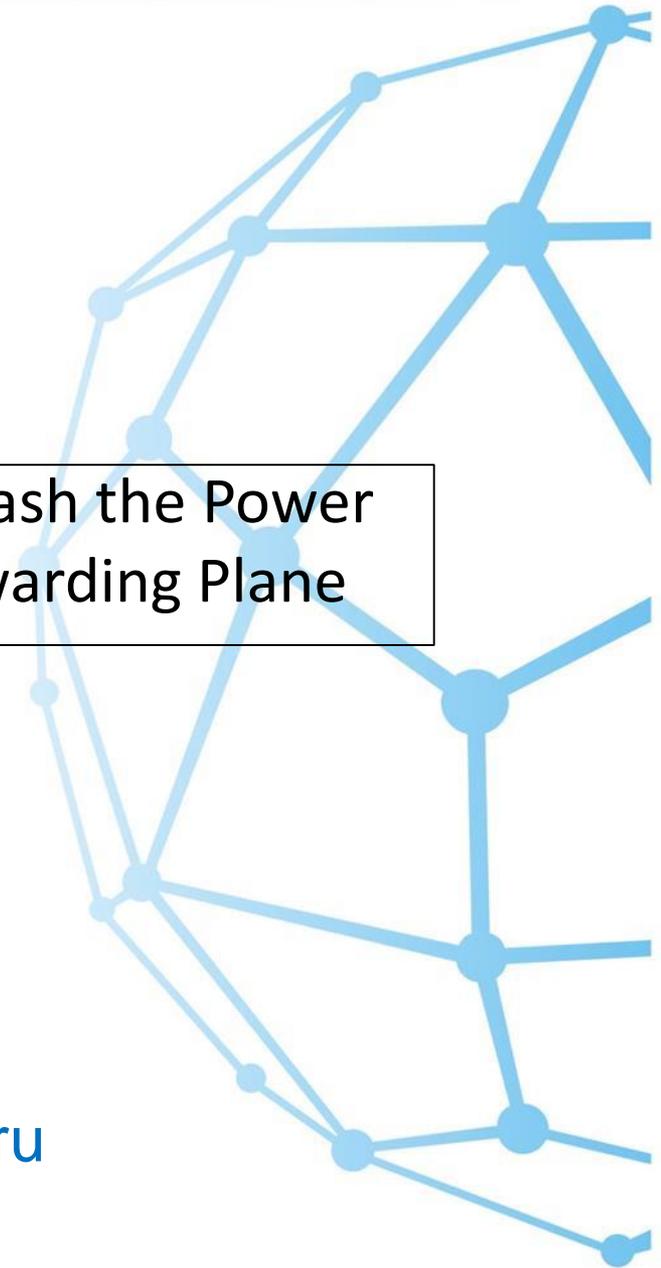


ЦЕНТР
ПРИКЛАДНЫХ
ИССЛЕДОВАНИЙ
КОМПЬЮТЕРНЫХ
СЕТЕЙ

Protocol-Oblivious Forwarding: Unleash the Power of SDN through a Future-Proof Forwarding Plane

Haoyu Song
Huawei Technologies, USA
Santa Clara, CA, 95050
haoyu.song@huawei.com

Вячеслав Васин: vvasin@arccn.ru



Постановка задачи:

OpenFlow являясь de facto стандартом общения контроллера и коммутатора имеет ряд недостатков:

- В процессе своего развития он тем не менее продолжает быть ориентирован на используемые в сети протоколы (IPv4, IPv6, MPLS) и заголовки (L2, L3, L4). Что является анахронизмом, оставшимся от сетей с объединенным уровнем управления и передачи данных.
- Уровень передачи данных не контролирует состояние потоков (stateless). Не может мониторить статус и изменять поведение потоков без участия контроллера. OF должен иметь возможность управлять более интеллектуальными возможностями аппаратуры.

Как следствие, это приводит к следующему :

- По мере появления новых протоколов будут усложняться требования к стандарту и к коммутаторам, что усложнит их дизайн.
- Нет возможности произвольно изменить формат пакета идущего по сети, добавить новые произвольные поля.
- Добавление новых протоколов приведет к необходимости переработки и протокола и ПО коммутатора, а возможно и к переработке аппаратной части, что каждый раз будет приводить к задержкам на время реализации новой опции и тормозить прогресс.
- OF не имеет возможности обрабатывать трафик с учетом L4-L7 сервисов на уровне DataPlane. Даже реализация L2 learning приводит к проблемам производительности и масштабируемости.

Каким должно быть будущее SDN:

Текущая реализация – black boxes

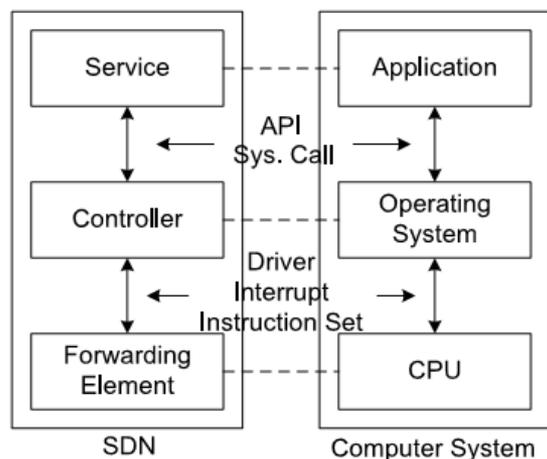
OpenFlow – gray boxes

- Ситуация улучшается, так как добавляется программируемость.
- Но программируемость ограничена возможностями контроллера и аппаратуры коммутатора.

Миру SDN нужна настоящая программируемость

- Forwarding plane должен быть white box, поведение которого полностью подчиняется контроллеру.

Аналогия SDN и PC



- CPU может делать все с помощью небольшого количества имеющихся инструкций.
- CPU взаимодействует с OS с помощью стандартных инструкций и прерываний.
- OS позволяет абстрагироваться от железа и предоставляет приложениям высокоуровневый интерфейс.

- FE в SDN должен стать аналогом CPU в PC. (RISC)
- FE не должен знать заранее с каким трафиком ему придется иметь дело.
- Нужно идти по пути упрощения, к более общей модели FE.
- Нужны независимые от протоколов инструкции, которые предлагается назвать Flow Instruction Set (FIS)
- FIS могут быть использованы для разбора пакетов, анализа данных за счет поддержки инструкций, реализующих логику сравнения, ветвления.
- Каждый такой примитив должен быть легко реализуем в платформе.
- Конструирование сетевого сервиса на такой основе аналогично программированию на Ассемблере, что может быть компенсировано в будущем высокоуровневым языком и библиотеками.

Protocol Oblivious Forwarding:

POF FE не обязан понимать формат пакета в отличии от OpenFlow FE.

FE должен по указанию контроллера:

- Анализировать заголовки пакетов на основе указаний контроллера
- Осуществлять поиск по таблицам
- Выполнять соответствующие инструкции

Взаимодействие с контроллером

- Контроллер описывает составные части ключа поиска в терминах {offset, length}+{offset, length}...
- Вместо стандартных действий (push vlan, swap mpls...) используются инструкции AddField {offset, length} или задействуются данные из metadata.
- Есть инструкции описывающие математические (Add, Substr, Shift) и логические (AND, OR, NOT, XOR) операции
- Для сложных функций – есть специальные инструкции (IP checksum, TCP checksum), определяя какие данные использовать, какой алгоритм и куда поместить результат.

Преимущество

- В случае появления необходимости обработать новый протокол или комбинацию (VXLAN, NVGRE, VLAN+MPLS+PBB) нет необходимости модернизировать стандарт OpenFlow.

Stateful FE

Data plane хранит информацию о статусе потока используя flow metadata (не путать с packet metadata)

- Содержит информацию о потоке (подобно counter и timeout в OF)
- Выделяется не для всех потоков (on-demand)
- Метаданные могут быть использованы сетевыми приложениями.
- Данные могут быть разных типов (Sequence #, time stamp)

FE может самостоятельно манипулировать потоками (без указания контроллера)

- По событию или при выполнении заранее определенного условия может быть:
 - создана таблица;
 - создан, удален, обновлен flow в таблице;
- Об этих действиях FE информирует контроллер для синхронизации состояния контроллера и сети.

Единый набор инструкций подходящий для множества потоков. Вызов инструкций с параметрами.

- Внутри блока инструкций существует возможность перехода на различные ветви операций с помощью условного и безусловного перехода.

Дополнения

Выделение счетчика статистики только по необходимости по запросу контроллера

- Экономия количества счетчиков и используемой памяти.
- Может быть один счетчик на несколько потоков и несколько счетчиков для одного потока
- Возможность иметь агрегированные счетчики

Дифференциация типов таблиц (id, capacity, parameters, type):

- Direct Table (DT)
- Exact Match (EM)
- Longest Prefix Match (LPM)
- Masked match (MM)

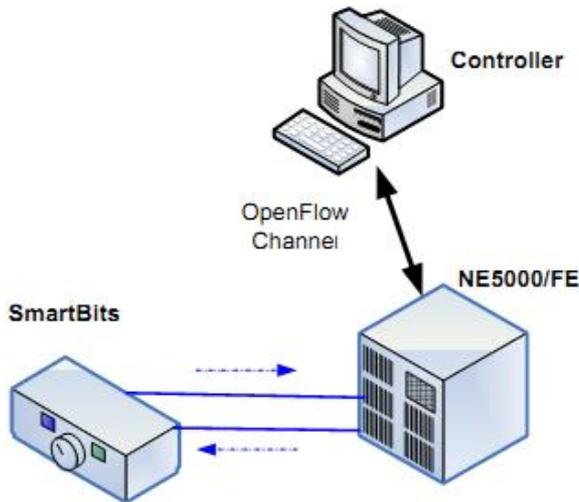
Возможно дальнейшее расширение в будущем:

- Range Match (RM)
- RegEx Match (REM)

Физические и логические таблицы (могут ссылаться на одну физическую)

- Одни и те же записи могут использоваться несколькими логическими таблицами но с разными инструкциями (IP routing, RPF)
- Несколько логических таблиц могут сосуществовать в рамках одной физической (несколько EM таблиц в одной физической Hash таблице)

Прототипирование



POF реализация на тестовом образце замедляла пропускную способность примерно на 20-30%

Надежды на POF optimized silicon

Контроллер на базе Floodlight + POF extension

- Open source

Hardwire based FE prototype

- Huawei NE5000 core router
- 40G NPU with On-chip and Off-chip memories
- 8x1G port

Software based prototype

- Linux based
- Written on C
- Open source

Use case

- Поддержка новых протоколов (VXLAN, NVGRE, FC, ATM, POS)
- Поддержка Named Data Network (имена вместо адресов)
- Использование наиболее эффективных форматов пакетов
- Трансформация пакетов, организация туннелей.
- Возможность добавлять к пакетам на границе сети любые новые поля – новые возможности для OAM
- Смещение внимания с поддержки протоколов к сервисам
- Перспектива реализации DPI, TCP flow state, NAT, Firewall

Выводы

В SDN должна быть более простая и обобщенная модель уровня передачи данных (FE).

- В рамках модели контроллер должен определять для FE не только ЧТО делать, но и КАК это делать.

Для сохранения совместимости с текущими версиями необходимо добавить POF инструкции в спецификацию OpenFlow

- В идеале для поддержки POF должны появиться новые чипы.
- В текущих реализациях NP поддержка POF может реализовываться как функция, описывающая POF инструкцию
- В ASIC предлагается реализовать POF инструкции в аппаратуре что даст выигрыш в производительности.

Подчеркивается важность исследовательских работ и работ по стандартизации под эгидой ONF в развитии OpenFlow.

Спасибо!

Вячеслав Васин: vvasin@arccn.ru

