Cloud Networking – an Enterprise View
Enterprise clouds – multiple delivery models

- implications on the delivery network
- levels of sharing (infrastructure, services, management …)
- security / isolation / privacy / compliance
Enterprise clouds – management “up the stack”

- hypervisor security patching
- hypervisor incident mgmnt
- storage for virtual images
- shared services
- ...

- security patching, software updates
- high-availability, cluster management
- monitoring
- backup / recovery
- SLAs and reporting
- user management
- storage configuration (related to OS)
- application onboarding
- ...

public cloud service

managed cloud service

© VMware, Inc.
Enterprise clouds – tools and infrastructure complexity

- Multitude of tools – some need globally unique endpoints
- Multiple isolated networks for management, mobility, backup
- Integration with on-premise tools and systems
- Multiple risks of address overlap
Networking support in current cloud offerings

- Adding features but limited control of the network
  - requires integration of third-party solutions
  - limits the opportunity to migrate production applications

Examples of Missing Features
- No multicast or broadcast
- No ability to create VLANs
- No facility for bandwidth or QoS
- Limited crafting of network segments
- No dynamically structured networks
- Limited network mgmnt or visibility
- No IPv6 support

Anatomy of an enterprise application moving to the cloud

**Configurations to preserve:** IP addresses, logical topology, firewall rules, VLAN, network bandwidth, fail-over plan, LDAP, DNS, ...

*Firewalls rules* must be validated and migrated. Multitenant cloud environment and mobility introduces additional challenges.

*All IP addresses* must be discovered. End-host reconfiguration is not scalable and are error prone.

Network bandwidth requirements for better VM placement for performance.

Failure planning for VM placement considerations.

VLAN migration requires reconfiguration of all routers and firewalls.

Shared network services remain on premise.
Networking-as-a-service for enterprise clouds

- Allow enterprises to re-create their on-premise network configuration in the cloud
- Unified framework for deploying applications and corresponding network services
- Provide a service-centric, rather than network device centric view

- **Cloud controller**
  - provides base IaaS service for managing VM instances and images
  - self-service provisioning UI
  - connects VMs via host virtual switches

- **Network controller**
  - works with cloud controller to provision virtual network services
  - provides VM placement directives to cloud controller
  - configures physical and virtual switches
User abstractions for specifying cloud networking functions

- traffic is allowed to flow only over explicitly defined virtual network segments ("default off")
- can provide standard templates to implement security policies, or application requirements

**group**
- logical grouping of VMs

**address**
- assign a custom address to the VM

**middlebox**
- instantiate and config a new middlebox

**virtualnet**
- segments connect groups of VMs
- associated with network services

**networkservice**
- attach capabilities to a virtualnet
- supports combination of network services

129.2.200.5

- middlebox
- resv bandwidth
- VLAN / scoped bcast
- ...

**EXTERNAL**
Example application with network policy specification

1. **address** dbserver1={128.1.104.103}
2. **address** dbserver2={128.1.104.13}
3. **group** frontend={httpdserver}
4. **group** businesslogic={jboss1, jboss2, jboss3}
5. **group** backend={dbserver1, dbserver2}
6. **middlebox** Class={type=classifier, config=""}
7. **middlebox** DPI={type=dpi, config=""}
8. **networkservice** protectfrontend={l2broadcast=no, qos=standard, mb=DPI}
9. **networkservice** besteffort={l2broadcast=no, qos=standard, mb=none}
10. **networkservice** reservedbw={l2broadcast=no, qos=10mbs, mb=Class}
11. **networkservice** allowfailover={l2broadcast=yes, qos=standard, mb=none}
12. **virtualnet** allowFailover (backend)
13. **virtualnet** protect FrontEnd(frontend,EXTERNAL)
14. **virtualnet** besteffort(frontend,businesslogic)
15. **virtualnet** reservedbw(businesslogic,backend)
CloudNaaS: A Cloud Networking Platform for Enterprise Applications
IBM Research / University of Wisconsin collaboration

- Cloud Controller: OpenNebula 1.4
  - modified to accept user-specified network policies and interact with the Network Controller
  - minimal modifications (~250 LOC)
  - network policy parser (~250 LOC)

- Network Controller: NOX and OpenFlow-enabled switches
  - HP Procurve 5400 switches w/ OpenFlow 1.0 firmware
  - network controller implemented as a C++ NOX application (~2500 LOC)
  - pulls new / updated communication matrices and VM mappings from Cloud controller
  - interfaces to non-OpenFlow switch-specific functions (e.g., queue management)

- End-host virtual switches: Open vSwitch
  - built-in support for OpenFlow protocol
Challenges at Cloud scale

- Evaluation: experimental and emulated
  - workloads:
    - multi-tier business application (e.g., SAP R/3)
    - enterprise search / analytics (e.g., MS SharePoint)
  - topologies: standard 3-tier, fat tree
    - 6K – 30K hosts, 200 – 1000 ToRs, 20 – 100 agg

- Computation and instantiation of network services
  - ~16K instances of 3-tier Web service (270K VMs) requires about 120s in experiments

- Recovering network paths / services when links or switches fail
  - Network controller takes 2 – 10s for recomputation in a large DCN (1000 ToR/100 agg/270K VMs) when a link fails
  - Can be reduced to 0.2s by precomputing solutions for core links
  - Switch failures require an order of magnitude more time to recover

- Managing hardware device limitations
  - $O(V^2N)$ forwarding entries per device ($V =$ #virtual networks; $N =$ #VMs)
  - TCAM space in switches may only support 2000 flow table entries
  - Optimizations can reduce in-network state (e.g., destination-based forwarding, entry aggregation with network-aware VM placement)
Research implications of enterprise clouds on the network

- Overcoming the scaling limitations of current network devices
  - table sizes: MAC addrs, ACL / TCAMs, VLANs, priority levels, …
  - dynamic updates: changing forwarding tables, queuing rules, etc.
  - take better advantage of device capabilities

- Management integration
  - provide a more comprehensive network management view to tenants
  - integrate the network with adjacent processes and tools
  - compensating for legacy tools and applications in the cloud

- Flexibility with simplicity – make it easy to write network apps
  - reconfigurable / optimized topologies, agile routing
  - leverage the emerging SDN abstractions approach
  - get above the “CCIE interface” to the network
Additional material
Cloud networking standards and models are evolving quickly

- **OpenStack Project**: open source cloud operating system
  - base Nova networking focuses on address management
    - flat subnet, flat subnet + DHCP, per-project private VLAN with OpenVPN access

**Networking services development**

- **Melange**: flexible services for IP addr management
- **Quantum**: virtual network service to create L2 networks, ports, attachments, connectivity
  - Open vSwitch Quantum plugin available (Nicira), Cisco Nexus/UCS
- **Donabe** (Network Containers): APIs to manage generalized resource container abstraction – network containers are a first instance of containers

- **Commercial cloud solutions**
  - Amazon: variety of network functions, incl. VPCs with subnets / ACLs
  - MS Azure: basic addresses / connectivity, VPC, CDN
  - VMWare: vCloud Director Networking (VXLAN) – isolated virtual networks
Related work from industry and academic research

- Network-related services and appliances from 3rd party ISVs / providers
  - require integration of multiple solutions, individual service models and functions

- Research proposals on cloud networking abstractions
  - single virtual router [Keller:10]; virt data center + bw guarantees [Guo:10]
  - access control services [Popa:10]
  - virt private cloud [Wood:09], WAN workload migration [Wood:11]

- Multi-tenant virtual networks
  - many proposals (see recent SIGCOMM, NSDI, CoNEXT for examples)
**Results**

- Optimizations allow support of 3X more VNs
  - Most savings at the core
- VM placement allows even better scaling
  - Applications supported: 4X

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Virtual switch</th>
<th>ToR</th>
<th>Aggregation</th>
<th>Core</th>
<th># of Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default Placement</td>
<td>313</td>
<td>13K</td>
<td>235K</td>
<td>1068K</td>
<td>4k</td>
</tr>
<tr>
<td>Default placement + Optimizations</td>
<td>0%</td>
<td>93%</td>
<td>95%</td>
<td>99%</td>
<td>12.2K</td>
</tr>
<tr>
<td>Placement Heuristic + Optimizations</td>
<td>0%</td>
<td>99.8%</td>
<td>99%</td>
<td>99%</td>
<td>15.9K</td>
</tr>
</tbody>
</table>