

Network virtualization: role of OpenFlow & acid test for network virtualisation

Didier Colle
didier.colle@intec.ugent.be
www.ibcn.intec.ugent.be
INTEC Broadband Communication Networks (IBCN)
Department of Information Technology (INTEC)
Ghent University – IBBT

EC SPARC project: www.fp7-sparc.eu

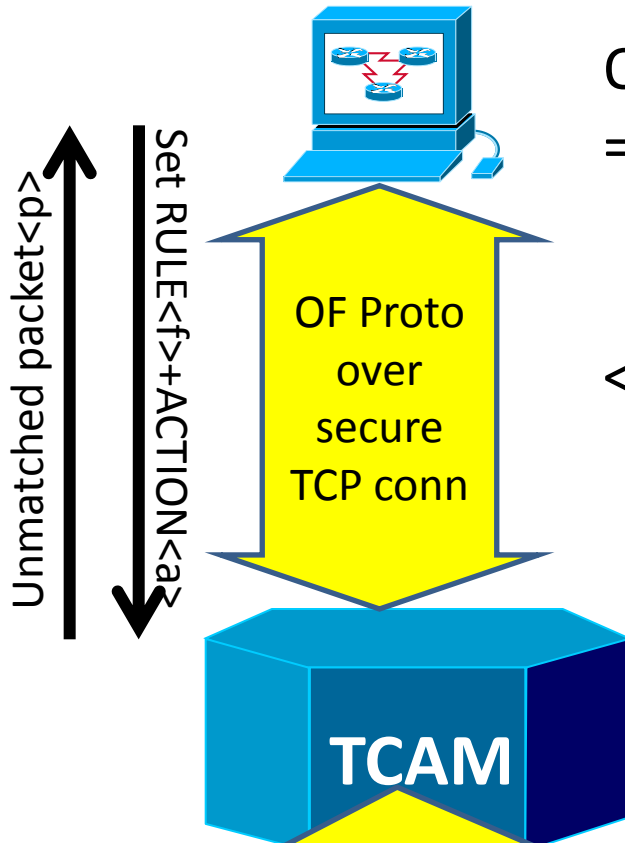
EC OFELIA project: www.fp7-ofelia.eu

OpenFlow

OpenFlow
= protocol

to make TCAMs available to ctrl
to hand-off exceptional packets to ctrl

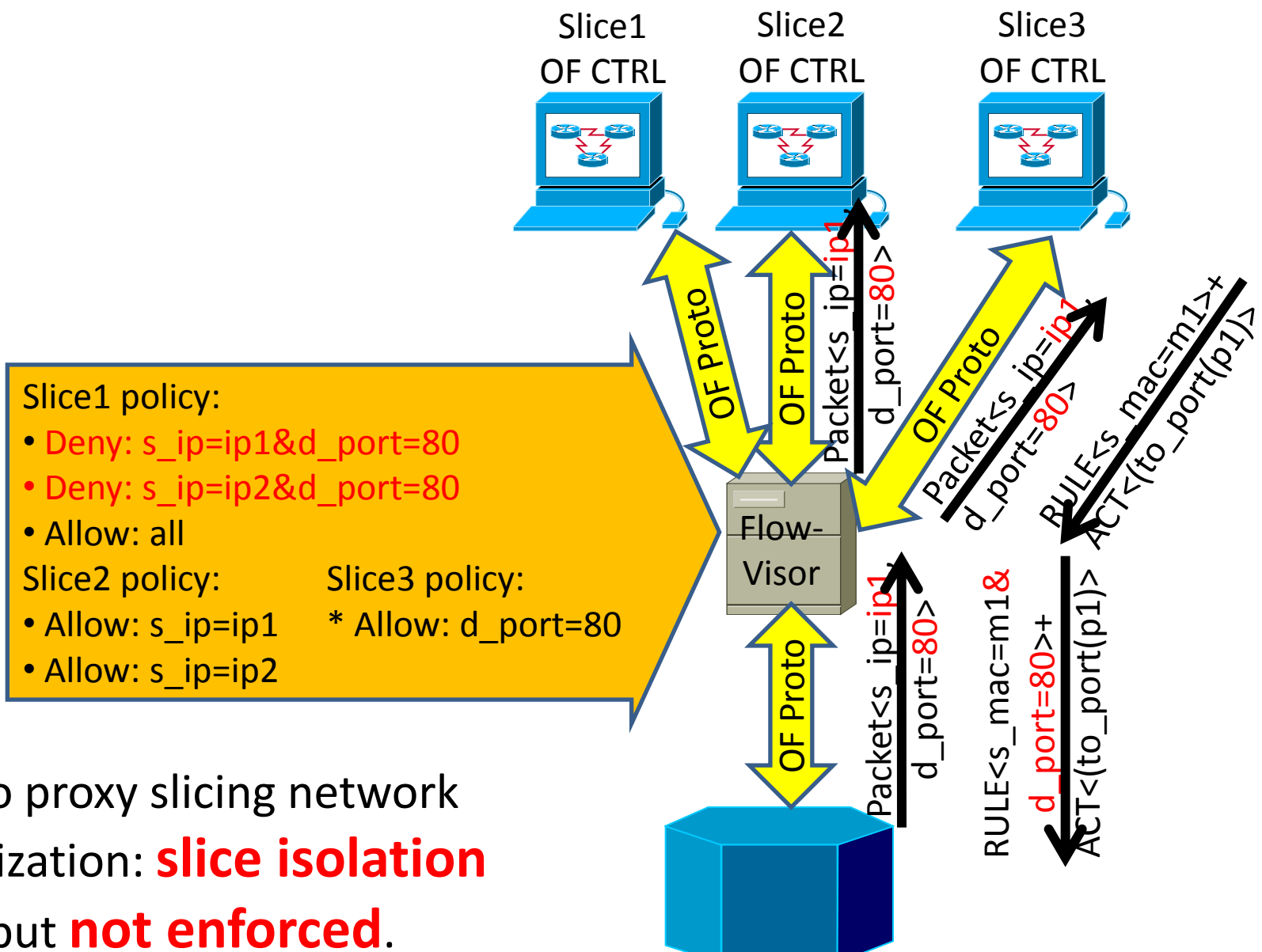
<> Virtualization



- **RULE** = filter on sw_port, s_mac, d_mac, eth_type, vid, s_ip, d_ip, ip_prot, s_port, d_port
- **ACTION** = to port(x) (&s_mac=y&d_mac=z) || drop || to Ofctrl || to legacy pipeline || ...
- **STATE** = packet&byte counters

VN ACID TEST 1: CTRL ISOLATION.

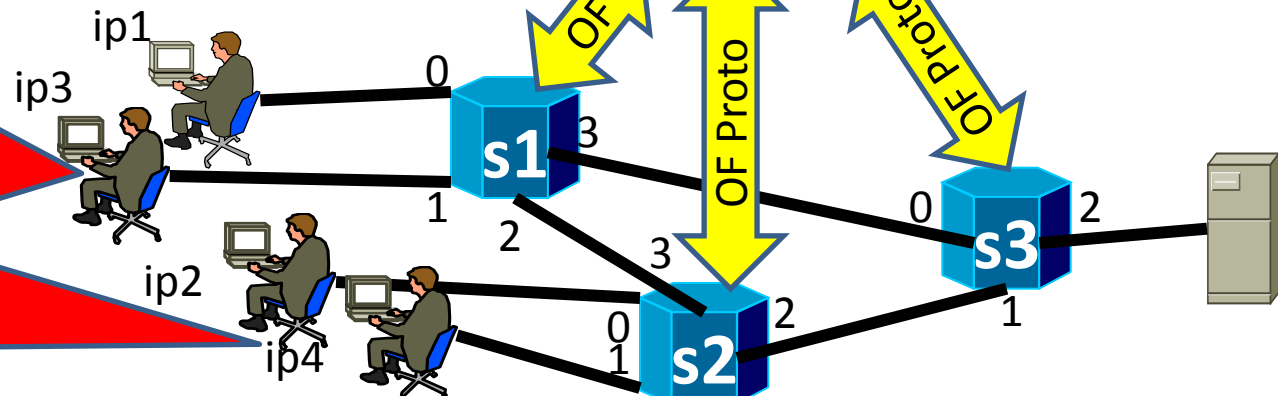
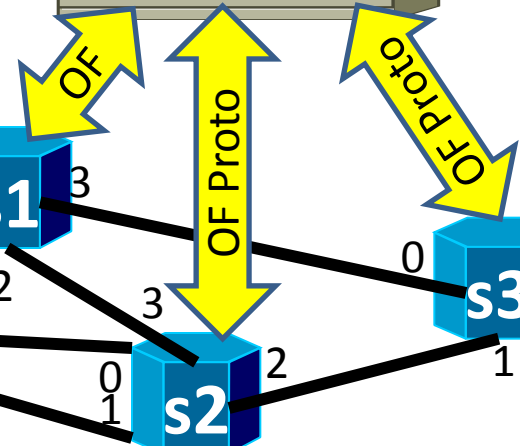
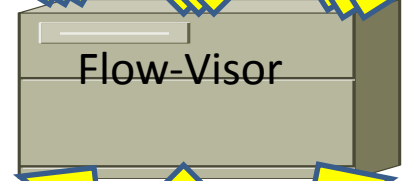
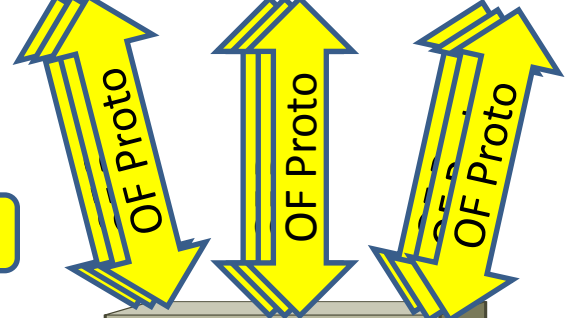
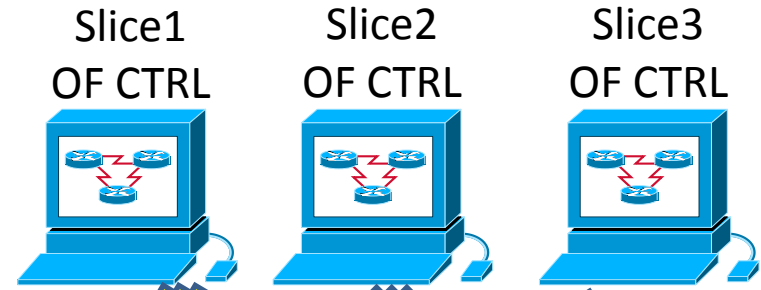
FlowVisor



FlowVisor
= OF proto proxy slicing network
<> Virtualization: **slice isolation**
possible, but **not enforced**.

VN ACID TEST 2: access control & VN labeling

FlowVisor



VN label/id

VN access control

- Slice1 policy:
- Allow: s_ip=ip1&d_port=80
 - Allow: s_ip=ip2&d_port=80
 - AllowedPorts: 0, 2, 3 s1
 - AllowedPorts: 0, 2, 3 s2
 - AllowedPorts:: all s3

Guarantee they are not high jacking IP1 or IP2 addr ==> drop rules on these ports for s_ip=ip1 || ip2 (FlowVisor support???)

VN ACID TEST 3: virtualization of address/port ranges

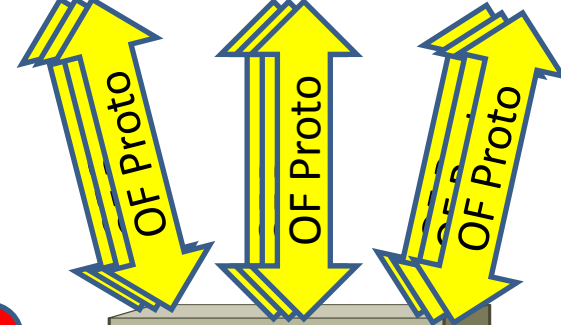
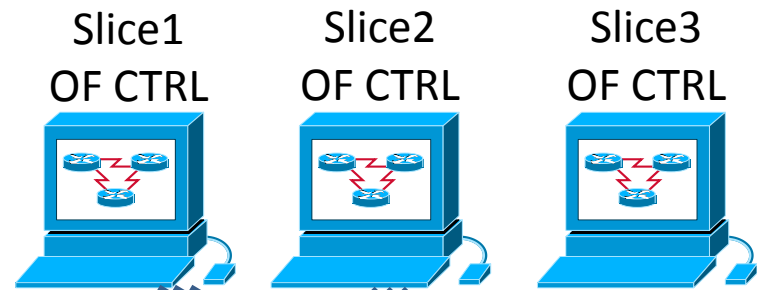
FlowVisor

Slice1 policy:

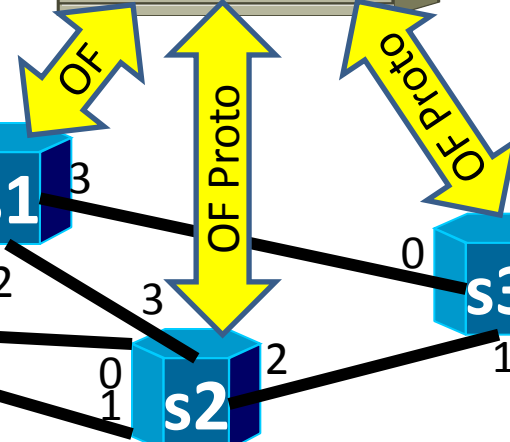
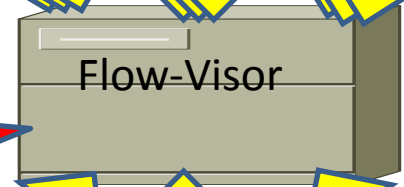
- Allow: s_ip=**ip1**
- Allow: s_ip=**ip2**
- AllowedPorts: 0, 2, 3 s1
- AllowedPorts: 0, 2, 3 s2
- AllowedPorts:: all s3

Slice2 policy:

- * Allow: s_ip=**ip1**
- * Allow: s_ip=**ip2**
- * AllowedPorts: 1, 2, 3 s1
- * AllowedPorts: 1, 2, 3 s2
- * AllowedPorts:: all s3

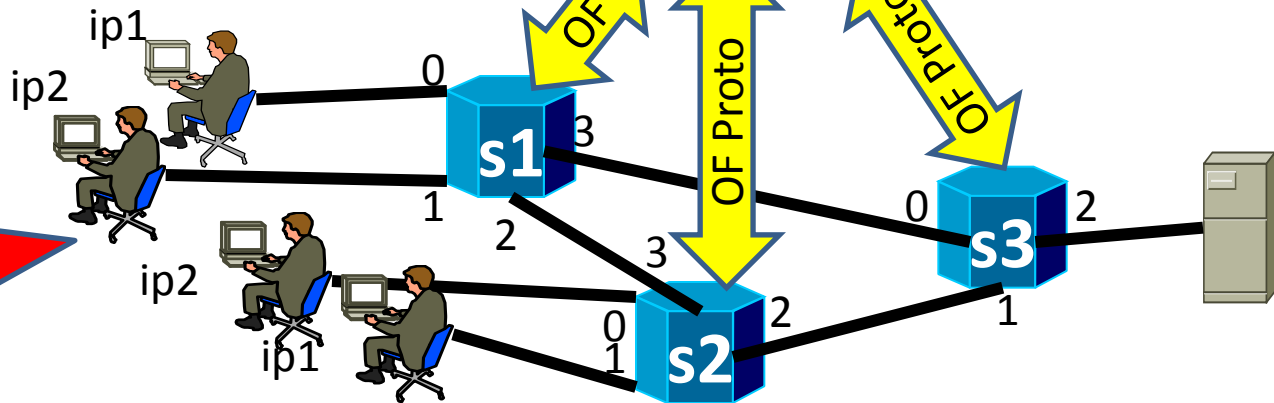


==> 2) similar translation in openflow messages (support in FlowVisor ???)



Overlapping address ranges should be possible

==> 1) translate virt. addr. into (internal) phys. addr. at ingress and vice versa at egress



FlowVisor

- FlowVisor also has built in measures for regulating usage of resources outside the wirespeed forwarding path:
 - Rate limiting unmatched packets
 - Rate limiting OF requests from OF CTRLs
 - Slow path (/ legacy) forwarding
 - ...

Conclusions: OpenFlow & FlowVisor

- OpenFlow & FlowVisor:
 - IS NOT per def virtualization, but
 - (at least conceptually) ALLOWS pretty rigorous virtualization
 - EXCEPT no virtual to (internal) physical address/port range mapping conceived in FlowVisor
 - RESTRICT to slice FlowSpace (rather than TRANSLATE) OF messages
 - OTHER features to make rigorous VIRTUALIZATION instead of SLICING should be possible (let's not blame existing solutions when not having defined our own acid tests):
 - Through FlowVisor code enhancements
 - setting proper FlowVisor Slice policies and/or
 - Setting proper OF rules at the VN edge.
- OpenFlow & FlowVisor enabled Network Virtualization:
 - ++: Slicing/virtualization ends up as regular entries in TCAM --> no performance degradation
 - ++: high flexibility in defining slices/VNs (e.g., coarse and fine grained coexist), while all layers are covered by the slices.
 - --: main focus of FlowVisor on on the OF protocol, less attention paid to edge of slices/VNs.

ACID tests: overview

- Ctrl isolation
- access control & VN labeling
- virtualization of address/port ranges
- CPU usage fairness