Сериализация и проактивная конфигурация сетей

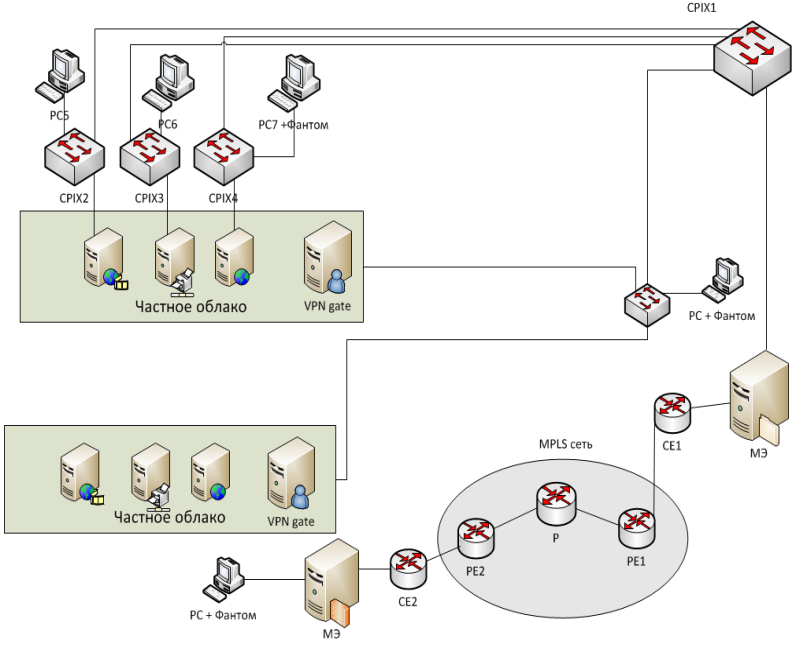
Писковский Виктор Олегович

План

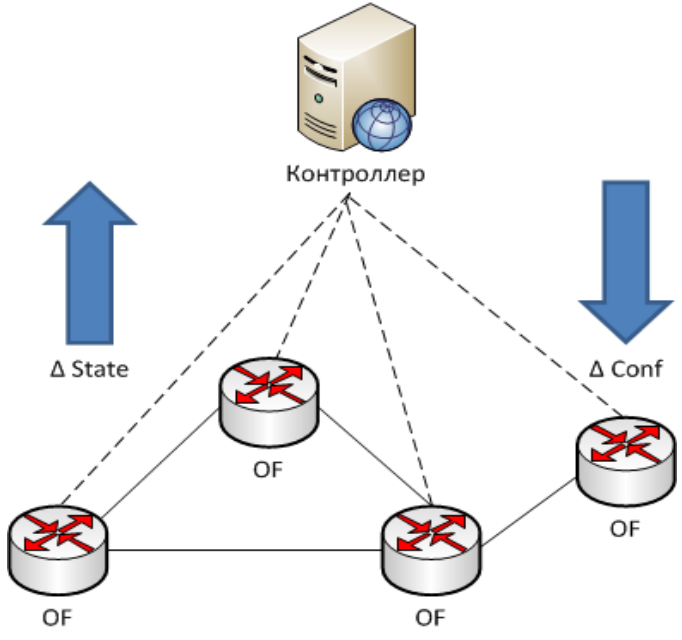
1. Традиционные и программно-конфигурируемые сети
2. Описание конфликтов и методы разрешения, решение Вермонт
3. Процедура реконфигурации и аномалии
4. Средство разработки приложений управления ПКС-сетью (fNetKAT)

Традиционная сеть и ПКС

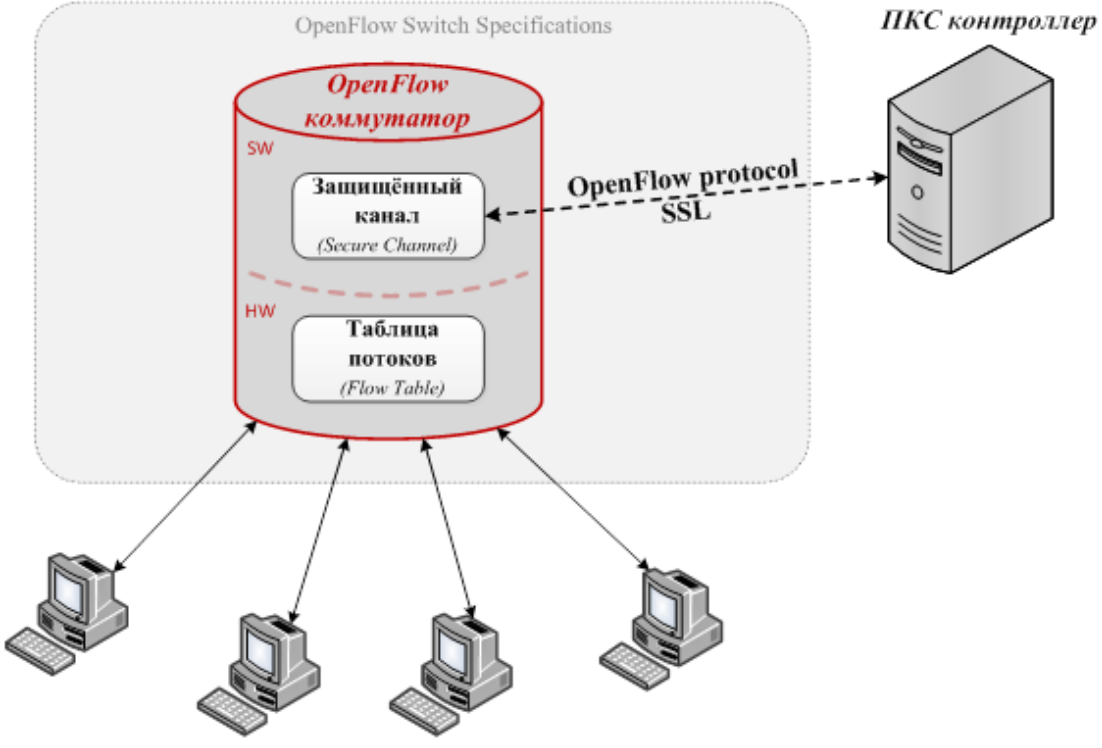
Традиционная сеть



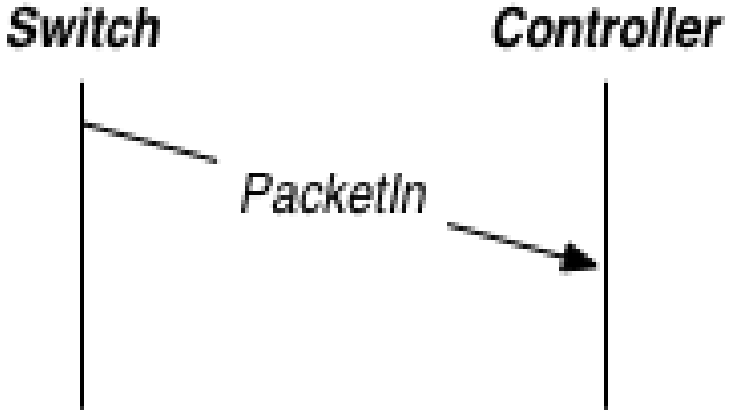
Программно-конфигурируемая сеть



OpenFlow


OpenFlow Switch Specifications
OpenFlow коммутатор
SW
Защищённый канал
(Secure Channel)
HW
Таблица потоков
(Flow Table)
OpenFlow protocol
SSL
ПКС контроллер

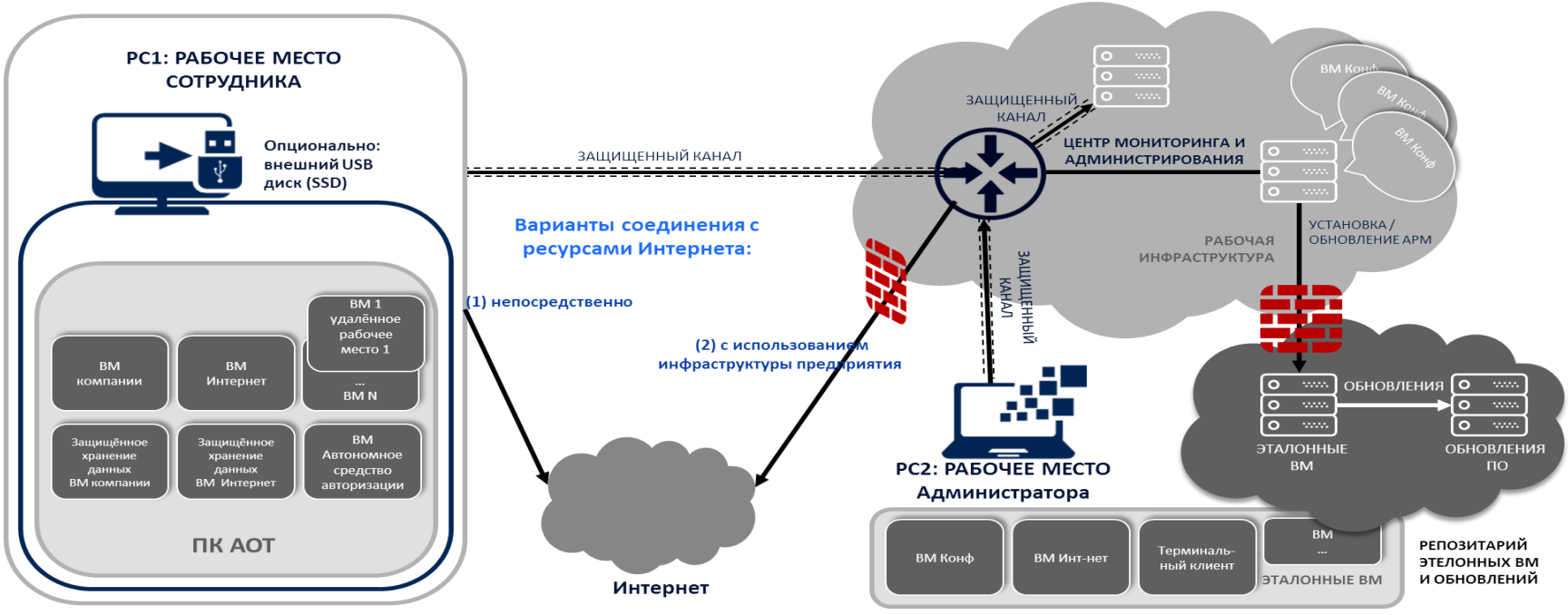
PacketIn



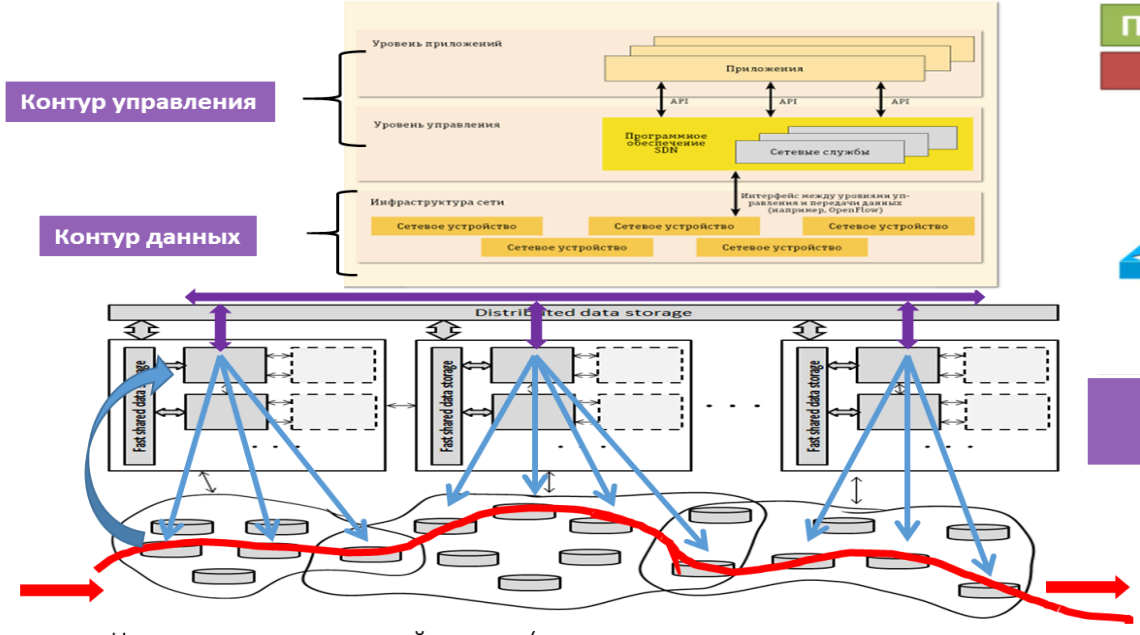
Сообщение PacketIn может возникнуть в двух случаях:

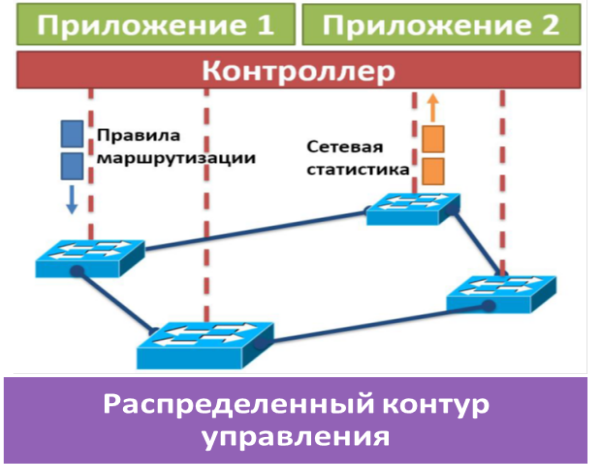
- Как явная реакция на присутствие некоторого правила требующего данного поведения или отсутствия запрошенного ранее правила
- Произошла TTL error

Принципиальная схема



Архитектура ПКС сети





Распределенный контур управления

- Надежность и отказоустойчивость (резервирование внутри кластера и между кластерами)
- Балансировка нагрузки (активизация новых узлов контроллера в зависимости от нагрузки)
- Согласованное управление и видение всей сети
- Работа с распределенными сетевыми приложениями
- Безопасность и противодействие внешним нагрузкам

Внесение изменений в продукционную ИТ-инфраструктуру

Традиционная сеть

- Длительное документальное согласование схем коммутации
- «Ручная» конфигурация сети, потоков, сетевых устройств на основе документально согласованных схем

Программно-конфигурируемая сеть

- Наличие предварительно разработанных и отлаженных приложений управления сетью передачи данных
- Автоматизированная конфигурация сети контроллером на основе пользовательских запросов при помощи указанных приложений управления сетью в соответствии с принятыми политиками безопасности
- Применение методов машинного обучения при управлении конфигурацией сети

Конфликты конфигурации

Традиционная сеть

- Разрешаются «вручную» на этапе документального согласования схем коммутации и опытно-промышленной эксплуатации внедряемого сервиса
- Изменения конфигурации единичные и всегда авторизованы ответственным сотрудником

Программно-конфигурируемая сеть

- Будучи замеченными, требуют внесения изменений в приложения, управляющими сетью, конфигурационные данные приложений
- Изменения конфигурации вызваны частыми конкурирующими запросами пользователей или сетевых сервисов. Приходится полагаться на корректность работы приложения

Методы минимизации рисков возникновения конфликтов

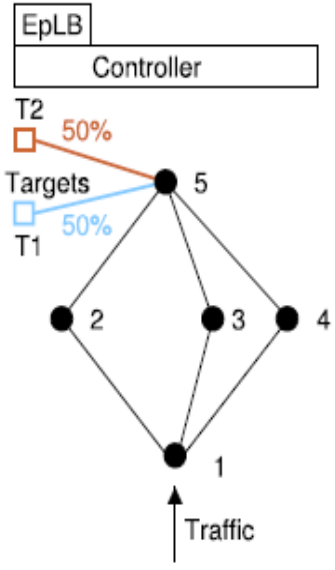
Традиционная сеть

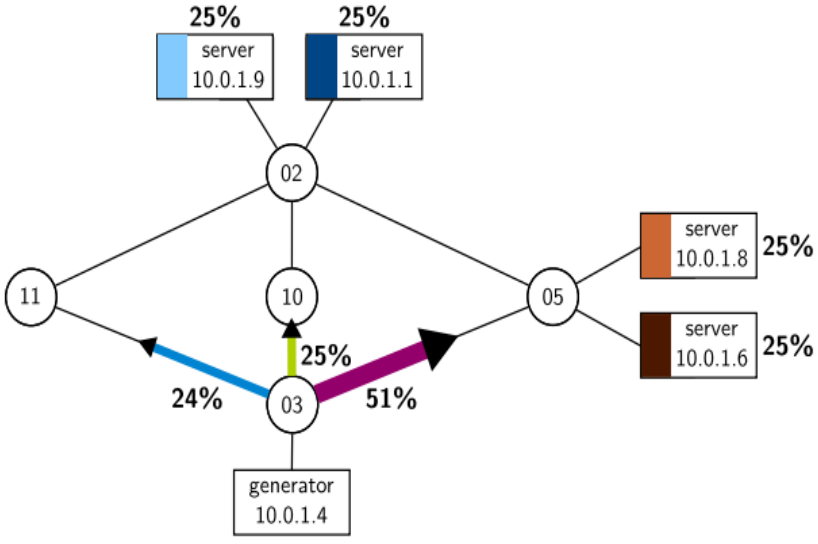
- Экспертный анализ, внесение изменений в документально согласованные схемы, повторное согласование документов
- Внесение «вручную» изменений в конфигурацию
- Последующий экспертный контроль эксплуатации сети

Программно-конфигурируемая сеть

- Теоретическое обоснование методов и алгоритмов, по которым работают названные приложения.
- Внесение изменений в приложения
- Экспертный контроль эксплуатации сети
- Применение методов машинного обучения при управлении конфигурацией сети

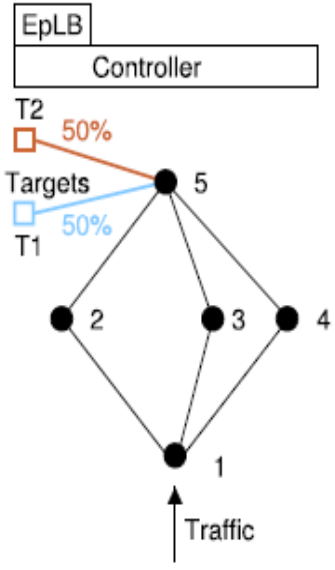
Что такое конфликты?

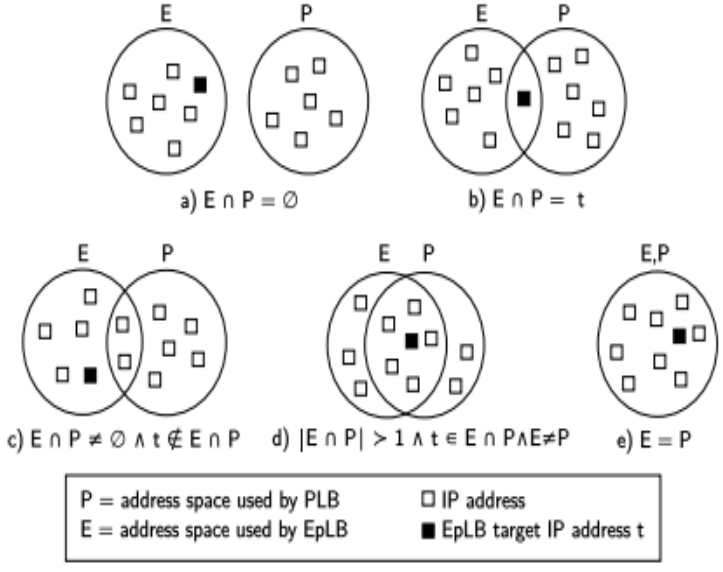




Kletzander R. A testbed for researching conflicts in SDN, 2017. Bachelor's thesis.

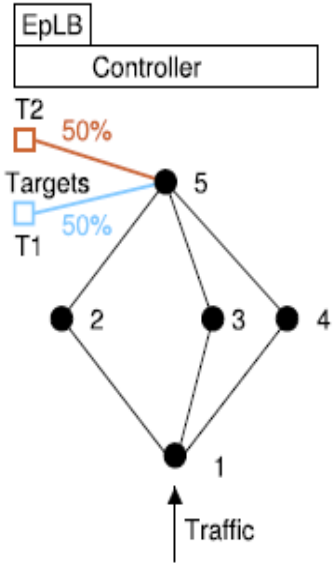
Что такое конфликты?

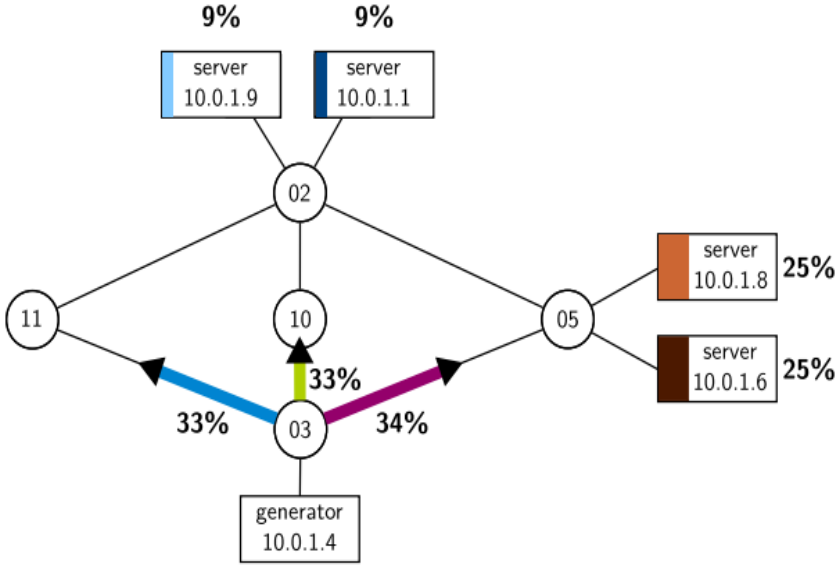




Kletzander R. A testbed for researching conflicts in SDN, 2017. Bachelor's thesis.

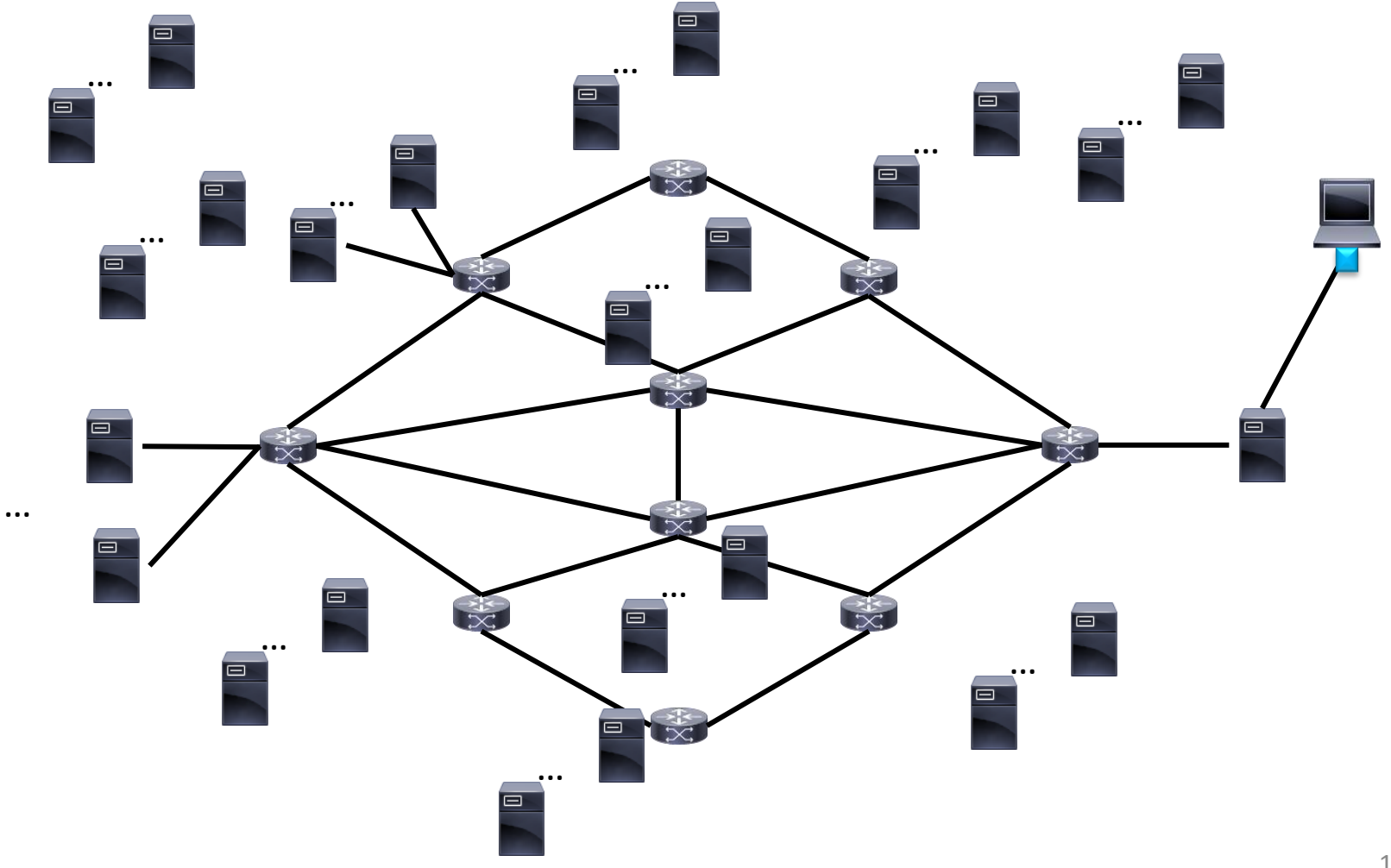
Что такое конфликты?





Kletzander R. A testbed for researching conflicts in SDN, 2017. Bachelor's thesis.

Суть проблемы



Суть проблемы

Решения для верификации команд реконфигурации

- VERMONT
- NetPlumber
- VeriFlow
- AP Verifier
- FlowChecker
- Anteater

Рабочие материалы по мероприятию 04 «Разработка синтаксической и понятийной основы языка спецификации политик Forwarding в ПКС сетях». Некоммерческое партнерство «Центр прикладных исследований компьютерных сетей, Москва 2013,

Вермонт

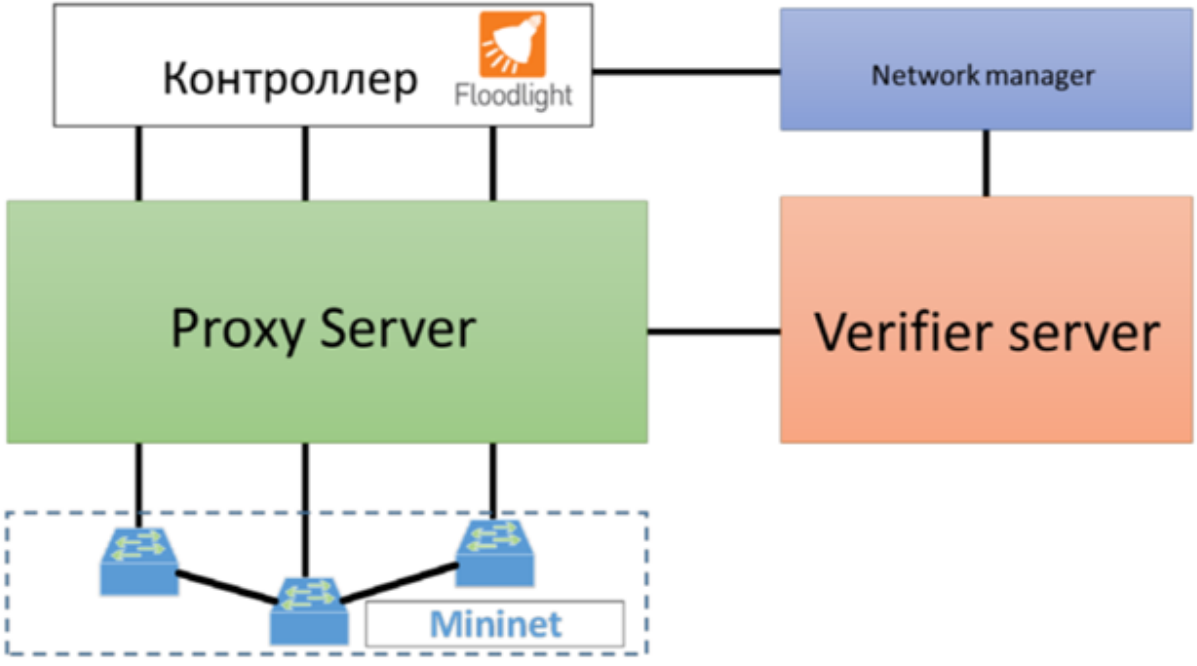


Рис. 1

Вермонт

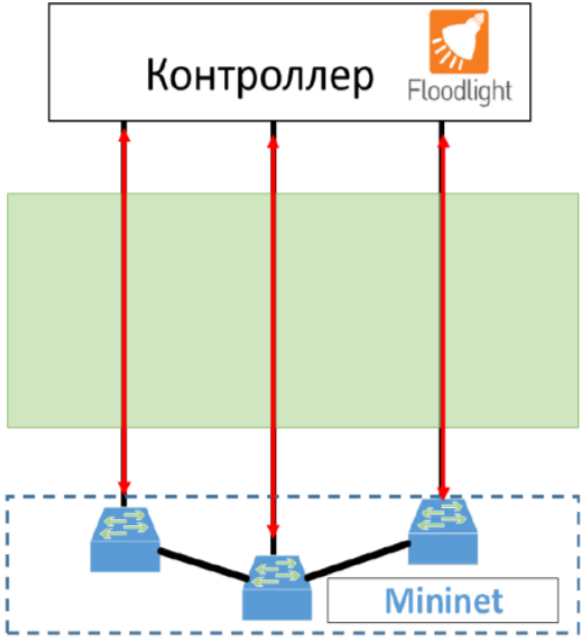


Рис. 2. Работа Proxy Server в режиме SEAMLESS

Вермонт

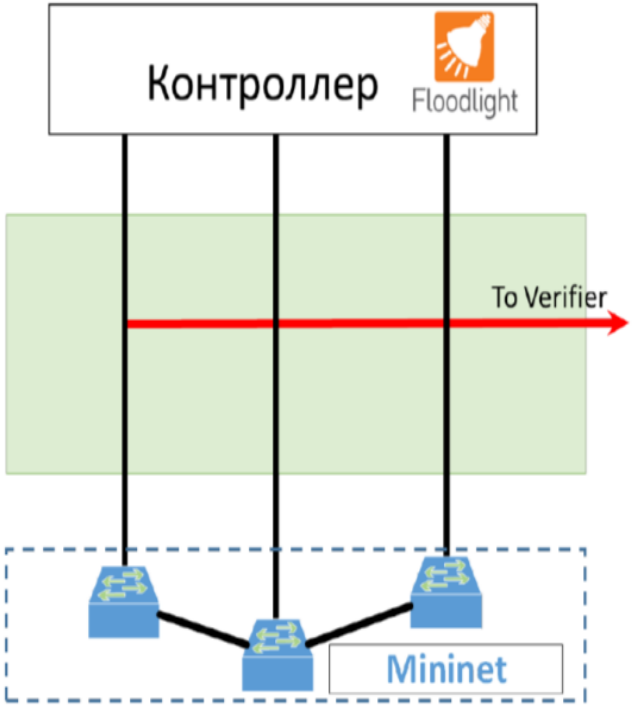


Рис. 3. Работа Proxy Server в режиме MIRROR

Рабочие материалы по мероприятию 04 «Разработка синтаксической и понятийной основы языка спецификации политик Forwarding в ПКС сетях». Некоммерческое партнерство «Центр прикладных исследований компьютерных сетей, Москва 2013,

Вермонт

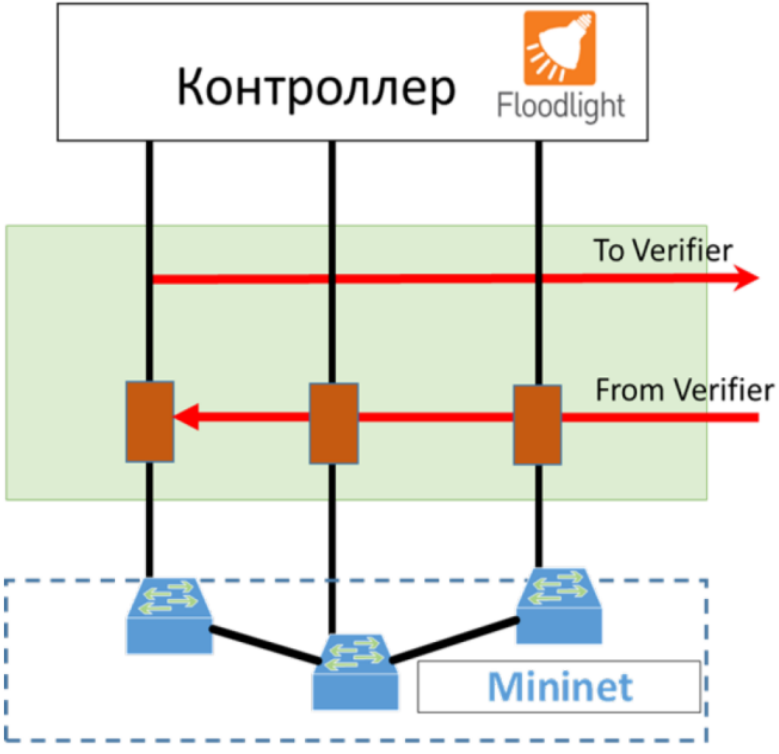
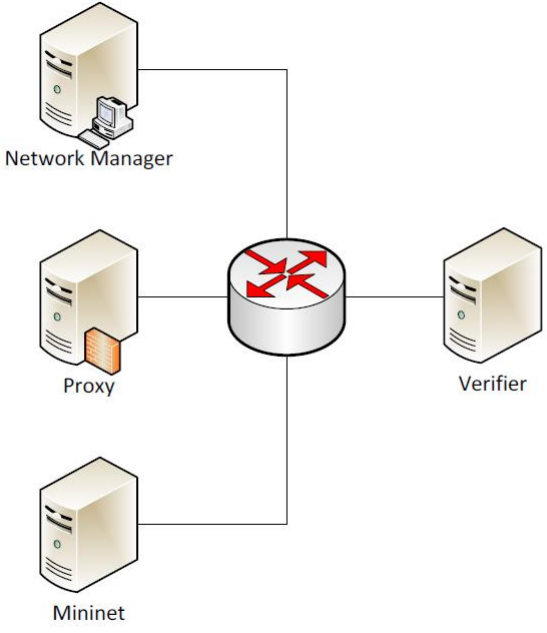


Рис. 4. Работа Proxy Server в режиме INTERRUPT.

Вермонт


Команды
Программно конфигурируемая сеть
Сообщения
Прокси-сервер
Команды
Сообщения
Контроллер
Вердикт верификации
Команды
Запросы
Таблицы коммутации и топологии
Редактор
Спецификации ПМП
Верификатор
Данные о сети
Запросы
Модуль загрузки

Вермонт



Процедура реконфигурации

$P^i = ((O_1^i, e_1^i), \dots, (O_n^i, e_n^i))$, где O_k^i – операция;

e_k^i – объекты реконфигурации k – й операции в i – й процедуре.

(O_j^i, e_j^i) - Выполнение некой операции O над объектом e на шаге j в процедуре i.

А. А. Грушо, М. И. Забежайло, А. А. Зацаринный, В. О. Писковский, “Безопасная автоматическая реконфигурация облачных вычислительных сред”, Системы и средства информ., 26:3 (2016), 83–92

Определение согласованности

$P = ((O_1, e_1), \dots, (O_n, e_n))$ – процедура реконфигурации

$(\mathrm{O_j}, \mathrm{e_j})$ - Выполнение некой операции O над объектом e на шаге j.

Введем функцию time $(\mathrm{O_j}, \mathrm{e_j}) = (\mathrm{t^j_{st}}, \mathrm{t}^{j}_{en})$, где $\mathrm{t^j_{st}}$ – время начала, а t^{j}_{en} – время завершения процедуры реконфигурации j.

Объекты $(\mathrm{O_j}, \mathrm{e_j})$и $(\mathrm{O_k}, \mathrm{e_k})$ являются ε согласованными, если :

$$\left(\mathrm{e_j} \cap \mathrm{e_k} \right) \text{ and } \left(\left(\text{time} \left(\mathrm{O_j}, \mathrm{e_j} \right) \cap \text{time}(\mathrm{O_k}, \mathrm{e_k}) \right) < \varepsilon \right) = \emptyset$$

В случае если ε=0,то это полная согласованность,никаких рисков нету.При росте
ε возрастает вероятность появления конфликта

Определение синхронизации

$P = ((O_1, e_1), \dots, (O_n, e_n))$ – процедура реконфигураци

(O_j, e_j) - Выполнение операции O над объектом e на шаге j

Процедура реконфигурации называется синхронизированной, если для любых двух шагов, порядок выполнения которых не задан, выполняется:

$P = (\dots, (O_j, e_j), \dots, (O_k, e_k), \dots) =$
$(\dots, (O_k, e_k), \dots, (O_j, e_j), \dots) = P'$

Формальное определение синхронизации

$P = ((O_1, e_1), \dots, (O_n, e_n))$– процедура реконфигураци
(O_j, e_j) - Выполнение некой операции O над объектом e на шаге j

Введем функцию time $(O_j, e_j) = (t_1, t_2)$, где t1 – время начала, а t2 – время завершения процедуры реконфигурации.

Объекты (O_j, e_j) и (O_k, e_k) являются синхронизированными, если :

$$(e_j \cap e_k) \text{ or } (\text{time } (O_j, e_j) \cap \text{time}(O_k, e_k)) = \emptyset$$

- Процедура $P = ((O_1, e_1), \ldots, (O_n, e_n))$ называется правильной, если ∀ j ∈ 1..n выполняются следующие условия:
- 1) Если $O_j =$ lock, объект e_j не был заблокирован на предыдущих j-1 шагах;
- 2) Если $O_j \neq$ lock, то объект e_j был заблокирован P до j-1 шага включительно;
- 3) Если в процедуре n операций, то $O_n = unlock$, а единственным заблокированным объектом является e_n

- Расписание реконфигурации S – последовательность операций из множества процедур P

Операция	Объект реконфигурации
Lock	A
A+=100	A
Lock	B
Unlock	A
B+=100	B
Unlock	B

- Процедура $P^i = (\left(O_1^i, e_1^i\right), \ldots, \left(O_n^i, e_n^i\right))$ называется двухфазной, если для любой j<n выполняются следующие условия:
- 1) если $i < j$, то $O_i \neq$ unlock;
- 2) если $i = j$, то $O_i =$unlock;
- 3) если $i > j$, то $O_i \neq$ lock;

Операция	Объект реконфигурации
Lock	A
A+=100	A
Unlock	A
Lock	B
B+=100	B
Unlock	B

- Процедура $P^i = (\left(O_1^i, e_1^i\right), \ldots, \left(O_n^i, e_n^i\right))$ называется двухфазной, если для любой j<n выполняются следующие условия:
- 1) если $i < j$, то $O_i \neq$ unlock;
- 2) если $i = j$, то $O_i =$unlock;
- 3) если $i > j$, то $O_i \neq$ lock;

Операция	Объект реконфигурации
Lock	A
A+=100	A
Lock	B
Unlock	A
B+=100	B
Unlock	B

Достаточные условия непротиворечивости процедур реконфигурации

- Если все процедуры $P^i, i \in 1, k$ являются правильными и двухфазными, то любое допустимое расписание синхронизировано
- Если процедура P не является правильной и двухфазной, то существует правильная и двухфазная процедура P' такая, что имеют допустимое, но не синхронизированное расписание

Eswaran P., Gray J., Lorie R., Traiger I. The notions of consistency and predicate locks in a database system // Commun. ACM, 1976. Vol. 19. No. 11. P. 624-633.

Аномалии

СУБД		ОВС
Аномалия	**Описание**	**Описание**
1. P0 Dirty Write (непредсказуемый результат записи)	Транзакция изменяет данные. Сторонняя транзакция также меняет эти же данные до «ввода в действие» или «отмены изменений» данных первой транзакцией. Если какая-либо из указанных транзакций отказывается от изменений и возвращает корректные с ее точки зрения данные, то неясно, какие данные считать корректными	Конкурентно не менее двух процедур переконфигурации вносят противоречащие друг другу изменения в один и тот же сетевой объект *x* (это могут быть противоречащие друг другу значения полей или строки таблиц маршрутизации) с отложенным вводом в действие измененных правил коммутации. Если оказывается, что какая-либо из указанных процедур переконфигурации отказывается от изменений и возвращает корректные с ее точки зрения данные полей или содержимое таблиц коммутации, то не определено, останется ли корректным состояние объекта переконфигурации и конфигурация в целом коммутатора, содержащего эти объекты

Berenson H., Bernstein Ph., Gray J., Melton J., O'Nei E., O'Neil P. A Critique of ANSI SQL Isolation Levels // Proceedings of the 1995 ACM SIGMOD international conference on Management of data. – New York: ACM, 1995. P. 1–10.

Аномалии

СУБД [10]		ОВС
Аномалия	**Описание**	**Описание**
2. P1 Dirty Read (непредсказуемый результат чтения)	Транзакция читает данные, которые записаны другой транзакцией, но не «введены в действие»	Контроллер(ы) опрашивает содержимое таблиц коммутации и учитывает при переконфигурации данные, которые не приобрели или утеряли актуальность по вине стороннего процесса переконфигурации (происходящего в интересах достижения других целей)
3. P4 Lost Update (потерянные данные)	Транзакция читает данные. В это время сторонняя транзакция изменяет эти данные (возможно, с использованием ранее прочитанных значений). Затем первая транзакция (на основе ранее прочитанных значений) меняет данные и «вводит их в действие». Теперь после ввода действия данных второй транзакцией, данные с точки зрения второй транзакции утеряны	Контроллер(ы) опрашивает содержимое таблиц коммутации и учитывает при переконфигурации данные, которые на момент выполнения самой целевой переконфигурации уже подверглись изменению другими переконфигурационными процессами. В результате получаем некорректно настроенный коммутатор

Аномалии

СУБД		ОВС
Аномалия	**Описание**	**Описание**
4. P2 Fuzzy Read (Nonrepeatable) (изменение набора ранее прочитанных данных)	Транзакция еще раз пытается прочесть ранее прочитанные данные и находит, что сторонняя транзакция изменила или удалила эти данные	Контроллеры вносят изменения в объект управления. При этом оказывается, что изменения используют данные, которые изменены или удалены конкурирующим процессом переконфигурации
5. P3 Phantom (фантом)	Транзакция еще раз выполняет запрос, который возвращает набор строк, удовлетворяющих условиям отбора, и выясняет, что другая транзакция добавила строки, также удовлетворяющие условиям отбора. Например, транзакция запрашивает количество сотрудников. Через пять минут этот же запрос возвращает большее значение, так как другой пользователь за это время добавил сотрудника, подпадающего под критерии выбора в запросе. В отличие от P2, ранее полученные данные не изменились	Контроллеры вносят изменения в объект управления. При этом оказывается, что конкурирующим процессом в таблицы маршрутизации внесены изменения, делающие внесенные изменения недействительными

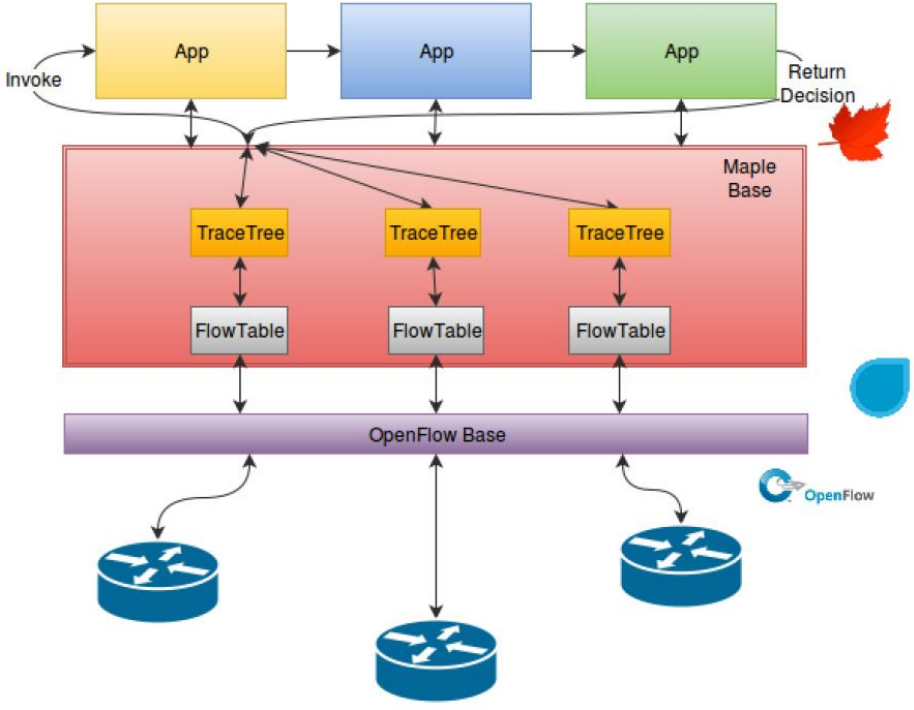
Аномалии

СУБД		ОВС
Аномалия	**Описание**	**Описание**
6. A5A Read Skew (асимметричное чтение)	Пусть данные X и Y находятся в некотором соотношении. Транзакция T_1 читает X. Сторонняя транзакция T_2 меняет X и Y и вводит их в действие. Если сейчас T_1 прочитает Y, то это значение не будет соответствовать ранее прочитанному X и транзакция T_1 вернет нарушение целостности	Объекту переконфигурации X (поле, строка, таблица коммутации, содержимое таблиц коммутатора) соответствует объект переконфигурации Y. Процесс переконфигурации/верификации запрашивает данные объекта X. Конкурирующий процесс переконфигурации меняет X и Y. Если первый процесс запросит содержимое объекта Y, то он найдет его несоответствие ранее прочитанному

Аномалии

СУБД		ОВС
Аномалия	**Описание**	**Описание**
7. A5B Write Skew (асимметричная запись)	Пусть данные X и Y находятся в некотором соотношении. Транзакция T_1 читает X и Y и находит их корректными. Сторонняя транзакция T_2 читает X и Y, меняет X и вводит данные в действие. Затем T_1 меняет Y. Таким образом, соотношение между X и Y может быть нарушено	Объекту переконфигурации X (поле, строка, таблица коммутации, содержимое таблиц коммутатора) соответствует объект конфигурации Y. Процесс переконфигурации/верификации запрашивает данные объектов X и Y и находит их корректными. Конкурирующий процесс переконфигурации меняет X. Если первый процесс изменит содержимое объекта Y, то соответствие X и Y будет нарушено

Архитектура Maple



fNetKAT

- Maple: simplifying SDN programming using algorithmic policies / A. Voellmy, J. Wang, Y. R. Yang, B. Ford, P. Hudak // ACM SIGCOMM Computer Communication Review. 2013. т. 43, □стр. 87-98.
- NetKAT: Semantic foundations for networks / C. J. Anderson, N. Foster, A. Guha, J.-B. Jeannin, D. Kozen, C. Schlesinger, D. Walker // Acm sigplan notices. 2014. т. 49, □стр.. 113-126.
- Frenetic: A network programming language / N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, D. Walker // ACM Sigplan Notices. 2011. т. 46, № □9. стр.. 279-291.