

Network Virtualization in Multi-Tenant Datacenters

Com a introdução da virtualização em servidores é possível ter várias imagens a correr na mesma máquina, o que diminui a quantidade de hardware necessário. Porém, se for introduzida uma nova aplicação ou um novo ambiente de desenvolvimento é preciso efetuar alterações na rede, tais como: na topologia e no espaço de endereçamento. Assim, os autores pretendem mitigar estes requisitos, fazendo com que os utilizadores possam mover os seus serviços, mantendo as suas configurações, sem que haja alterações na rede.

Para o suporte das VMs irá existir uma camada, que tem um *switch*, que aceita os pacotes, de forma a reencaminhar para uma VM local ou para uma outra camada com as mesmas funções, localizada noutra máquina. Esta camada chama-se *hypervisor* e este paradigma irá ser usado na arquitectura do NVP (*Network Virtualization Platform*), como *network hypervisor*.

O *network hypervisor* está localizado entre a infraestrutura física do fornecedor e o *control plane* do *tenant*. Este tem como objectivo, fornecer abstrações apropriadas sobre o controlo e os pacotes. Em relação à abstração do controlo, esta tem de permitir que os *tenants* possam definir um conjunto de elementos lógicos da rede (*datapaths* lógicos). Estes *datapaths* lógicos estão implementados nos *switches* virtuais, da mesma maneira que são implementados nos físicos, ou seja, com um *pipeline* de reencaminhamento de pacotes e tabelas de *matching*. Relativamente, à abstração de pacotes, é requerido que os pacotes que sejam enviados de *tenant* para *tenant* tenham o mesmo reconhecimento e tratamento (*switching*, *routing* e *filtering*) nos dois lados, como se se tratasse de uma rede local.

Na arquitetura é usado um controlador SDN centralizado, que se responsabiliza por configurar os *switches* virtuais do *cluster*, com as regras de reencaminhamento adequadas para cada *tenant*. Os autores tiveram de encarar uma série de problemas, como por exemplo a implementação dos *datapaths* e modificações que pudessem surgir para permitir o encapsulamento a grandes velocidades; minimizar o custo da computação, garantir consistência nas ordens dos eventos e suportar modificações na rede, isto devido à programação no controlador; por fim, o escalonamento do controlador do cluster.

Em relação à computação da rede, o controlador tem de saber reagir de imediato quando surgem alterações (seja por novos *tenants* terem entrado na rede, configurações da rede lógica ou mesmo a configuração física do sistema) e para isto usaram o *nlog* (uma linguagem declarativa) para lidar com os eventos que ocorrem na rede. O *nlog* permite computação incremental de todos os estados de encaminhamento, de forma eficiente através de queries em Datalog, e permite separar as especificações lógicas, do estado da máquina. Assim, foi possível garantir escalabilidade, dividindo o processo de computação em duas camadas, possibilitando múltiplos controladores a correrem no cluster. Estas duas camadas representam controladores lógicos e físicos, distintamente. Os controladores lógicos serão responsáveis pelos fluxos e túneis que são necessários para implementar os *datapaths* lógicos. Esta será a camada mais acima do processo, que depois irá enviar fluxos universais (codificação das tabelas de pesquisa dos *datapaths* com instruções para criar túneis e filas) para os controladores físicos. Estes têm a tarefa de comunicar com os *hypervisors*, *gateways* e nodos de serviços, fazendo a tradução dos fluxos universais (endereços IP, portas físicas, entre outros) de forma a enviar fluxos físicos (reais) para os nodos que realizam o transporte.

Existem outros mecanismos de virtualização de redes para datacenters, como as VLANs, porém estas usam um mecanismo de *slicing*, o que no ponto de vista dos autores não fornece as abstrações de pacotes e controlo que permitem aos *tenants* “viver” numa rede lógica fiel. Também foi proposto implementar VM’s para virtualizar os *routers*, porém não é uma solução escalável.