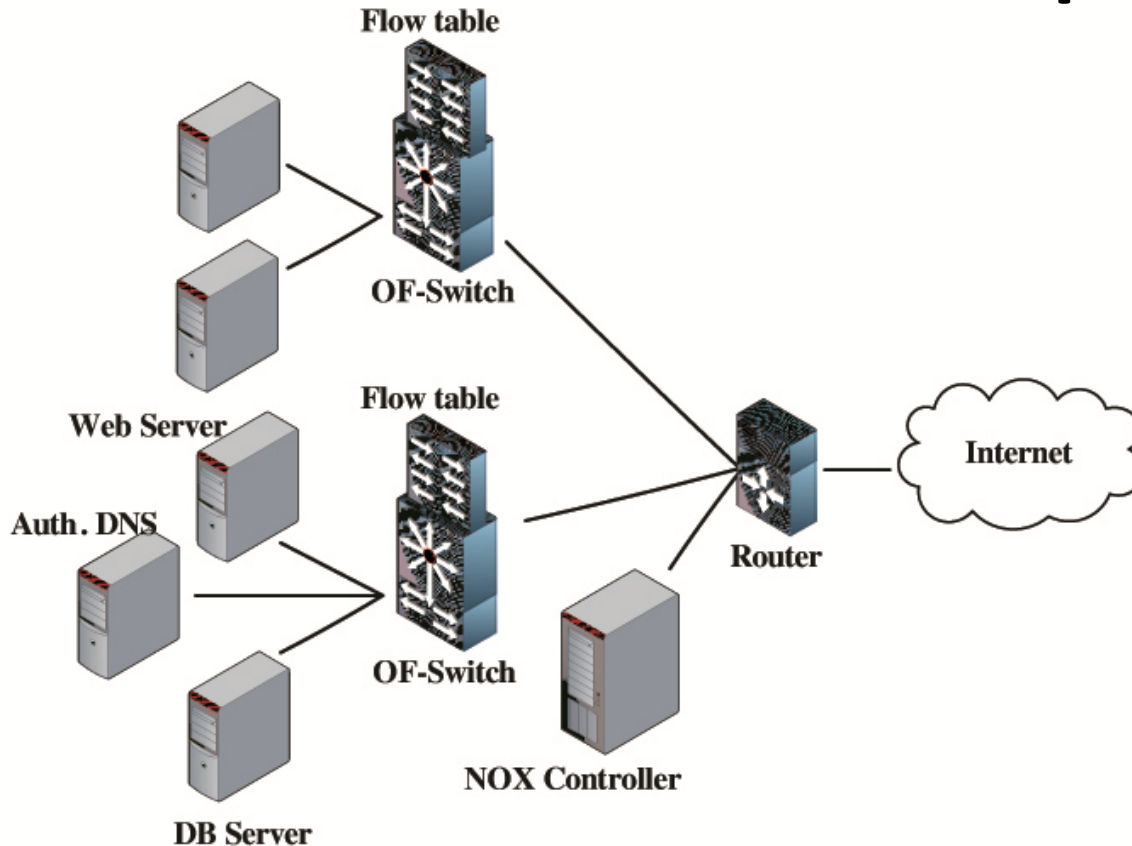




Architecture & Implementation

- We implemented OF-RHM on a *mininet* network controlled by a NOX controller
 - a network including **1024 hosts** with OpenFlow switches
 - Open vSwitch kernel switches
- NOX Controller Tasks (acts as the central authority)
 - Managing IP mutation: run SMT solver globally, and avoid collision locally
 - Installing flow entries in switches
 - Updates DNS responses
- The architecture can be extended to include several controllers
 - Each controller can be autonomous and it can manage its designated subnets independently

Architecture & Implementation





- OF-switches are configured to send unmatched packets to the controller
- If packet is destined to rIP it is authorized
 - If authorization succeeds, necessary flows are installed in path switches
- If packet is destined to vIP
 - Necessary flows are installed in path switches with corresponding actions
 - rIPs are translated to vIPs for outgoing packets
 - vIPs are translated to rIPs for incoming packets



UNC CHARLOTTE

Effectiveness



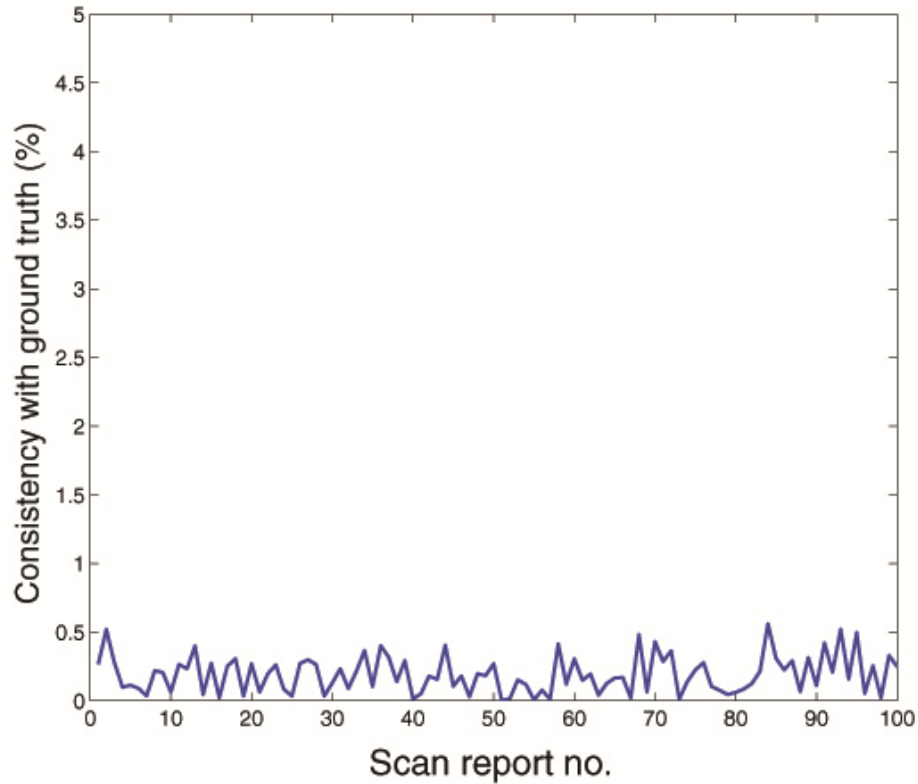
Random External Scanners (1)

- Scanning is usually the precursory step for attacks
- attackers usually use scanning tools such as Nmap to discover active hosts
- We run 100 *Nmap* scan on our Mininet class B network which consists of 1024 hosts
- We compared the result with ground truth
- Less than 1% are discovered in any scan



UNC CHARLOTTE

Random External Scanners (1)

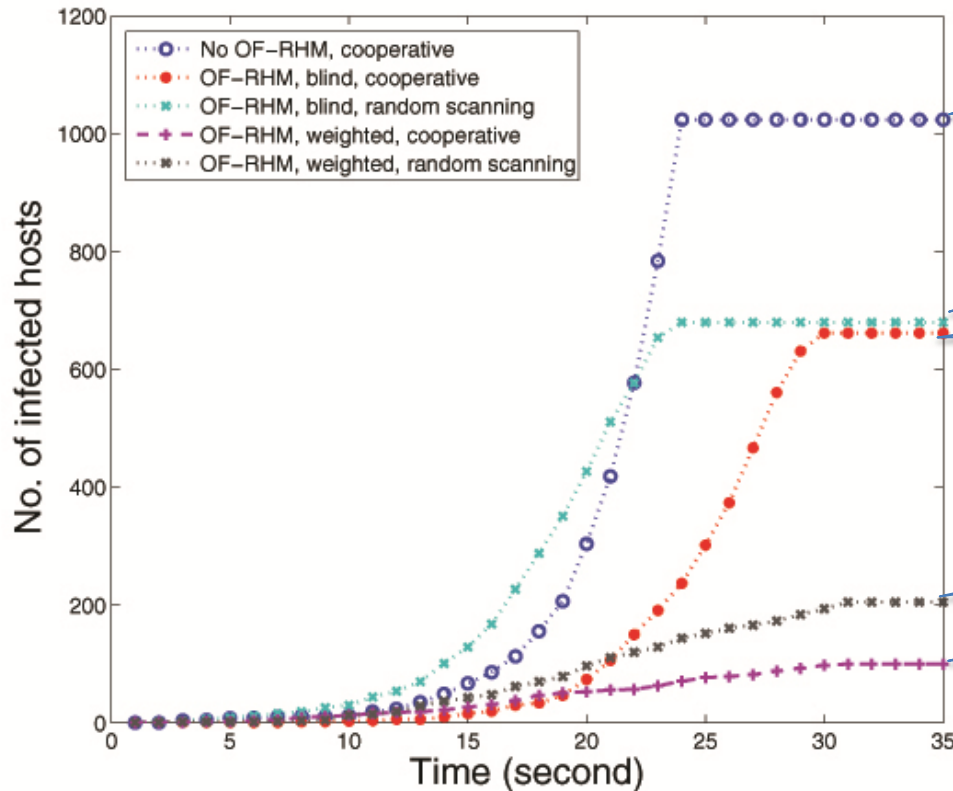




- We examined propagation of
 - random scanning worms
 - cooperative worms
- We studied their propagation for both
 - Blind Mutation
 - Weighted mutation
 - Higher weight is assigned to highly scanned Ips



Worms (2)



NP OF-RHM = 100%

Random + blind = 65%

Cooperative + blind = 65%

Random + weighted = 18%

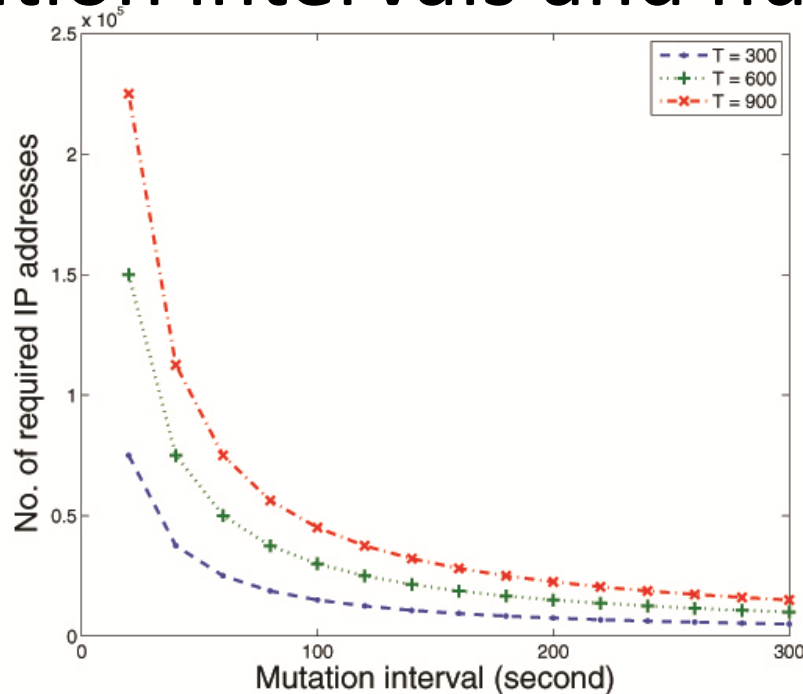
Cooperative + weighted = 10%



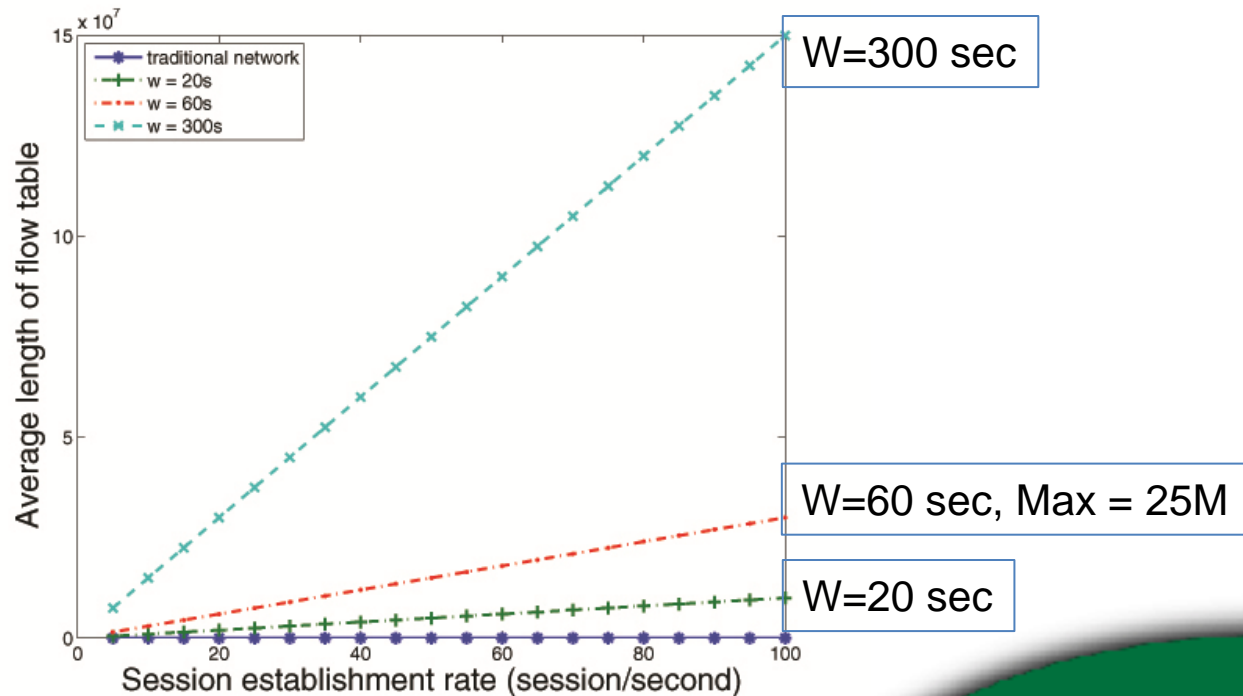
UNC CHARLOTTE

Overhead

- Required IP address size for various mutation intervals and number of hosts



- Flow table length for different session establishment rates and session durations



- The longer the session the less effective



Conclusion and Future Work

- Random IP Mutation is shown to be effective to counter many reconnaissance attacks
 - We are working on configurable evaluation tool for RHM
- Based on our implementation of RHM on both traditional and OpenFlow networks, SDN shows a great flexibility and efficiency in developing/deploying novel cyber defense techniques
 - Much easier, efficient and deployable (cost-effective)
- Future Work
 - Exploring other reconnaissance and Cyber attack models
 - Exploring mutation techniques other than time-based on SDN
 - Exploring distributed controller approach



UNC CHARLOTTE

Questions?



Controller Algorithm

Algorithm 1 NOX controller algorithm

```
determine unused ranges.
determine range-to-subnet assignments
for all packets  $p$  from OF-Switches do
  if  $p$  is a Type-A DNS response for host  $h_i$  then
    set DNS  $addr$  to current  $vIP(h_i)$ ,  $TTL \simeq 0$ 
  else if  $p$  is a TCP-SYN or UDP from  $h_i$  to  $h_j$  then
    if  $p.src$  is internal then
      install  $in$  flow in src OF-switch with
        action  $srcIP(p) := vIP(h_i)$ 
      install  $out$  flow in src OF-switch with
        action  $dstIP(p) := rIP(h_i)$ 
    end if
    if  $p.dst$  is rIP then
      if  $h_i$  access to  $h_j$  is authorized then
        install  $in$  and  $out$  flows in dest OF-switch
      end if
    else [ $p.dst$  is vIP]
      install  $in$  flow in dest OF-switch with
        action  $dstIP(p) := rIP(h_j)$ 
      install  $out$  flow in dest OF-switch with
        action  $srcIP(p) := vIP(h_j)$ 
    end if
  end if
end if
for all mutation of each host  $h_i$  do
  set  $vIP(h_i)$  to a new vIP
end for
end for
```
