PTPBox — это фреймворк на основе bash-скриптов, который использует Linux Network Namespace для создания изолированных PTP-сетей на одной машине с несколькими сетевыми интерфейсами (NIC). Это позволяет эмулировать распределённую PTP-инфраструктуру (например, Grandmaster, Boundary Clock, Ordinary Clock) без необходимости в физических устройствах. Вот ключевые детали:

**Назначение**

* Эмуляция PTP-сетей  
  Создание сложных топологий (например, несколько хопов между Grandmaster и клиентами) на одном сервере.
* Тестирование и разработка  
  Используется для проверки работы PTP-стека Linux (LinuxPTP), отладки конфигураций и нагрузочного тестирования.
* Бюджетные решения  
  Работает даже на оборудовании с несколькими встроенными NIC (например, Protectli Vault-6 с шестью портами).

**Особенности**

1. Изоляция через Network Namespace
   * Каждый узел (Grandmaster, Boundary Clock и т.д.) запускается в отдельном namespace, что обеспечивает изоляцию сетевых настроек и процессов (аналог контейнеров).
   * Взаимодействие между узлами происходит через физические NIC или виртуальные интерфейсы (veth pairs).
2. Управление через Tmux
   * Все процессы (например, **ptp4l**, **phc2sys**) запускаются в отдельных панелях и окнах Tmux, что упрощает мониторинг и отладку.
   * Запуск осуществляется одной командой: **./run.sh**.
3. Поддержка аппаратных NIC
   * Совместим с высокопроизводительными картами (NVIDIA CX6) и бюджетными решениями (Intel i210 в Protectli Vault-6).
   * Позволяет тестировать PTP-синхронизацию на реальном железе.

**Лицензирование**

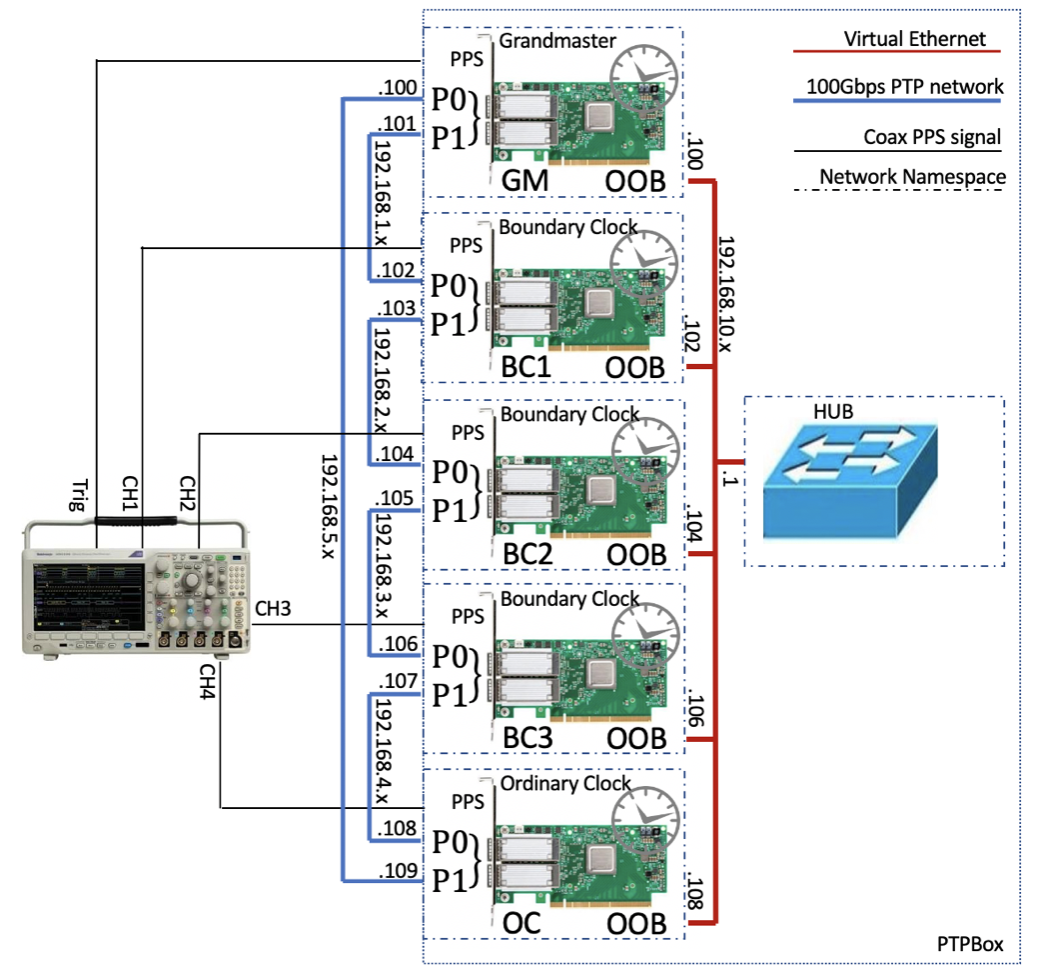
* Распространяется под лицензией OWF CLA 1.0 / OWFa 1.0 (Open Web Foundation).
* Лицензия не гарантирует отсутствие патентных претензий и требует самостоятельного обеспечения соответствия правам третьих лиц.
* Код предоставляется «как есть», без гарантий работоспособности.

**Примеры использования**

1. Эмуляция двух хопов PTP
   * Grandmaster (GM) → Boundary Clock (BC) → Ordinary Clock (OC) .
   * Настройка на Protectli Vault-6:

./run.sh --topology gm-bc-oc

* + Каждый узел использует отдельный NIC и namespace.

1. Тестирование с NVIDIA CX6
   * Запуск 7 узлов на сервере с семью NIC для нагрузочного тестирования.
   * Мониторинг через Tmux:  
     

**Требования**

* Программные
  + LinuxPTP (демоны **ptp4l**, **phc2sys**).
  + Tmux для управления сессиями.
  + Поддержка Network Namespace в ядре Linux.
* Аппаратные
  + Минимум 2 NIC (рекомендуется 4+ для сложных топологий).
  + Примеры совместимости:
    - Protectli Vault-6 (6x Intel i210).
    - Серверы с NVIDIA CX6 (7+ портов).

**Связь с другими инструментами**

* Clientgen  
  Генерирует нагрузку PTP-клиентов, которую можно направить в сеть, созданную PTPBox.
* DiffPHC  
  Измеряет разницу между PHC-часами на разных узлах в эмулированной сети.
* Open Time Instrument  
  Собирает PPS-метки с NIC для анализа точности синхронизации.

**Преимущества**

* Экономия ресурсов  
  Заменяет десятки физических устройств одной машиной.
* Гибкость  
  Поддержка произвольных топологий (линейных, кольцевых, звёздных).
* Интеграция с LinuxPTP  
  Позволяет тестировать upstream-версии PTP-стека.

**Ограничения**

* Зависимость от железа  
  Для высокоточной синхронизации требуются NIC с поддержкой аппаратного таймстампинга (например, Intel i210, NVIDIA CX6).
* Сложность настройки  
  Требует понимания Network Namespace и работы LinuxPTP.

PTPBox идеально подходит для:

* Разработчиков PTP-стека.
* Инженеров, тестирующих оборудование с PTP.
* Исследователей временных протоколов в изолированных средах.