

NOX: Towards an Operating System for Networks

Desde o início da computação existiam certas dificuldades em programar sistemas de informação, na medida em que não existia uma abstração entre os recursos físicos e as aplicações. Após isto, surgiram sistemas operativos que vieram facilitar esta tarefa. Com base nisto, os autores decidiram que era preciso criar um sistema operativo direcionado para as redes que, através de uma interface, se pudesse observar e controlar uma rede inteira. Porém existem certos desafios na implementação deste S.O. tais como, manter a independência das duas partes (física e aplicacional) e ainda assim conseguir ligá-las e sabendo que a rede é propícia a aumento de componentes, é necessário que este seja capaz de gerir este crescimento – escalabilidade.

Os autores adotaram o paradigma de ter servidores a correr o NOX, à parte das componentes da rede (e.g *switches* OpenFlow), como já é sabido atualmente pelo conceito das SDNs. Estes servidores têm o propósito de controlar os fluxos da rede e ter uma visão global da rede. Um aspeto de *design* importante que tiveram em conta foi a granularidade, pois esta envolve uma troca entre a escalabilidade e a flexibilidade, ambas importantes para gerir grandes redes. Portanto, a visão da rede inclui a topologia do nível dos *switches*, a localização dos utilizadores, *hosts*, bem como outros elementos da rede e, finalmente, os serviços oferecidos. O que não inclui é o estado atual do tráfego da rede, isto para que possa ser escalável em redes de larga escala, mas, ainda assim, permitindo ter um controlo flexível. Resumidamente, o NOX fornece observação e controlo de uma rede. Para a abstração dos *switches* foi adotado o OpenFlow, pois as instruções dos *switches* devem ser independentes do hardware e suportar o controlo de granularidade do nível de fluxo. Relativamente ao processo feito nos *switches*, e que é descrito pelos autores, é que se um pacote ao chegar a corresponder a uma entrada de fluxo no *switch*, este atualiza os contadores correspondentes e realiza as ações necessárias. Se o pacote não corresponder a uma entrada, então é reencaminhado para o processo controlador. O NOX utiliza estas inicializações de fluxo e reencaminhamento de tráfego para construir a visão geral da rede (observação) e determinar se deve reencaminhar ou não o tráfego, caso seja necessário, este tem de saber qual a rota a ser tomada (controlo).

A interface de programação do NOX baseia-se em três fatores (*events*, *namespaces* e *network view*), que serão explicadas de seguida.

Os eventos são ações que ocorrem no normal funcionamento da rede, tais como, fluxos ou utilizadores a chegar e a partir ou *links* a ligarem-se ou desligarem-se. As aplicações registam handlers para executarem quando certos eventos ocorrem e estes são executados pela sua prioridade, que é estabelecida quando são registados. Certos eventos são gerados diretamente por mensagens do OpenFlow, outros são gerados por aplicações do NOX ao processar eventos de baixo nível ou eventos gerados por outras aplicações. Os autores implementaram serviços com eventos associados para depois estes poderem ser utilizados por outras aplicações. Através da construção da visão global da rede, são mantidos *namespaces* de alto nível que podem ser usados por aplicações externas. Estas aplicações são usadas para a autenticação de utilizadores, de *hosts* e, por monitorizarem o DNS deixam que outras aplicações convertam nomes de alto nível em endereços de baixo nível, ou vice-versa. Como a visão da rede tem de ser consistente e tem de estar disponível para todas as instâncias dos controladores, as aplicações apenas devem modificá-la quando uma alteração ocorre na rede e não por cada pacote recebido. A partir da camada de abstração criada pelo OpenFlow, as aplicações podem editar a tabela de fluxos e ter o controle das rotas da L2, podem alterar os cabeçalhos dos pacotes e ter o controlo sobre os acessos (ACLs). O NOX inclui um conjunto de bibliotecas implementadas no sistema que contêm funções eficientes que são usadas na maioria das aplicações de rede.

Apesar de do NOX ter muitas vantagens, esta interface contém algumas limitações. Cada aplicação que seja mal implementada pode prender eventos e ocupar memória desnecessária ou criar um ciclo infinito. A verificação das aplicações ainda não está implementada nesta versão do NOX.

A ideia apresentada pelo NOX foi inicialmente desenvolvida no contexto do projeto 4D. Este fornecia uma visão que requeria um paradigma baseado em *master-slave*, centralizando o poder decisivo num elemento que vê a rede toda. Este paradigma centralizado é muito mais flexível, visto que apenas se pode programar o elemento decisivo em vez de um algoritmo distribuído, mas, no fim acabamos por ter um problema de segurança, que é um único ponto de falha. No entanto, podemos conseguir fiabilidade, aplicando técnicas de replicação ao elemento decisivo e visto que estas técnicas são completamente desacopladas do algoritmo de controlo de rede, há sempre espaço para inovar.