TNG TECHNOLOGY CONSULTING

Software Defined Networking

A quantum leap for Devops?

TNG Technology Consulting GmbH, http://www.tngtech.com/

Networking is bottleneck in today‘s devops

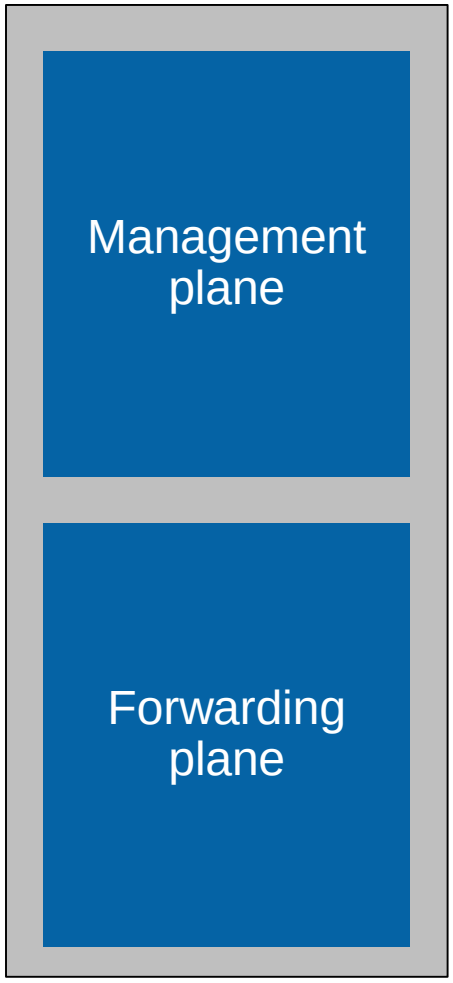
- Agile software development and devops is increasing pressure on operations departments due to more rapid changes
- Server and storage teams are increasing speed and flexibility with virtualization
- Network and security teams are becoming the bottleneck

Networking needs to become more flexible

Classical networking

- Classical network devices are autonomous, loosely coupled
  - Each device must be configured separately
- Lots of appliances for Firewall, VPN, intrusion detection, load balancing, web proxy / filtering etc.
  - Complex setup due to different configuration methods
  - Configuration discrepancies may result in outages

What‘s in the network box?

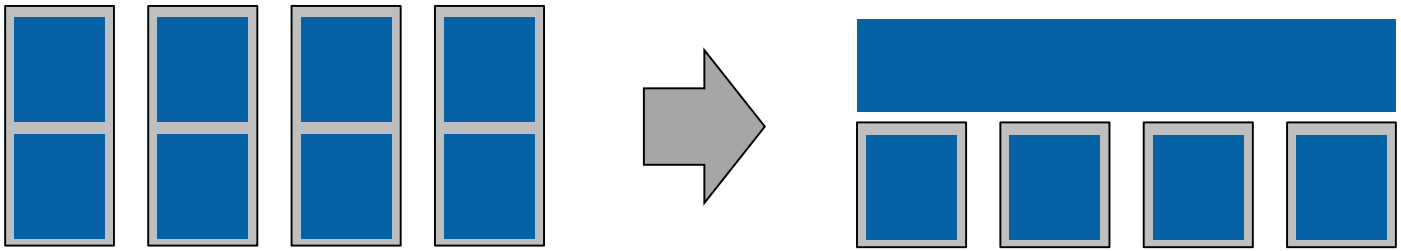


- „Intelligent“ part of the device
- Runs ARP, Spanning Tree, IGMP, OSPF etc.
- Based on industry-standard CPU
- Runs an operating system (often proprietary)

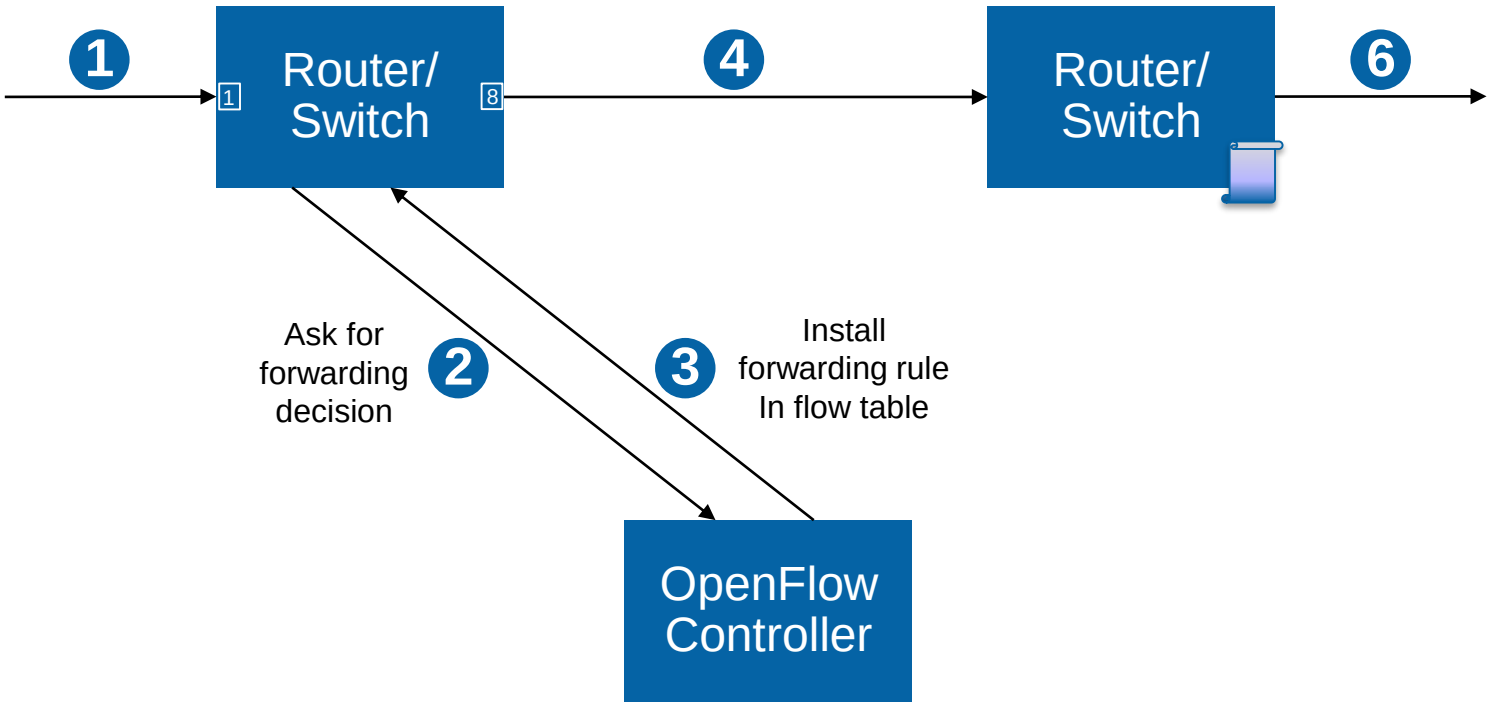
- Packet forwarding
- Contains specialized hardware
- Very little logic involved

Software defined networking: OpenFlow

- The management plane is centralized and called „controller“
- It decides about forwarding decisions and paths for the whole network
- OpenFlow works on layers 2-4 of the OSI model



OpenFlow Example



Switch port	MAC src	MAC dst	VLAN ID	MPLS label	IP src	IP dst	IP TOS	TCP sport	TCP dport	*Action*
Port 1		AABBCC DDEEFF				1.2.3.4			80	*Port 8*

Only forwarding of packets?

- OpenFlow can be used for other things
  - Load balancing
  - Firewalling
  - Intrusion detection / prevention
  - Traffic accounting
  - Tracing / traffic mirroring
- We won‘t need a zoo of networking appliances any more!

OpenFlow Controllers have APIs

- Automate network configuration deployment
- Generation of forwarding rules based on configuration management (e.g. puppet)
- Easy to replicate production constraints in development and test

```
class network::dmz::web {
  firewall { 'accept https':
    proto   => 'tcp',
    source => all,
    dport => 443,
    action  => 'accept'
  }
}

node 'www.tngtech.com' {
  include network::dmz::web
  include network::loadbalancer::https
  vip => "217.110.29.214"
  realservers => "vm1", "vm2"
}
```

Benefits

- Automated, fast changes are possible throughout the whole network
- No discrepancies between network and host configuration

- Application development can develop communication rules
  - No surprises during deployment

Security benefits

- Responsibilities can be split by application, not by network device
- Security improvement due to accountability
  - Change control for all configuration changes
  - No undocumented, old firewall rules any more
  - Rule sets will match currently deployed applications

Networking tests in continuous delivery

- Virtualization hosts contain a software switch
- Hypervisor networking can be controlled via OpenFlow
- I have an enterprise class network in my laptop
  - and in my Jenkins server!
- A complete end-2-end view of the network can be set up in one virtualization host
  - No "*We can't afford a load balancer for testing*" any more

Software development best practices can be applied

- Firewall and load balancing rules can be tested without actual deployment
  - Just simulate a switch, send queries to the OpenFlow controller and evaluate the answer
  - Unit tests for firewall rule sets, anyone?
- Packaging for network functions
  - Repositories, versions, dependency management
- Compliance to security policy can be checked similar to code quality metrics
  - Sonar for network security

Vendor support

- All networking vendors have some kind of SDN story
  - Cisco, Juniper, HP, Arista, Brocade support OpenFlow
  - OpenFlow products partially available,
    product support is being extended
- Every vendor wants to own the controller,
  nobody wants to deliver dumb iron
- Google is using pre-standard version of SDN for private connectivity between their datacenters („G-scale network“), Amazon uses a unknown / proprietary SDN implementation in EC2.

What‘s still missing

- Good solutions for integration between classical network device configuration and OpenFlow networks
- Vendor interoperability is still in early stages
- Standards for northbound APIs (e.g. OpenDaylight project)
  - Necessary for higher abstractions

OpenFlow is not yet production-ready,
but is likely to be the next major
evolutionary step of networking.

Be prepared!