Software Defined Networking Exercise

Case Study II

Julius Rückert

Credits: Case Study II was originally proposed by Matthias Wichtlhuber

http://www.ps.tu-darmstadt.de/teaching/ws1516/sdn/

PS - Peer-to-Peer Systems Engineering Lab
Dept. of Electrical Engineering and Information Technology
Technische Universität Darmstadt
Rundeturmstr. 12, D-64283 Darmstadt, Germany
http://www.ps.tu-darmstadt.de/

0. Disclaimer

TECHNISCHE UNIVERSITÄT DARMSTADT

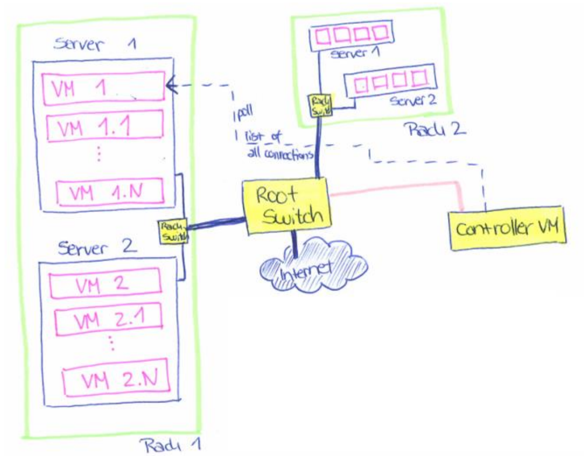
Case Studies

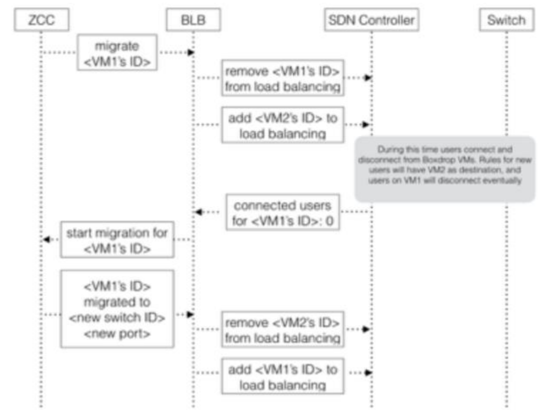
- You are supposed to demonstrate theoretical concepts defined in the lecture in an applied setting.
- Only the problem and its rough context is defined. The context may be extended, if necessary.
- You are intended to define processes and procedures to solve the problem.
- Your solution should be defined to an extent allowing a team of skilled staff to implement your solution, i.e., details may be omitted, if they do not have a large impact on your solution.
- The solution should be presented in a text-based form. Additional literature may be used.

→ There is nothing such as a right solution. Solutions presented are just one example for an acceptable solution.

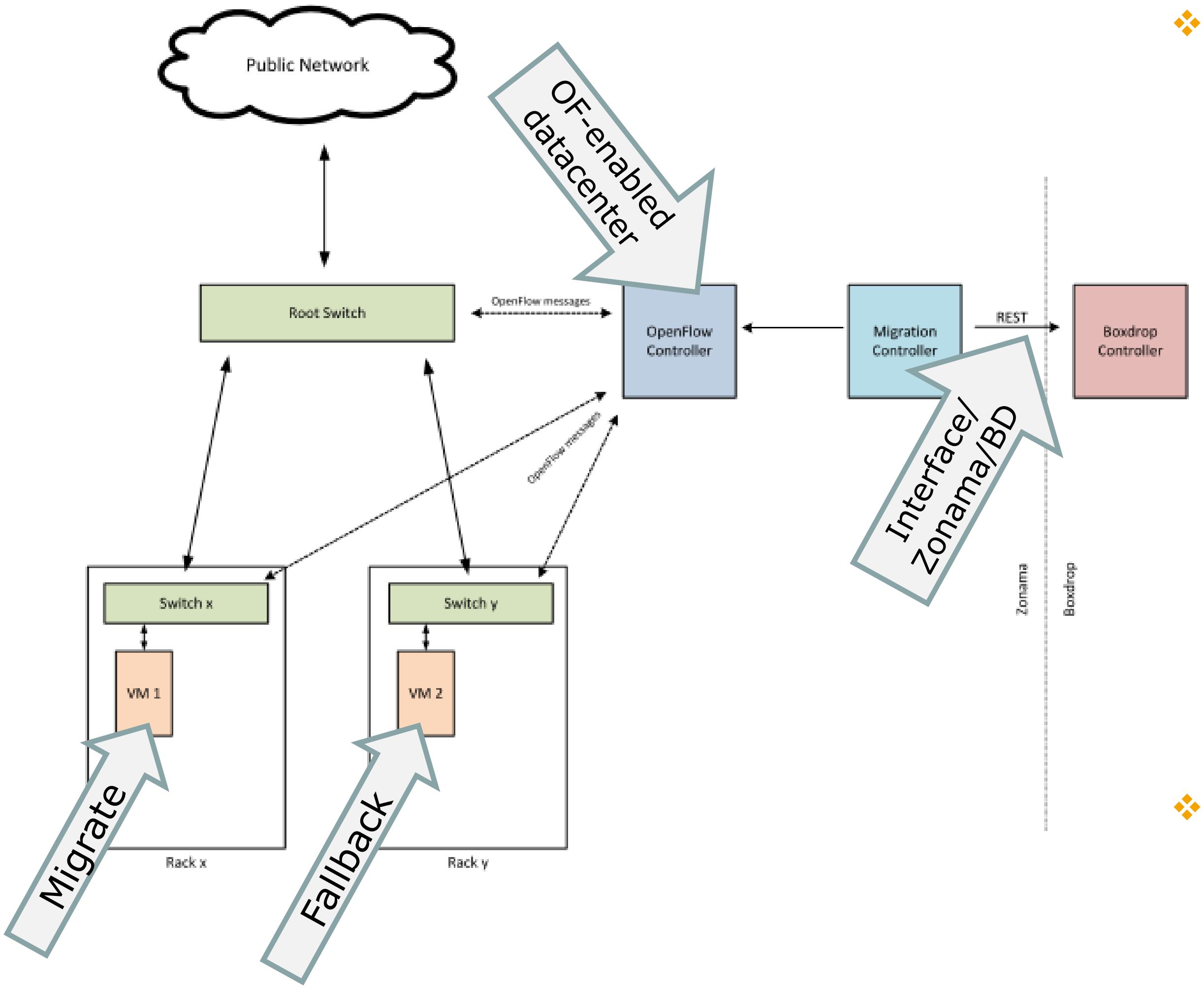
0. Remarks

- Please draw some figures!
  - Recommendation: Lucidchart (https://www.lucidchart.com)
  - Drawings by hand are also fine! (but: try to make them readable)
- We were missing more details on steps of communication
  - Sequence charts are very nice for that
- Hardly anyone specified OpenFlow rules that we asked for
  - Tables could be used here
- Deployment figures
  - Should include port numbers
- Please submit PDF files





1. Case Study – Task in a Nutshell



- **Task:** Support migration of virtual machines in Zonama's data center
  - VM 1 is to be migrated to different rack
  - VM 1 a has open connections to customers of Boxdrop
  - VM 1 informs Boxdrop on planned migration
  - Boxdrop proposes a fallback machine (VM 2)
  - Zonama redirects all new sessions intended for VM1 to VM 2
  - If VM 1 has no connections anymore, it is migrated
  - The process is reversed once the migration of VM 1 is finished

- System Architecture, components, communication, OpenFlow rules

Example Student Solution

TECHNISCHE UNIVERSITÄT DARMSTADT

- Incoming connections to Zonama VMs trigger creation of OpenFlow rules like this:

  idle_timeout=60, ip_dst=<VM IP>, ip_src=<client IP>, action=output:<n>

  - where <n> is the port on the root / rack switch that leads to the VM that should currently serve the given destination IP.
  - The control program needs to keep track of the physical locations of VMs and their hosts to make this decision.

- When a Boxdrop VM is migrated, <u>new</u> connections are forwarded to the fallback machine, while the old rules are still active and forward their clients to the old VM.

- After the last rule for the old VM has timed out, the old VM can be disabled. (This assumes that Boxdrop replicates its data so that customers can access via the new VM all the backups that they sent to the old VM.)

  - No problem: Boxdrop proposes the fallback machine.

- If necessary, a hard timeout can be implemented to avoid Denial of Service scenarios where a client keeps up a connection to the old VM indefinitely, thereby prohibiting migration.

- Pros: Straightforward and easy to understand solution; OF rules specified
- Cons: Could be more sophisticated → Per-client matching required (scalability implications)
- Missing: No figures, no explicit discussion of communication patterns between components

SOLUTION WS14/15

2. Case Study (Assumptions)

To solve the Task 1.2, I had to take a few assumptions:

- The Backup Hosts (End User Hosts) access the backup system of BoxDrop using IP (TCP).
- "Basic switching mechanism" means, that OpenFlow rules can only match L2 header fields at the current stage. I assume this can be extended by the proposed architecture
- The OpenFlow system allows to modify L2 header fields in hardware
- VM2 behaves exactly as VM1, but has another IP address (assume it is another, already running VM, which has its own customers)
- As Zonama and BoxDrop are quite large companies, it does not scale to use flow switching for every single flow
- As migration takes time, it is acceptable to have a certain time between the migration request and the actual migration
- A suitable timeout exists for existing connections, which do not have any traffic for a certain amount of time.

Given the assumptions above, I recommend the following architecture:

- *Pros:* Detailed list of assumptions, scalability is considered, pure L2 solution (cheap)
- *Cons:* Architecture diagram is missing in this case study to introduce assumptions and explain approach.

2. Case Study (Default Operation)

In general, a standard switching algorithm is used. The router (which is not part of this architecture) does an ARP request and resolves VM1s MAC address. Let this MAC address be X. Now, the packet is sent into the datacenter, where the following flow table entry exists (given that Port A is the port for the packet to reach VM1). This entry is always created, when a new VM is created. In normal operation, this basically is a standard switch, except that the MACs are not learned by the backward learning algorithm, but inserted by the controller as soon as a new VM is created.

Switch port	MAC src	MAC dst	Eth type	VLAN ID	IP src	IP dst	IP Prot	TCP sport	TCP dport	Action
*	*	X	*	*	*	*	*	*	*	Port A
*	*	Y	*	*	*	*	*	*	*	Port B

2. Case Study (Learning Phase I)

Now, VM1 needs to be migrated from PM1 to PM2. Existing connections must not be dropped. For VM1, the system now enters a learning phase. All packets which match the existing rule for VM1 are sent to the controller, which in turn creates a new rule with higher priority, which directly identifies the customer and the VM1 in a flow-based approach. These rules are now used to forward existing (and new) connections to VM1.

The table (only VM1) now looks like this (example with 2 already existing connections):

Switch port	MAC src	MAC dst	Eth type	VLAN ID	IP src	IP dst	IP Prot	TCP sport	TCP dport	Action
*	*	X	0x0800	*	Cust1IP	VM1	0x06	Sport1	Dport	Port A
*	*	X	0x0800	*	Cust2IP	VM1	0x06	Sport1	Dport	Port A
*	*	X	*	*	*	*	*	*	*	Port A FwdControl

- *Cons:* One rule per customer may limit scalability

2. Case Study (Learning Phase II)

In parallel, Zonama informs BoxDrop about the upcoming migration, and BoxDrop returns VM2. The Controller resolves the IP and then the MAC address of VM2. Let this MAC address be Y. At the current stage, all existing connections are still forwarded to VM1, existing connections using the specific entries, new connections using the initial entry (which however instantly creates another specific entry).

This learning phase must be at least as long as the application timeout, because otherwise open, but temporarily inactive connections might not be learned.

- *Cons:* Learning phase must be as long as application timeout (discussed!). This could be prevented by asking BoxDrop on the number of connections handled by VM1.

2. Case Study (Migration - failover)

As soon as VM2 is ready to accept new connections (needs to be announced by BoxDrop to Zonama) and the timeout is expired, the last "catchall" entry for VM1 needs to be changed. All packets matching this rule are now redirected to VM2 by using another port and by rewriting the destinati[illegible] MAC. The table now looks like this:

Old Connections

New Connections

Switch port	MAC src	MAC dst	Eth type	VLAN ID	IP src	IP dst	IP Prot	TCP sport	TCP dport	Action
[illegible]	*	[illegible]	0x0800	*	Cust1IP	VM1	0x06	Sport1	Dport	Port A
*	*	X	0x0800	*	Cust2IP	VM1	0x06	Sport1	Dport	Port A
*	*	X	*	*	*	*	*	*	*	Port B dstMac=Y
*	*	Y	*	*	*	*	*	*	*	Port B

At this stage, existing connections are still forwarded to VM1 (line 1-2), while new connections to VM 1 are now transparently redirected to VM2 (line 3). Line 4 represents the existing entry flow entry for VM2 (see default operation).

- *Pros:* Splitting of old and new connections can be handled elegantly.

2. Case Study (Migration - failover)

At this stage the system waits for the existing connections to be closed. As soon as a connection is closed, the controller removes the specific flow entry from the table. Finally, when all connections are closed, the table looks as follows:

Switch port	MAC src	MAC dst	Eth type	VLAN ID	IP src	IP dst	IP Prot	TCP sport	TCP dport	Action
*	*	X	*	*	*	*	*	*	*	Port B dstMac=Y
*	*	Y	*	*	*	*	*	*	*	Port B

Thus, all traffic is forwarded to VM2 now.

- *Remark:* Active removal is not necessary, as flow table entries can be equipped with a timeout *t*. If the switch does not see a packet in this period, the entry is removed.

2. Case Study (Migration)

TECHNISCHE UNIVERSITÄT DARMSTADT

Migration

Now, VM1 can be migrated without any user traffic on it. As soon as the migration has finished, the process starts over again. Connections to VM2, which were originally sent to VM1, are learned and flow entries are created. After the timeout, the rule changing the MAC address is removed and replaced by the normal entry for VM1.

3. Further Remarks

- **Some solutions relied on detecting a TCP SYN ACK packet**
  - Remark: This feature is only available from OpenFlow version v1.5.0. State assumptions on version to be used.

- **Authorship**
  - For this solution material from two case studies was used
  - If the authors wish to get credits in the slides, please contact me
    - rueckert@ps.tu-darmstadt.de