RIFT:
ROUTING IN FAT TREES

Обзор протокола

Александр Беспалов (abespalov@juniper.net)

JUNIPER NETWORKS | Engineering Simplicity

RIFT

Новый протокол маршрутизации для ЦОД

Тенденции в современных ЦОД:

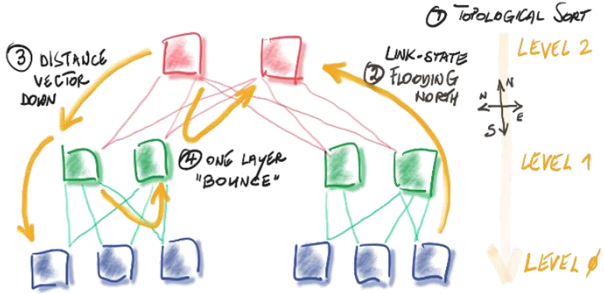
- Широкое использование топологии Clos и Fat Tree
- Переход от Layer 2 коммутации к Layer 3 маршрутизации с Layer 2/3 Overlay
- IGP или BGP (RFC7938) для Underlay маршрутизации

Сложности при использовании BGP или IGP:

- В случае с IGP отказ даже одного линка приводит к значительному флудингу маршрутной информации и пересчету SPF алгоритма на всех узлах
- Сложность конфигурации в случае с BGP
- Избыточность маршрутной информации (в идеале Leaf узлы должны иметь только маршрут по умолчанию)

Что такое RIFT (Routing in Fat Trees):

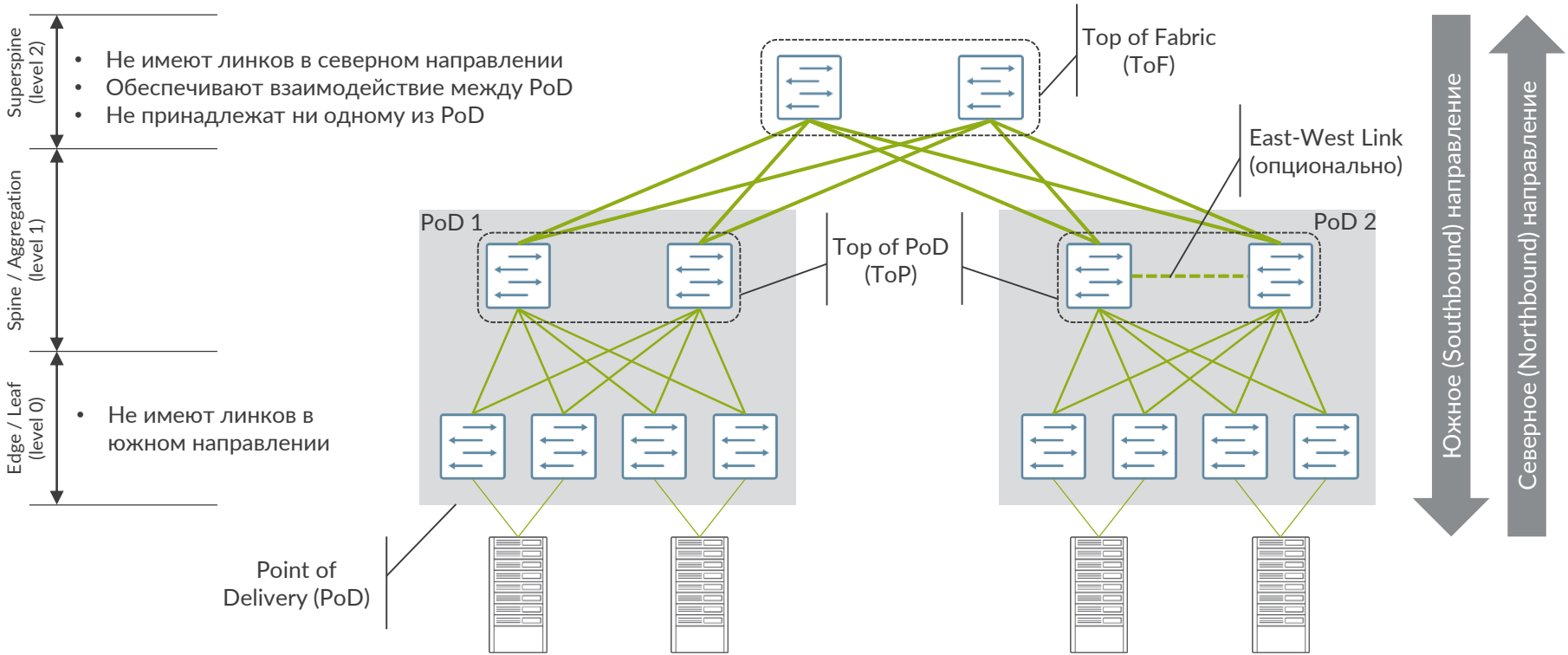
- Новый протокол маршрутизации для ЦОД (для Clos топологий)
- https://tools.ietf.org/html/draft-ietf-rift-rift-03
- Совмещает особенности Link-State протоколов (в “северном” направлении) и Distance Vector (в “южном” направлении)
- В штатной ситуации нижележащему уровню анонсируется исключительно маршрут по умолчанию



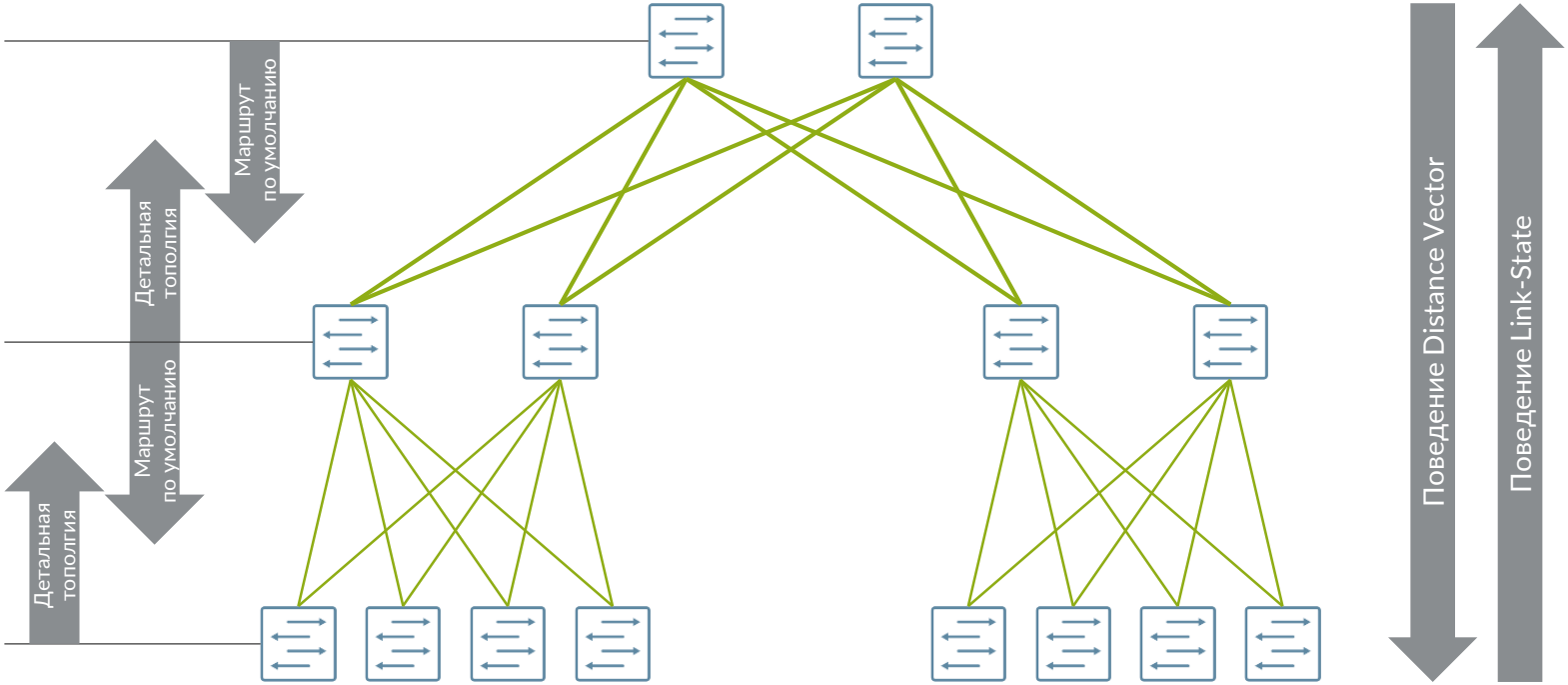
ОСОБЕННОСТИ RIFT

- Автоматическое построение топологии (с использованием ZTP)
- Предотвращение построения ошибочных соседств (при ошибочной коммутации)
- Минимизация состояний и маршрутной информации
- Автоматическая дезагрегация маршрутов для предотвращения потерь трафика и неоптимальной маршрутизации
- Неэквивалентная балансировка
- Автоматическая перебалансировка трафика в зависимости от доступной полосы пропускания на уровне выше

ТЕРМИНОЛОГИЯ



ПРИНЦИП РАБОТЫ



ПЕРЕДАЧА СООБЩЕНИЙ

Одним из требований к протоколу является возможность работы с Unnumbered адресами (минимизация сетевых настроек)

Транспорт:

- UDP
- Возможно в будущем QUIC (https://tools.ietf.org/html/draft-ietf-quic-transport-15)

Кодирование данных:

- Thrift (https://thrift.apache.org/)
- Возможно в будущем Protocol Buffers (https://developers.google.com/protocol-buffers/)

Neighbor Discovery: Link Information Elements (LIE) Exchange

- Аналог Hello в IGP
- IPv4: 224.0.0.120 (может быть изменено в конфигурации)
- IPv6: FF02::A1F7 (может быть изменено в конфигурации)
- UDP порт назначения: 911
- TTL: 1

Topology Exchange: Topology Information Elements (TIE) Exchange

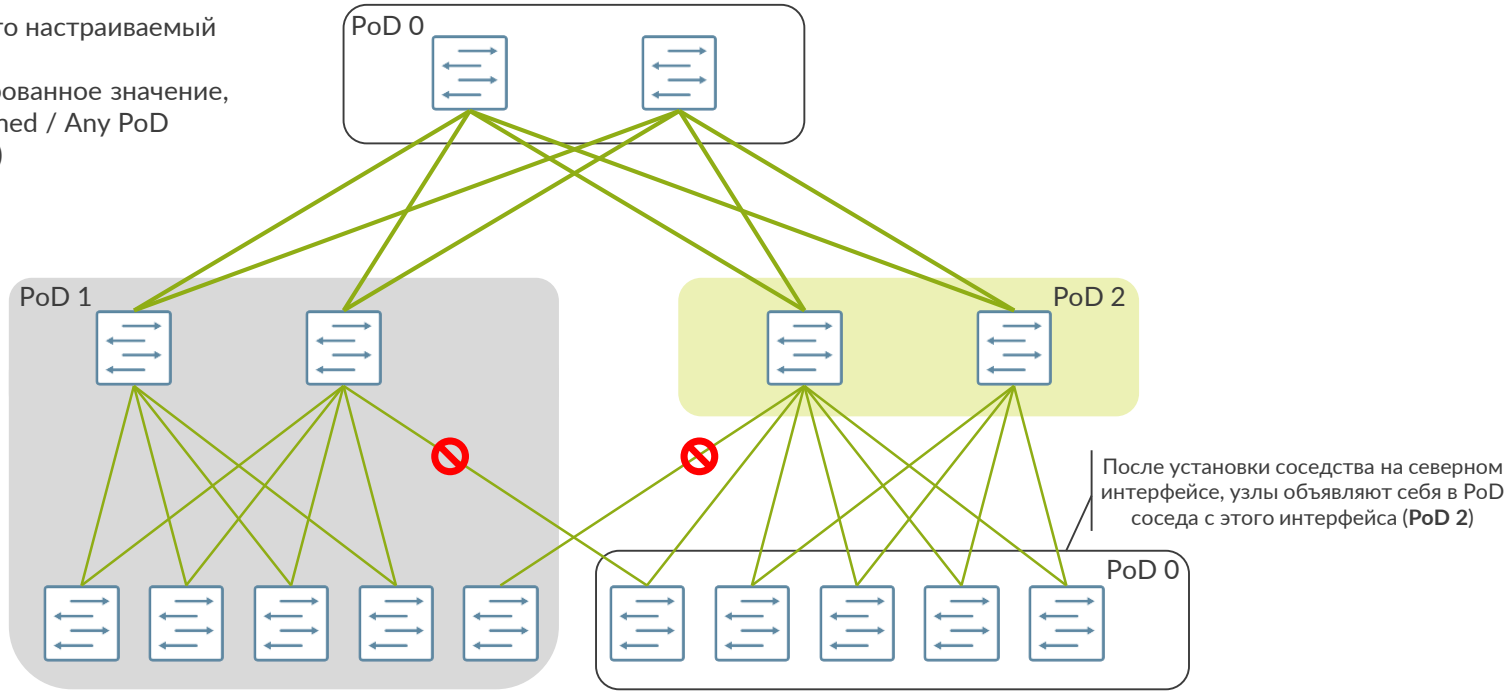
- Аналог LSA в OSPF
- Адрес назначения: любой адрес источника LIE от соседа
- UDP порт назначения: сообщается соседом в LIE

УСТАНОВКА СОСЕДСТВА

Встроенная защита от некорректных топологий

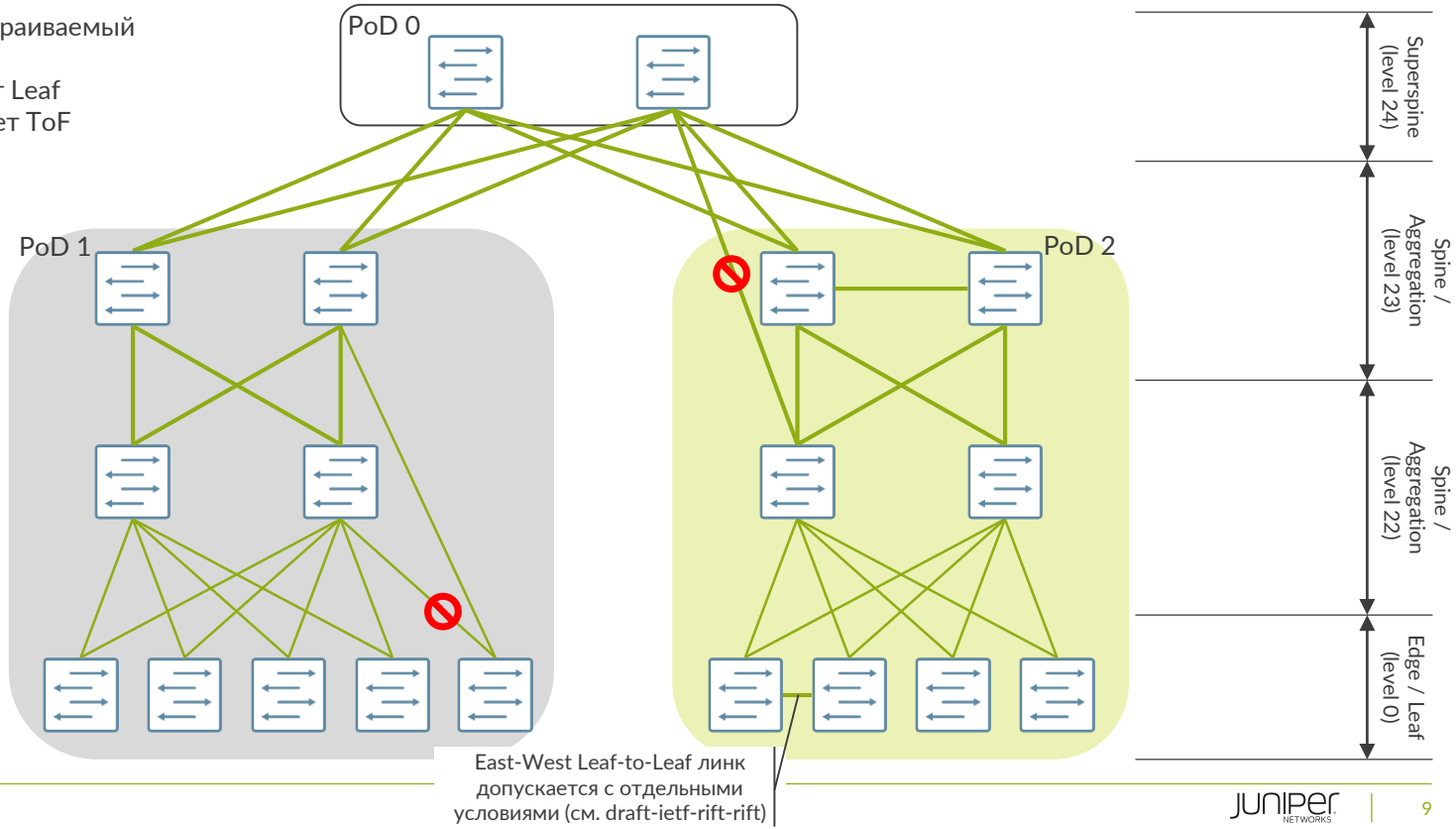
ПРОВЕРКА ПРИНАДЛЕЖНОСТИ К POD

- Идентификатор PoD – это настраиваемый параметр
- **PoD = 0** – это зарезервированное значение, которое означает Undefined / Any PoD (значение по умолчанию)



ПРОВЕРКА ВЗАИМОДЕЙСТВИЯ МЕЖДУ УРОВНЯМИ

- Уровень (level) – это настраиваемый параметр
- **Level = 0** – соответствует Leaf
- **Level = 24** – соответствует ToF



ДОПОЛНИТЕЛЬНЫЕ ПАРАМЕТРЫ

Для успешного установления соседства дополнительно проверяются следующие параметры:

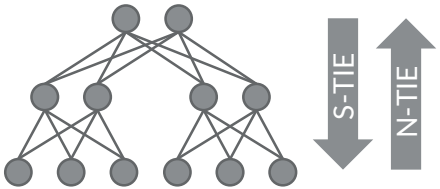
- Совпадение MAJOR версии протокола
- Валидность System ID
- System ID соседа не совпадает с локальным System ID
- Совпадение MTU с двух сторон линка
- TTL = 1

ОБМЕН МАРШРУТНОЙ ИНФОРМАЦИЕЙ

ОРГАНИЗАЦИЯ МАРШРУТНОЙ БАЗЫ ДАННЫХ

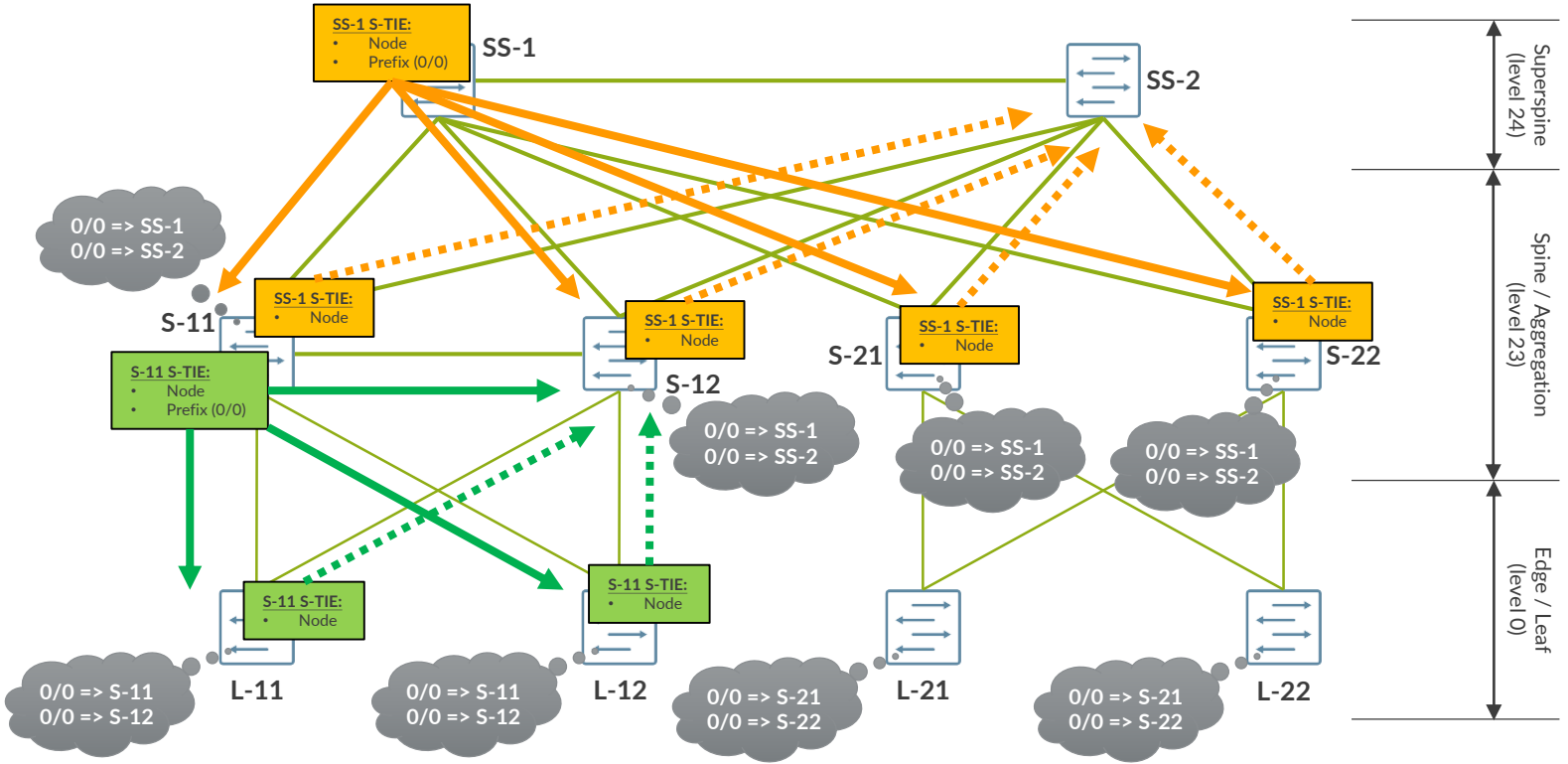
Каждый узел оперирует двумя базами данных:

- Северная (N-TIE) – отсылается верхнему уровню:
  - Node N-TIE – информация о соседях и линках
  - Prefix N-TIE – **детальная** информация об IP префиксах
- Южная (S-TIE) – отсылается нижнему уровню
  - Node S-TIE – информация о соседях и линках
  - Prefix S-TIE – **маршрут по умолчанию (0/0)** или дезагрегированные префиксы
  - В южную сторону распространяются только собственные S-TIE

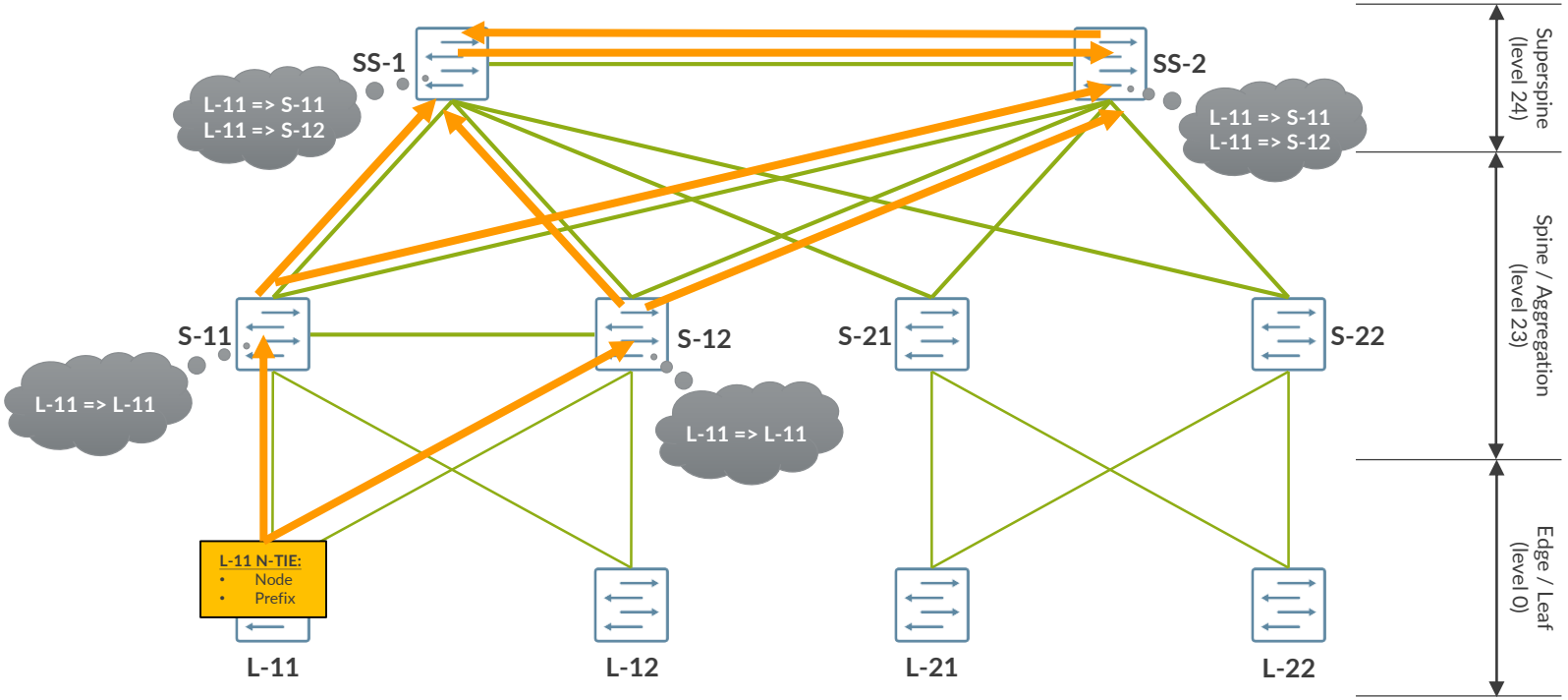


Существуют и другие типы TIE (см. draft-ietf-rift-rift)

РАСПРОСТРАНЕНИЕ ИНФОРМАЦИИ В ЮЖНОМ НАПРАВЛЕНИИ

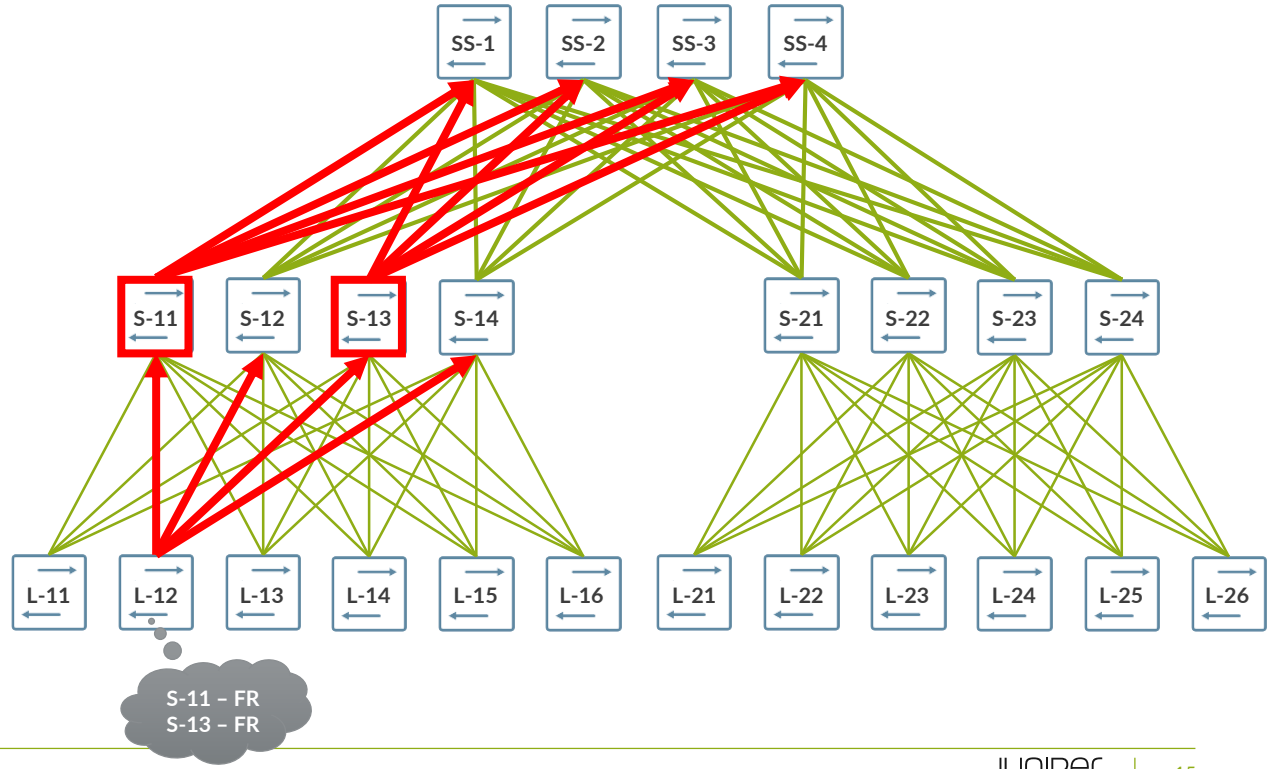


РАСПРОСТРАНЕНИЕ ИНФОРМАЦИИ В СЕВЕРНОМ НАПРАВЛЕНИИ



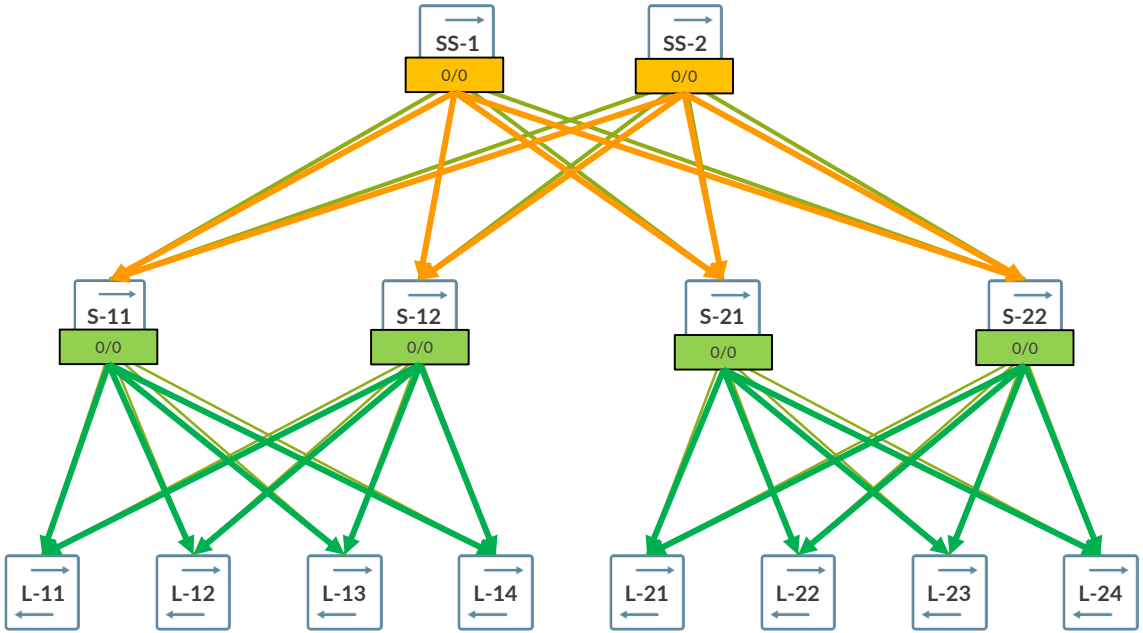
ОПТИМИЗАЦИЯ ФЛУДИНГА В СЕВЕРНОМ НАПРАВЛЕНИИ

- За пересылку (флудинг) сообщений в северном направлении отвечает Flood Repeater (FR)
- Каждый узел выбирает пару FR на уровне выше (для резервирования)
- Выбор FR локален для каждого узла
- О том, что узел выбрали в качестве FR, нижележащий узел сообщает вышестоящему узлу в LIE пакетах
- Изменение топологии на верхних уровнях отслеживается через S-TIE пакеты
- При изменении топологии (потеря соседств на верхних уровнях) происходит перевыбор FR
- Алгоритм описан в draft-ietf-rift-rift



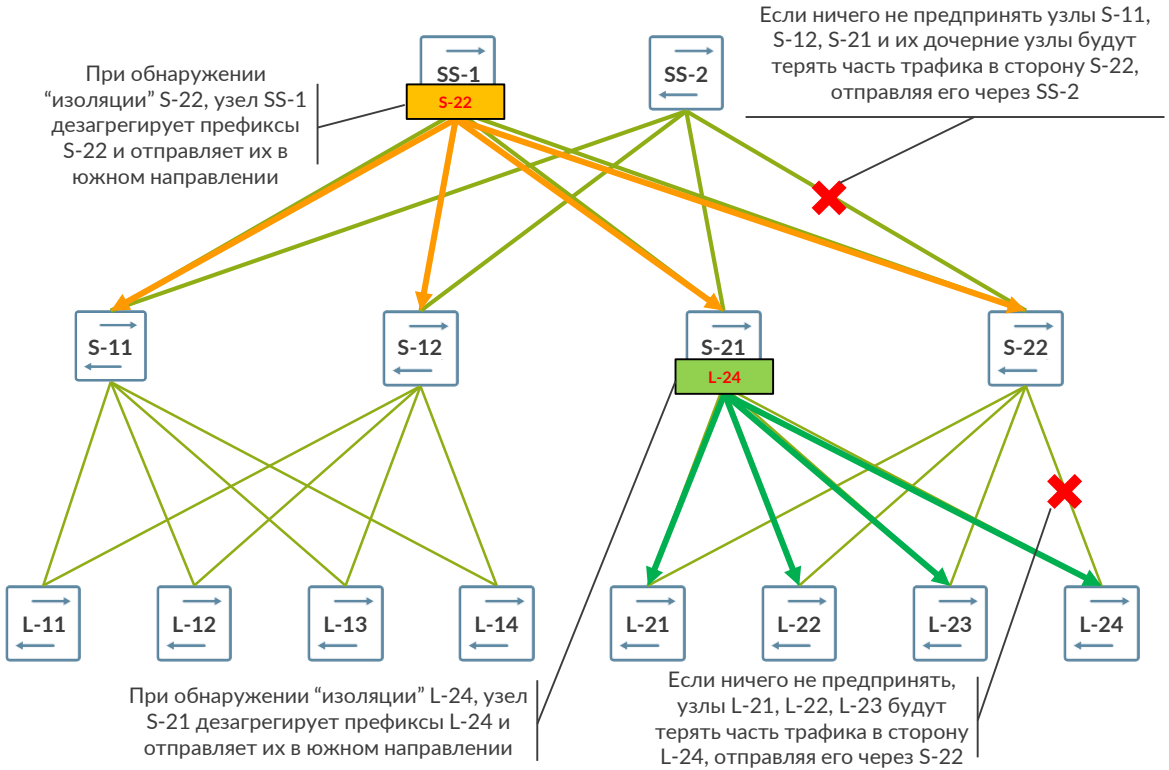
ДЕЗАГРЕГАЦИЯ МАРШРУТОВ ПРИ АВАРИЯХ

РАСПРОСТРАНЕНИЕ МАРШРУТОВ В ЮЖНОМ НАПРАВЛЕНИИ (ПОВТОРЕНИЕ)



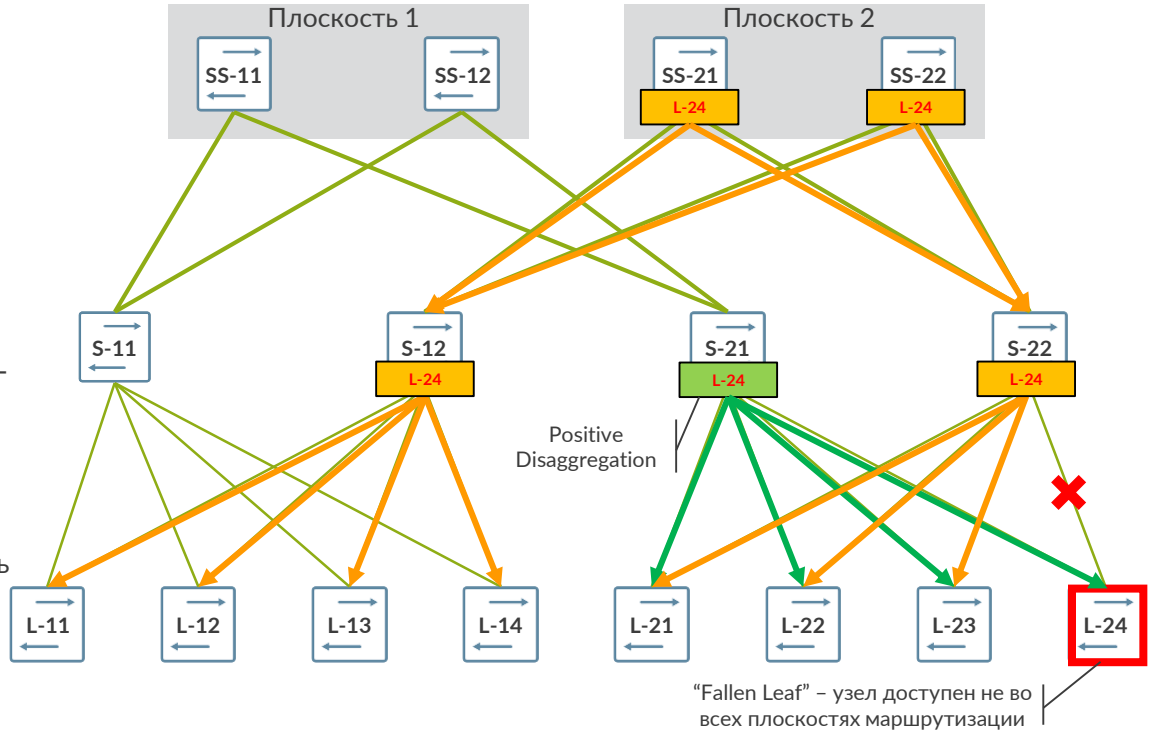
POSITIVE (NON-TRANSITIVE) DISAGGREGATION

- Процесс дезагрегации позволяет избежать Blackholing трафика при авариях
- Это автоматический процесс
- Для детектирования “изоляций” используются S-TIE, которые “отражаются” нижним уровнем верхнему
- Дезагрегированные префиксы передаются в S-TIE, поэтому распространяются только на один уровень вниз



NEGATIVE (TRANSITIVE) DISAGGREGATION

- В некоторых топологиях Positive Disaggregation не исключает возникновения Blackholing (например, дизайн с несколькими плоскостями маршрутизации)
- “Negative Disaggregation” – означает “не использовать путь для достижения определенных префиксов”
- Это автоматический процесс
- ToF узлы детектируют появление “Fallen Leaf” из N-TIE
- Дезагрегированные префиксы передаются в S-TIE
- Negative Disaggregated префиксы могут распространятся на несколько уровней вниз (вплоть до Leaf), если:
  - префикс не объявляется ни одним дочерним (южным) узлом **и**
  - все родительские (северные) узлы объявляют этот префикс как Negative Disaggregation



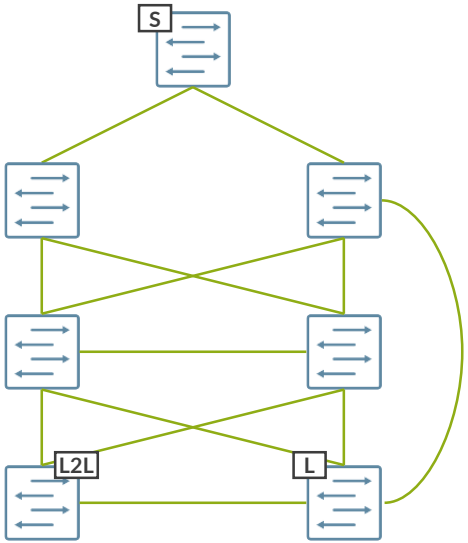
ZERO TOUCH PROVISIONING (ZTP)

ZERO TOUCH PROVISIONING (ZTP)

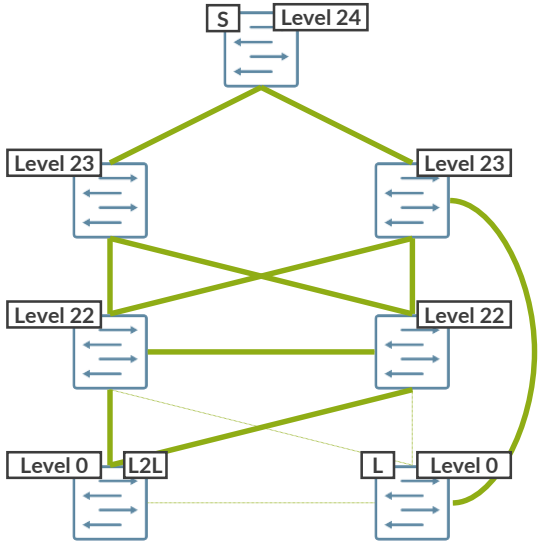
- Опциональный режим
- Узел определяет свои настройки после подключения сети
- Может быть запущен на любом узле кроме:
  - Top-of-Fabric (ToF)
  - Leaf, на которых используются Leaf-to-Leaf линки
- Топология (сеть) может может совмещать узлы с ZTP и без ZTP
- Информация для ZTP распространяется в LIE (Hello)
- Дополнительные конфигурационные флаги (задаются вручную):
  - **TOP_OF_FABRIC** – должен присутствовать на всех ToF узлах (сконфигурированный TOP_OF_FABRIC флаг автоматически означает, что узел располагается на level 24)
  - **LEAF_ONLY** – позволяет узлу работать только в режиме Leaf (быть на нижнем уровне IP фабрики – level 0)
  - **LEAF_2_LEAF** – задается на узле, если он должен поддерживать линки Leaf-to-Leaf

ФОРМИРОВАНИЕ ТОПОЛОГИИ С ZTP (ПРИМЕР №1)

Исходная топология:



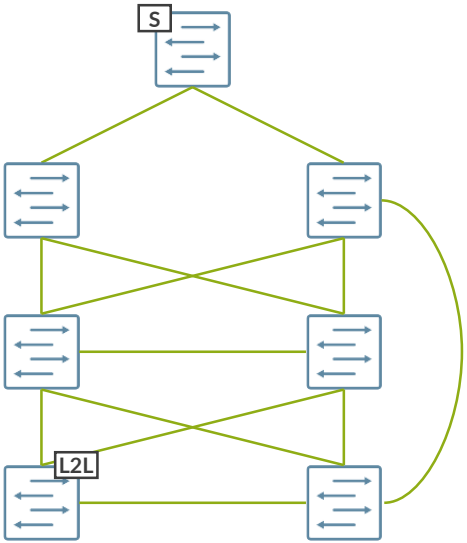
Топология RIFT:



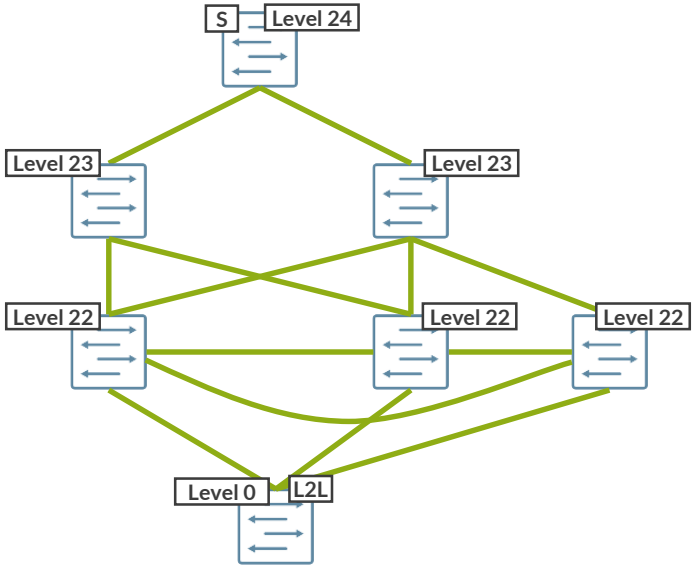
S – TOP_OF_FABRIC флаг (всегда level 24)
L – LEAF_ONLY флаг (всегда level 0)
L2L – LEAF_TO_LEAF флаг

ФОРМИРОВАНИЕ ТОПОЛОГИИ С ZTP (ПРИМЕР №2)

Исходная топология:



Топология RIFT:



S – TOP_OF_FABRIC флаг (всегда level 24)
L – LEAF_ONLY флаг (всегда level 0)
L2L – LEAF_TO_LEAF флаг

FABRIC BANDWIDTH BALANCING

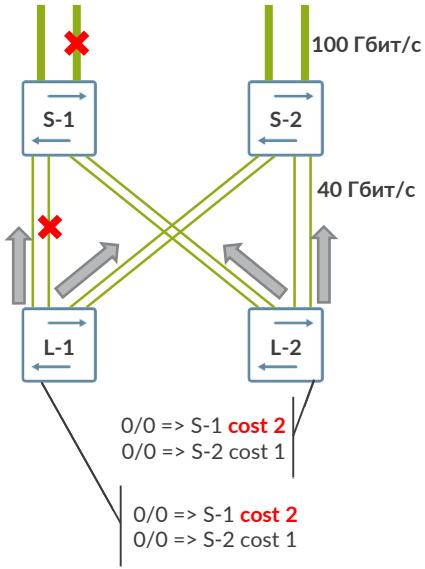
НЕЭКВИВАЛЕНТНАЯ БАЛАНСИРОВКА

Проблема

- В топологиях с параллельными линками желательно иметь механизм перебалансировки трафика во время отказов отдельных линков
- Наиболее критична данная проблема для трафика в “северном” направлении, т.к. нижележащие узлы имеют только маршрут по умолчанию (не знают детальной топологии выше)

Решение

- RIFT позволяет производить неэквивалентную балансировку в зависимости от доступной полосы пропускания на маршрутизаторе верхнего уровня (используется информация из S-TIE, полученных от выше стоящего узла)
- Детальный алгоритм расчета метрик приведен в draft-ietf-rift-rift
- В результате, метрика маршрутов по умолчанию принятых сверху меняется на взвешенную метрику
- Расчет затрагивает максимум два уровня топологии и не зависит от размера топологии (сети)
- Неэквивалентная балансировка может использоваться и в “южном” направлении (это проще, т.к. выше стоящие узлы обладают детальной топологией нижележащих уровней)



ЗАКЛЮЧЕНИЕ

ТЕКУЩЕЕ СОСТОЯНИЕ

- Текущее описание протокола в статусе IETF драфта:
  - https://datatracker.ietf.org/doc/draft-ietf-rift-rift/
- Основным автором драфта является Juniper
- В разработке протокола кроме Juniper участвует несколько крупных компаний (операторы и производители оборудования)
- Для изучения эмулятор работы протокола можно получить на нашем сайте:
  - https://www.juniper.net/us/en/dm/free-rift-trial/
- Доступна демо версия пакета RIFT для Junos (для предварительного ознакомления)
- Реализация на оборудовании ожидается в 2019 году

ПРИМЕРЫ НАСТРОЕК

Настройка Superspine:

```
protocols {
    rift {
        level top-of-fabric;
        name superspine-1;
        interface xe-0/0/0.0;
        interface xe-0/0/1.0;
    }
}
```

- Настройка уровня “Top-of-Fabric” (level 24)
- Добавление интерфейсов

Настройка Spine:

```
protocols {
    rift {
        name spine-1-1;
        interface xe-0/0/0.0;
        interface xe-0/0/1.0;
        interface xe-0/0/2.0;
        interface xe-0/0/3.0;
    }
}
```

- Добавление интерфейсов
- Уровень определяется автоматически (по ZTP)

Настройка Leaf:

```
protocols {
    rift {
        level leaf;
        name leaf-1-1
        interface xe-0/0/0.0;
        interface xe-0/0/1.0;
    }
}
```

- Настройка уровня “Leaf ” (level 0)
- Добавление интерфейсов

СПАСИБО!