Подготовка PNETLab

Образы и ресурсы

Узлы	Образ	Ресурсы на один узел
SPINE1/2, LEAF1/2	`nxosv9k-9300v-10.5.2.F`	4 vCPU, 12288 МБ RAM, 5 NIC
PC1/2	`vios-adventerprisek9-m.spa.159-3.m9`	1 vCPU, 1024 МБ RAM

В сумме: 18 назначенных vCPU и 50 ГиБ гостевой RAM; отдельно требуется память для PNETLab. Проверенный хост: 24 vCPU и около 63 ГиБ RAM. Это одна полная одновременно работающая топология, не несколько студенческих экземпляров.

Клиенты IOSv моделируют конечные узлы, не участвуют в маршрутизации underlay. Для ICMP и ARP достаточно их консолей. Для `curl` можно заменить клиентов Linux, сохранив адреса и MTU 1500.

Импорт и доступ

1. Используйте static_initial.zip либо evpn_initial.zip. Альтернатива — преподаватель размещает `.unl` в `/opt/unetlab/labs/VXLAN-Course/`.
2. Создайте персональную копию с **новым UUID** средствами PNETLab. Простое переименование файла не создаёт независимую сессию.
3. Убедитесь, что четыре Nexus имеют по 4 vCPU и 12 ГиБ RAM. Не используйте старые значения шаблона 2 vCPU / 8 ГиБ.
4. Запускайте узлы по одному. Первичная загрузка Nexus и применение bootstrap занимают несколько минут. Дождитесь приглашений `SPINE1#`, `LEAF1#` и т. д.
5. Учебная учётная запись Nexus: **`admin / LabVxlan2026!`**. Она публичная, только для изолированной работы. В стартовых конфигурациях уже заданы hostname, loopback, IP/MTU физических uplink; OSPF и VXLAN студент настраивает самостоятельно. PC уже имеют IP.
6. Проверьте `show version`, `show interface brief`, `show running-config`. Если hostname/адреса не совпадают — импорт не завершился либо не сработал. Не считайте наличие `.unl` доказательством применения конфигурации.
7. После выполнения сохраните конфигурацию: Nexus — `copy running-config startup-config`, IOSv — `write memory`.

Печатные методички: Lab 1 PDF, Lab 2 PDF. Ссылки и загрузки удобнее открывать из Markdown-версии.

После проверки старта переходите к общему underlay с командами и объяснениями, затем к пошаговому заданию нужной работы. Оба задания начинаются со своего initial-стенда; готовую Lab 1 не следует использовать как скрытую основу Lab 2.

`Mgmt0` на Nexus — NIC 0, не подключён. `Ethernet1/1` — NIC 1; ошибочная нумерация на один порт нарушает топологию. Интернет/Cloud/pnet0 не нужны.

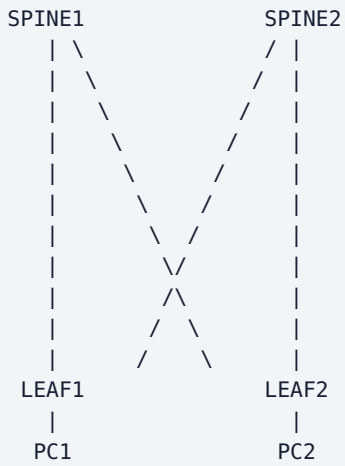
Стартовые и эталонные лаборатории имеют разные UUID. Эталоны помечены admin-only; проверка реальных прав веб-доступа требует отдельной непривилегированной учётной записи.

Захват

Используйте Capture на uplink Leaf и отдельно на access-порту. При ECMP один конкретный поток может выбрать любой Spine: отсутствие пакетов на одном uplink не доказывает отсутствие VXLAN. Захватите оба либо временно отключите второй uplink **в своей копии** и затем восстановите его.

Источник требований: Cisco Nexus 9000v Guide 10.5(x). Виртуальный dataplane не воспроизводит производительность и все аппаратные функции физического Nexus.

Общая топология и адресация



Каждый Leaf связан с обоими Spine. **Физического линка Leaf–Leaf и Spine–Spine нет.** Точные связи — в таблице:

Сегмент	Сторона A	IP A	Сторона B	IP B
S1-L1	SPINE1 E1/1	10.0.11.1/30	LEAF1 E1/1	10.0.11.2/30
S1-L2	SPINE1 E1/2	10.0.12.1/30	LEAF2 E1/1	10.0.12.2/30
S2-L1	SPINE2 E1/1	10.0.21.1/30	LEAF1 E1/2	10.0.21.2/30
S2-L2	SPINE2 E1/2	10.0.22.1/30	LEAF2 E1/2	10.0.22.2/30
ACCESS-L1	LEAF1 E1/3	access VLAN 10	PC1 Gi0/0	192.168.10.11/24
ACCESS-L2	LEAF2 E1/3	access VLAN 10	PC2 Gi0/0	192.168.10.12/24

Узел	Loopback0: router-id/BGP	Loopback1: VTEP
SPINE1	10.255.0.1/32	—
SPINE2	10.255.0.2/32	—
LEAF1	10.255.0.11/32	10.255.1.11/32

Узел	Loopback0: router-id/BGP	Loopback1: VTEP
LEAF2	10.255.0.12/32	10.255.1.12/32

- Underlay: OSPF `UNDERLAY`, area `0.0.0.0`, point-to-point на routed uplink, MTU 9216.
- Overlay: VLAN **10** ↔ L2VNI **10010**, NVE1 source `loopback1`.
- EVPN: iBGP AS **65000**, соседи по Loopback0; оба Spine — route reflector, оба Leaf — RR clients.
- Spine не имеют tenant VLAN, VNI или NVE. Передают outer IP и отражают EVPN-маршруты.
- Шлюз клиентам не требуется: они в одной подсети. Не настраивайте SVI или маршруты `192.168.10.0/24` в underlay вместо VXLAN.

Теория: основы VXLAN и статическая ingress replication

1. Какую задачу решает VXLAN

Представим два сервера одной подсети, подключённые к разным коммутаторам. Обычный Ethernet требует L2-пути между ними: например, последовательности trunk с одним VLAN. Если вместо этого между коммутаторами находится маршрутизируемая IP-сеть, широковещательный ARP-запрос через неё не пройдёт. Маршрутизатор не пересылает произвольные Ethernet-кадры соседнего сегмента.

VXLAN (Virtual Extensible LAN) решает эту задачу инкапсуляцией: исходный Ethernet-кадр помещается внутрь UDP/IP-пакета. Транзитные маршрутизаторы доставляют внешний IP-пакет, а крайний коммутатор извлекает исходный кадр. Так создаётся логический L2-сегмент поверх L3-инфраструктуры.

В нашей работе PC1 `192.168.10.11/24` и PC2 `192.168.10.12/24` находятся в одном логическом сегменте. Физического trunk между Leaf нет. Оба Spine и uplink Leaf работают как обычная маршрутизируемая сеть.

VXLAN полезен для разделения инфраструктурной IP-связности и сетей арендаторов, а также для построения fabric с несколькими равностоимостными L3-путями. Это не означает, что L2 следует растягивать без ограничений: broadcast domain и последствия L2-ошибок сохраняются внутри соответствующего overlay-сегмента.

2. Термины и роли устройств

Термин	Смысл	В нашей топологии
Underlay	Физическая IP-инфраструктура, доставляющая outer IP-пакеты	Routed uplink, OSPF, loopback
Overlay	Логическая сеть поверх underlay	Общая клиентская Ethernet-сеть
VTEP	VXLAN Tunnel Endpoint: инкапсуляция и декапсуляция	LEAF1 и LEAF2
NVE	Логический интерфейс Network Virtualization Edge в NX-OS	`interface nve1` на Leaf

Термин	Смысл	В нашей топологии
VLAN	Локальный Ethernet broadcast domain	VLAN 10 на access-порту
VNI	VXLAN Network Identifier, идентификатор overlay-сегмента	L2VNI 10010
BUM	Broadcast, unknown unicast, multicast	Например, ARP до изучения MAC

Leaf подключает конечные узлы. **Spine** связывает Leaf через L3. В этой схеме Spine не являются VTEP: им не нужны клиентский VLAN, VNI и NVE. Пакет проходит через Spine без снятия VXLAN-инкапсуляции.

VNI — 24-битное поле, то есть пространство примерно 16 миллионов идентификаторов. VLAN ID — 12-битное поле с 4094 обычными пользовательскими VLAN. Это размер пространства идентификаторов, а не гарантия такого количества одновременно поддерживаемых сегментов на конкретной платформе.

Отображение **VLAN 10 → VNI 10010** означает: кадр, пришедший в локальный VLAN 10, переносится через VXLAN с меткой 10010. Значения VLAN и VNI не обязаны совпадать численно. В этой работе они должны быть согласованы на обоих Leaf; разные VNI представляют разные overlay-сегменты.

3. Разделение underlay и overlay

Underlay должен знать, как доставить пакет до **удалённого VTEP**, но не обязан знать адреса клиентов. Поэтому в OSPF мы передаём физические подсети и loopback, но не `192.168.10.0/24`.

Назначение	LEAF1	LEAF2
Loopback0: router-id, позже BGP	10.255.0.11/32	10.255.0.12/32
Loopback1: адрес VTEP	10.255.1.11/32	10.255.1.12/32
Подключённый клиент	192.168.10.11/24	192.168.10.12/24

Loopback не привязан к одному физическому uplink. Пока существует хотя бы один маршрут, адрес VTEP может оставаться доступен после отказа отдельного линка. Это не делает сам Leaf отказоустойчивым: при его остановке единственный подключённый к нему клиент потеряет сеть.

Клиенты находятся в одной `/24`, поэтому сначала выполняют ARP друг для друга, а не обращаются к шлюзу. SVI и default gateway в этой работе не требуются. Добавление маршрута до клиентской подсети в OSPF не заменяет VXLAN.

4. Что находится внутри VXLAN-пакета

```
Outer Ethernet
  Outer IPv4: 10.255.1.11 -> 10.255.1.12
    UDP: destination port 4789
      VXLAN: VNI 10010
        Inner Ethernet: MAC-PC1 -> MAC-PC2
          Inner IPv4: 192.168.10.11 -> 192.168.10.12
            ICMP echo request
```

Для ARP внутри VXLAN вместо inner IPv4 будет Ethernet payload ARP. **Outer** описывает доставку между VTEP, **inner** — исходный клиентский кадр.

На каждом routed hop изменяются внешний Ethernet-заголовок и outer IP TTL. Inner MAC и inner IP не подменяются адресами Spine: клиентский кадр переносится через сеть как полезная нагрузка. Поэтому MAC назначения внешнего кадра на LEAF1 — MAC ближайшего Spine, а IP назначения внешнего пакета — `10.255.1.12`, удалённый VTEP.

Внешний UDP source port обычно вычисляется из внутреннего потока и создаёт энтропию для ECMP. Один поток может идти только через один Spine, а другой — через второй. Равностоимостная маршрутизация не означает обязательное чередование каждого пакета по двум uplink. Outer destination port в этой работе — **4789**.

5. Как проходит первый ARP и последующий ping

Рассмотрим ситуацию, когда таблицы ARP и MAC ещё не содержат нужных записей.

1. PC1 определяет по маске, что PC2 находится в той же подсети. Он отправляет broadcast ARP: «Кто имеет `192.168.10.12`?».
2. LEAF1 получает кадр на E1/3 в VLAN 10 и изучает MAC PC1 как локальный.
3. LEAF1 отображает VLAN 10 в VNI 10010. Поскольку это broadcast, ему нужен список удалённых VTEP, участвующих в данном VNI.
4. Статическая ingress replication задаёт peer `10.255.1.12`. LEAF1 формирует unicast VXLAN-пакет к этому VTEP, сохраняя broadcast внутри.
5. OSPF-маршрут выбирает один из Spine. Spine маршрутизирует outer IP-пакет, не обрабатывая ARP клиента.
6. LEAF2 принимает VXLAN для VNI 10010, декапсулирует кадр и передаёт ARP в свой VLAN 10 на PC2. Из принятого трафика он может изучить удалённый MAC PC1 за VTEP `10.255.1.11`.
7. PC2 отвечает своим MAC. Ответ идёт обратно через VXLAN. LEAF1 изучает MAC PC2 за удалённым VTEP.
8. Последующие ICMP-пакеты передаются с известными MAC и VTEP. Первый ping иногда теряет пакет во время ARP; повторный тест должен давать 5/5.

Это механизм **flood-and-learn**: неизвестные назначения сначала затапливаются в пределах VNI, а соответствия MAC → VTEP изучаются из data plane.

6. Что означает ingress replication

Ingress — входной VTEP, получивший исходный кадр от клиента. **Replication** — создание копий BUM-трафика для удалённых участников VNI.

Вместо multicast underlay VTEP отправляет отдельный unicast VXLAN-пакет каждому настроенному peer. При двух VTEP достаточно одной удалённой копии. Если удалённых участников десять, отправитель должен создать до десяти копий одного BUM-кадра. Это важно для оценки нагрузки, хотя в маленькой лаборатории она почти незаметна.

Статический `peer-ip` задаёт **участника replication list**, а не IP клиента и не IP соседнего Spine. На LEAF1 это `10.255.1.12`; на LEAF2 — `10.255.1.11`.

Multicast underlay и PIM здесь не нужны. Это не запрещает служебный multicast самого OSPF. В Lab 2 список участников VNI будет строиться по EVPN вместо ручного `peer-ip`.

7. MTU и накладные расходы

При внешнем IPv4, без дополнительных VLAN-тегов и без учёта FCS:

Добавляемый заголовок	Размер
Outer Ethernet	14 байт
Outer IPv4	20 байт
UDP	8 байт
VXLAN	8 байт
Всего относительно исходного Ethernet-кадра	**50 байт**

Если клиент отправляет inner IP-пакет 1500 байт, его Ethernet-кадр содержит ещё 14 байт. Outer IP должен вместить `20 + 8 + 8 + 14 + 1500 = 1550` байт. Поэтому для такого трафика **outer IP MTU нужен не меньше 1550**, а длина внешнего Ethernet-кадра без FCS будет 1564.

Мы задаём MTU **9216** на всех routed uplink underlay и оставляем клиентам 1500. Нужна согласованность всего пути, а не только одного интерфейса. Маленький ping может работать при ошибочном MTU и создавать ложное впечатление исправности.

Проверка на IOSv `ping ... size 1500 df-bit repeat 5` задаёт размер IP-пакета и запрещает его фрагментацию флагом DF. Сопоставьте вывод CLI с inner IP total length в Wireshark. Это проверяет выбранный путь и размер, а не гарантирует доставку любого размера на всех путях fabric.

8. Как отличать уровни неисправности

Симптом	Что проверять сначала
Нет OSPF FULL	Физический линк, IP/маску, area, MTU, network type
OSPF есть, remote VTEP не пингуется	Анонс Loopback1 и маршрут до него
VTEP доступен, VNI не Up	VLAN/VNI, NVE, source-interface и access-порт
NVE/VNI Up, PC не отвечает	Access VLAN, remote MAC, peer-IP, ARP и захват
Маленький ping работает, большой нет	MTU всего выбранного underlay-пути
Трафика нет в одном Capture	Какой uplink выбран ECMP; захват на обоих

Неверный peer-IP может нарушить BUM, хотя уже изученный unicast некоторое время продолжит работать. Нельзя делать вывод только по одному ping, не учитывая кэши ARP/MAC.

9. Чего эта схема не делает

- VXLAN **не шифрует** трафик и не заменяет IPSec. VNI — сегментация, а не криптографическая защита.
- L2VNI не выполняет автоматически маршрутизацию между клиентскими подсетями. Для неё нужны отдельные gateway/IRB-механизмы, которые здесь не настраиваются.
- Два Spine дают резервирование underlay-пути, но не резервируют одиночное подключение клиента к Leaf.
- Виртуальный Nexus проверяет логику конфигурации и протоколов, но не производительность ASIC физического коммутатора.

10. Вопросы перед практикой

1. Почему в OSPF нет клиентской подсети, хотя PC1 достигает PC2?
2. Чем `source-interface loopback1` отличается от `peer-ip`?
3. Какой MAC видит Spine: PC2 или ближайшего underlay-соседа?
4. Почему VLAN 10 и VNI 10010 — не одно и то же поле?
5. Почему 5/5 между VTEP недостаточно для зачёта клиентского VXLAN?

Далее: общий underlay пошагово и ход Lab 1.

Источники

- RFC 7348 — VXLAN.
- Cisco Nexus 9000v Guide, Release 10.5(x).
- Cisco Nexus 9000 NX-OS VXLAN Configuration Guide, Release 10.5(x).

Общий этап: underlay и OSPF пошагово

Эту последовательность выполняют **перед настройкой VXLAN в каждой работе**. В Lab 2 используйте отдельную `evpn_initial.unl`, а не стенд с оставшимися static peers из Lab 1. Полная адресация одинакова.

1. Проверьте, что стартовые конфигурации применились

Стартовый `.unl` уже задаёт hostname, учётную запись, IP loopback, IP/MTU routed uplink и адреса PC. OSPF, VLAN 10/VNI 10010, NVE и BGP студент добавляет сам. Примеры ниже повторяют IP/MTU намеренно: их можно сверить со стартом и восстановить правильные значения, не угадывая контекст.

На каждом Nexus:

```
show version
show interface brief
show running-config
```

На PC1 и PC2:

```
show ip interface brief
```

Ожидается C9300v с NX-OS 10.5(2), правильные имена узлов, PC1 `192.168.10.11/24`, PC2 `192.168.10.12/24`. У Nexus `Mgmt0` не подключён; рабочие порты начинаются с E1/1. Если вместо нужного имени видно `switch`, дождитесь окончания bootstrap и проверьте импорт, а не продолжайте на случайной конфигурации.

2. Разберитесь с режимами CLI

Команда	Что меняет
`configure terminal`	Переходит из EXEC LEAF1# в глобальную конфигурацию
`feature ospf`	Включает функцию OSPF в модульном NX-OS CLI
`router ospf UNDERLAY`	Создаёт/открывает процесс OSPF; `UNDERLAY` — локальное имя процесса
`router-id ...`	Задаёт стабильный уникальный идентификатор OSPF-маршрутизатора
`interface ...`	Открывает конфигурацию конкретного интерфейса
`exit`	Возвращается на один уровень CLI выше
`end`	Возвращается из любого режима конфигурации в EXEC

Копируйте только содержимое блока для указанного устройства. Приглашения вида `LEAF1(config-if)#` вводить не нужно. Отступы помогают читать иерархию, но **режим CLI задают команды**, а не количество пробелов. Поэтому `exit` и `end` важны.

3. Настройте SPINE1

```
configure terminal
feature ospf
router ospf UNDERLAY
  router-id 10.255.0.1
exit
interface loopback0
  ip address 10.255.0.1/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface Ethernet1/1
  no switchport
  mtu 9216
  ip address 10.0.11.1/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
exit
interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.0.12.1/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
end
```

Объяснение интерфейсных команд, общее для всех четырёх Nexus:

Команда	**Зачем нужна**
`interface loopback0`	Создаёт логический интерфейс, не зависящий от состояния одного uplink
`ip address .../32`	Задаёт адрес одного инфраструктурного узла; /32 — host prefix
`no switchport`	Делает Ethernet-порт маршрутизируемым, а не access/trunk
`mtu 9216`	Даёт underlay место для клиентского кадра и VXLAN overhead
`ip address .../30`	Задаёт адрес физического L3-линка; адреса на концах должны быть из одной /30
`ip ospf network point-to-point`	Убирает выбор DR/BDR на двухточечном линке; задаётся согласованно на обоих концах
`ip router ospf UNDERLAY area 0.0.0.0`	Включает интерфейс в процесс OSPF и area 0; анонсирует его сеть
`no shutdown`	Снимает административное отключение интерфейса

Spine не получает Loopback1: в этой схеме он не VTEP.

4. Настройте SPINE2

```
configure terminal
feature ospf
router ospf UNDERLAY
  router-id 10.255.0.2
exit
interface loopback0
  ip address 10.255.0.2/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface Ethernet1/1
  no switchport
  mtu 9216
  ip address 10.0.21.1/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
exit
interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.0.22.1/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
end
```

От SPINE1 отличаются router-id, Loopback0 и адреса двух линков. Не копируйте `.1` Loopback0 на оба Spine: идентификаторы должны быть уникальны.

5. Настройте LEAF1

```
configure terminal
feature ospf
router ospf UNDERLAY
  router-id 10.255.0.11
exit
interface loopback0
  ip address 10.255.0.11/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface loopback1
  ip address 10.255.1.11/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface Ethernet1/1
  no switchport
  mtu 9216
  ip address 10.0.11.2/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
exit
interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.0.21.2/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
end
```

Loopback0 используется как router-id и позже BGP source. **Loopback1 — отдельный VTEP source**. Оба должны быть известны underlay: без маршрута к Loopback1 клиентский VXLAN не заработает, даже если BGP по Loopback0 установлен.

6. Настройте LEAF2

```
configure terminal
feature ospf
router ospf UNDERLAY
  router-id 10.255.0.12
exit
interface loopback0
  ip address 10.255.0.12/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface loopback1
  ip address 10.255.1.12/32
  ip router ospf UNDERLAY area 0.0.0.0
exit
interface Ethernet1/1
  no switchport
  mtu 9216
  ip address 10.0.12.2/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
exit
interface Ethernet1/2
  no switchport
  mtu 9216
  ip address 10.0.22.2/30
  ip ospf network point-to-point
  ip router ospf UNDERLAY area 0.0.0.0
  no shutdown
end
```

E1/3 обоих Leaf пока не трогаем: его access VLAN будет настроен в конкретной работе. Не превращайте uplink E1/1 и E1/2 в trunk.

7. Контрольная точка: OSPF и ECMP

На всех четырёх Nexus:

```
show ip ospf neighbors
show ip route ospf
```

Должно быть **два соседа FULL** на каждом устройстве. `FULL/ -` нормально для point-to-point: выбор DR/BDR не выполняется. Кратковременные INIT/EXSTART во время сходимости допустимы, но не являются готовым результатом.

На LEAF1:

```
show ip route 10.255.1.12/32
ping 10.255.1.12 source 10.255.1.11 count 5
```

На LEAF2:

```
show ip route 10.255.1.11/32
ping 10.255.1.11 source 10.255.1.12 count 5
```

`source` выбирает локальный адрес-источник — в данном случае будущий VTEP. `count 5` ограничивает число echo request. В NX-OS это синтаксис `count`, а на IOSv-клиентах далее используется `repeat`.

При обоих исправных путях LEAF1 должен иметь next-hop **10.0.11.1 и 10.0.21.1** до remote VTEP; LEAF2 — **10.0.12.1 и 10.0.22.1**. В проверенном выводе это `ubest/mbest: 2/0`. Точная стоимость маршрута зависит от reference bandwidth и параметров образа; важны достижимость и два равностоимостных пути.

Если FULL нет, сначала проверьте адреса/маски, administrative state, area, MTU и network type на обоих концах. Если соседство есть, но remote Loopback1 отсутствует, проверьте включение именно этого loopback в OSPF. **Не переходите к NVE при неработающем underlay.**

8. Сохраните этот этап и продолжите свою работу

На каждом Nexus:

```
copy running-config startup-config
```

`running-config` — текущее состояние в памяти, `startup-config` — сохранённая конфигурация для следующего запуска. Выполненная команда настройки без сохранения и без проверки состояния ещё не является завершённой работой.

- Продолжить Lab 1: VLAN/VNI и статический NVE.
- Продолжить Lab 2: RR и BGP EVPN.

Lab 1. VXLAN со статической ingress replication

Время: ориентировочно 90–120 минут, не считая загрузки. Предварительно: VLAN, ARP, OSPF, loopback, основы Wireshark. Это учебная последовательность: вводите команды по этапам, объясняйте их действие и фиксируйте проверки, а не вставляйте сразу весь конфиг.

Материалы: подготовка, адресация, теория, проверки, Wireshark, стартовый ZIP.

Результат

PC1 и PC2 обмениваются трафиком в одной подсети через два VTEP. В транзите виден VXLAN с VNI 10010; при отказе одного uplink Leaf связь восстанавливается через второй Spine.

Шаг 1. Подготовьте отдельную копию и underlay

Импортируйте `static_initial.unl` по инструкции подготовки, создайте персональную копию с новым UUID, запустите узлы по одному и дождитесь применения стартовых конфигураций.

Прочитайте теорию, затем выполните общий этап underlay: там приведены команды для SPINE1, SPINE2, LEAF1 и LEAF2, построчное объяснение и проверки OSPF/ECMP. Клиенты уже имеют адреса `.11` и `.12` в `192.168.10.0/24`; их шлюз не требуется.

Контрольная точка: каждый Nexus имеет два FULL, remote Loopback1 доступен с source local Loopback1, до remote VTEP есть два next-hop. На PC1 выполните `ping 192.168.10.12 repeat 5`: до overlay он не должен проходить. Если проходит, ищите лишний L2-мост или оставшуюся конфигурацию.

Шаг 2. Включите функции VXLAN и создайте VLAN/VNI

Выполните **на LEAF1 и затем на LEAF2** один и тот же блок:

```
configure terminal
feature nv overlay
feature vn-segment-vlan-based
vlan 10
  name TENANT-A
  vn-segment 10010
exit
interface Ethernet1/3
  switchport
  switchport mode access
  switchport access vlan 10
  spanning-tree port type edge
  no shutdown
end
```

Команда	Объяснение
`feature nv overlay`	Включает функцию NVE/VXLAN на Leaf
`feature vn-segment-vlan-based`	Разрешает VLAN-based отображение VLAN в VNI
`vlan 10`	Создаёт локальный клиентский broadcast domain
`name TENANT-A`	Задаёт понятное имя VLAN; само имя не влияет на доставку
`vn-segment 10010`	Связывает локальный VLAN 10 с VXLAN VNI 10010
`switchport`	Переводит E1/3 в L2-порт, если он был routed
`switchport mode access`	Делает порт нетегированным access, не trunk
`switchport access vlan 10`	Относит кадры клиента к VLAN 10
`spanning-tree port type edge`	Обозначает порт конечного узла и ускоряет переход в forwarding; не применять бездумно к межкоммутаторному линку
`no shutdown`	Включает порт административно

Эти команды **не выполняются на Spine**. E1/1 и E1/2 Leaf должны оставаться `no switchport`, поскольку переносят outer IP, а не клиентский VLAN.

Проверка на обоих Leaf:

```
show vlan brief
show interface Ethernet1/3 switchport
show interface brief
```

Контрольная точка: VLAN 10 существует, E1/3 — access VLAN 10 и up; routed uplink остались routed. Само создание VLAN/VNI ещё не соединяет два Leaf.

Шаг 3. Создайте статический NVE на LEAF1

```
configure terminal
interface nve1
  no shutdown
  source-interface loopback1
  member vni 10010
    ingress-replication protocol static
      peer-ip 10.255.1.12
end
```

Команда	Объяснение
`interface nve1`	Создаёт логический VXLAN-интерфейс VTEP; это не физический порт и не GRE Tunnel
`no shutdown`	Разрешает работу NVE
`source-interface loopback1`	Выбирает локальный VTEP IP `10.255.1.11` для outer VXLAN
`member vni 10010`	Включает VNI 10010 в данный NVE
`ingress-replication protocol static`	Выбирает ручной список удалённых получателей BUM
`peer-ip 10.255.1.12`	Добавляет VTEP LEAF2 в этот список; это не адрес PC2 и не ближайшего Spine

Шаг 4. Создайте обратную сторону на LEAF2

```
configure terminal
interface nve1
  no shutdown
  source-interface loopback1
  member vni 10010
    ingress-replication protocol static
      peer-ip 10.255.1.11
end
```

На LEAF2 local source теперь `10.255.1.12`, а remote peer — `10.255.1.11`. Обе стороны должны согласовать **VNI 10010**, но peer-IP у них противоположные. В этой работе не включайте BGP EVPN и `host-reachability protocol bgp`.

Проверка на обоих Leaf:

```
show interface nve1
show nve vni
show nve peers
```

Контрольная точка: NVE включён, VNI 10010 отображается как L2 VLAN 10, `UnicastStatic`, Up, DP. Отображение peer в статическом режиме может зависеть от поступившего трафика; окончательно проверяем его после ping.

Шаг 5. Проверка и захват

Сначала включите Capture на E1/3 и на обоих uplink одного Leaf. Затем **на PC1**:

```
ping 192.168.10.12 repeat 5
show arp
ping 192.168.10.12 size 1500 df-bit repeat 5
```

На PC2:

```
ping 192.168.10.11 repeat 5
show arp
```

`repeat 5` ограничивает число запросов на IOSv. `show arp` показывает, узнал ли клиент MAC другого клиента. `size 1500` проверяет большой IP-пакет, `df-bit` запрещает его фрагментацию. Первый ping может потерять пакет на ARP; повторный должен дать **100%, 5/5**.

На обоих Leaf:

```
show nve peers
show nve vni
show mac address-table vlan 10
show l2route mac all
```

Локальный MAC должен указывать на E1/3, удалённый — на `nve1(<remote-VTEP>)`. На проверенном NX-OS `show l2route mac all` в static flood-and-learn пуст; это не отменяет корректную динамическую MAC-запись и передачу. Producer BGP здесь не ожидается.

В Wireshark найдите `udp.port == 4789 && vxlan.vni == 10010`. Укажите outer IP VTEP и inner IP клиентов; для большого ping найдите inner IP total length 1500. Один ping между loopback не заменяет этот опыт. Подробные задания — в разделе Wireshark.

Шаг 6. Отказ и неисправность

Только **в своей копии** на LEAF1:

```
configure terminal
interface Ethernet1/1
  shutdown
end
show ip ospf neighbors
show ip route 10.255.1.12/32
```

`shutdown` имитирует административный отказ uplink. После сходимости остаётся один OSPF-сосед и next-hop `10.0.21.1` через E1/2. Повторите оба клиентских ping и DF 1500; в Capture докажите переход на uplink2. Не требуйте отсутствия потерь во время самой сходимости.

Затем **обязательно восстановите**:

```
configure terminal
interface Ethernet1/1
  no shutdown
end
show ip ospf neighbors
show ip route 10.255.1.12/32
```

Дождитесь двух FULL и двух next-hop, повторите клиентский ping. Не сохраняйте стенд с намеренно выключенным интерфейсом.

Дополнительный опыт по согласованию с преподавателем: измените peer-IP LEAF1 на несуществующий и объясните, почему BUM перестал доставляться при исправном underlay. Старый известный unicast может временно работать; учитывайте ARP/MAC-кэши. Верните peer `10.255.1.12` и подтвердите восстановление. Не очищайте таблицы чужих лабораторий.

Шаг 7. Сохраните результат и объясните его

На всех Nexus:

```
copy running-config startup-config
```

На IOSv-клиентах, если меняли их конфигурацию:

```
write memory
```

Объясните своими словами цепочку: access VLAN → VNI → local VTEP → outer IP route → remote VTEP → access VLAN. Назовите, какие таблицы/пакеты подтвердили каждый переход. Используйте чек-лист проверки, а не только факт принятия команд CLI.

Отчёт и оценка

Приложите не полные конфиги, а ключевые строки, реальные выводы проверок, PCAP и краткие объяснения. Требуются: underlay/ECMP (20%), VLAN/VNI/NVE и remote MAC (25%), двусторонняя передача и VXLAN в PCAP (30%), отказ/восстановление и диагностика (15%), ответы по инкапсуляции/MTU (10%). Без доказательства реального VXLAN dataplane работа не считается выполненной.

Проверка Lab 1

На всех Nexus:

```
show version
show interface brief
show ip ospf neighbors
show ip route ospf
```

На LEAF1:

```
show ip route 10.255.1.12/32
ping 10.255.1.12 source 10.255.1.11 count 5
show vlan brief
show interface Ethernet1/3 switchport
show interface nve1
show nve peers
show nve vni
show mac address-table vlan 10
show l2route mac all
```

Ожидается: два OSPF FULL, два next-hop до удалённого VTEP при обоих исправных путях; VNI 10010 Up, VLAN 10, remote peer `10.255.1.12`. В статическом режиме peer изучается по data plane; его отображение может зависеть от наличия трафика. Сначала сгенерируйте клиентский ping.

На PC1 и PC2 соответственно:

```
ping 192.168.10.12 repeat 5
ping 192.168.10.11 repeat 5
show arp
```

Повторные тесты — 100%, 5/5. Для удалённого MAC должен существовать путь через NVE, а не локальный access. На Spine нет NVE и клиентской подсети; это нормально.

На проверенном образе статический VNI отображается как `UnicastStatic`, Up, DP; remote MAC — динамический на `nve1(<remote-VTEP>)`. `show l2route mac all` может оставаться пустым в flood-and-learn: не подменяйте им проверку MAC-таблицы и фактической передачи. Producer BGP здесь не ожидается.

После отказа LEAF1 E1/1: один OSPF сосед, один маршрутный путь; после сходимости ping проходит через SPINE2. После восстановления снова два соседа и два пути. Сохраните выводы до, во время и после отказа, не только конфигурацию `no shutdown`.

Wireshark: статический VXLAN

Захват: access E1/3 Leaf и оба underlay uplink того же Leaf. Сначала включите захват, затем генерируйте трафик. При ECMP поток может попасть на любой uplink.

Фильтр отображения	**Что искать**
`udp.port == 4789`	VXLAN в underlay
`vxlan.vni == 10010`	Нужный overlay-сегмент
`arp`	Запрос/ответ ARP; фильтр совпадает и с inner ARP
`icmp`	Inner ICMP, в том числе внутри VXLAN

Фильтр отображения	Что искать
`ospf`	Underlay control plane, не VXLAN

1. Найдите ARP, перенесённый в VXLAN. Подпишите outer source/destination IP, UDP destination port, VNI, inner MAC, target IP ARP.
2. Найдите echo request и echo reply. Сравните outer IP `10.255.1.11/12` с inner IP `192.168.10.11/12`.
3. Сравните access-кадр и VXLAN-пакет: где появляется инкапсуляция? Меняются ли внутренние адреса клиентов?
4. Объясните, почему outer Ethernet destination на Leaf — MAC соседнего Spine, а outer IP destination — удалённый VTEP.
5. Посчитайте overhead; учитывайте VLAN-теги, FCS и то, что IP MTU не включает внешний Ethernet header.
6. После отказа одного uplink найдите трафик на резервном, сохранив корректные VNI и VTEP IP.

Приложите PCAP и номера пакетов. Если Wireshark не распознал VXLAN автоматически, используйте Decode As для UDP/4789; одного наличия UDP/4789 без корректного VXLAN header недостаточно.