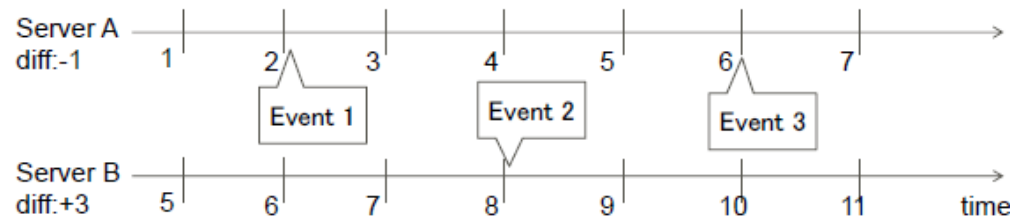
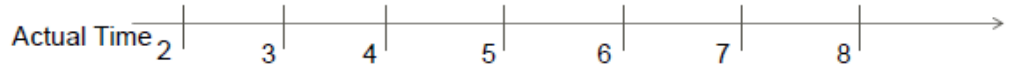


1) Precision Time Protocol (PTP) on Linux

Ссылка: файл "1 Precision Time Protocol on Linux.pdf"

Определение порядка возникновения событий в системе основано на метках времени (timestamp). Метки времени собираются из нескольких источников.

NTP (Network Time Protocol) обеспечивает синхронизацию с точностью до миллисекунд, но может быть так, что в одну миллисекунду произойдет несколько событий.



Event Ordering based on Actual Time

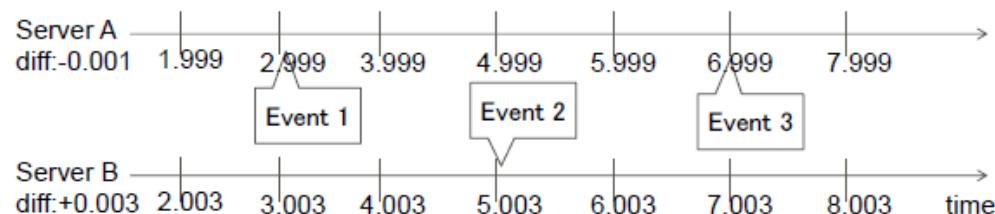
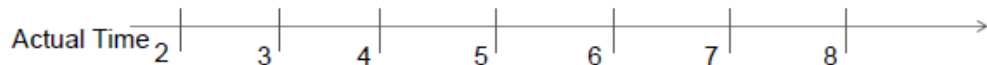
Time	Event
3	Event 1
5	Event 2
7	Event 3

Event Ordering based on Timestamp

Time	Event
2	Event 1
6	Event 3
8	Event 2

← reversel

PTP (Precision Time Protocol) обеспечивает точность с долями микросекунд и синхронизирует время в локально работающих системах и устройствах.



Event Ordering based on Actual Time

Time	Event
3	Event 1
5	Event 2
7	Event 3

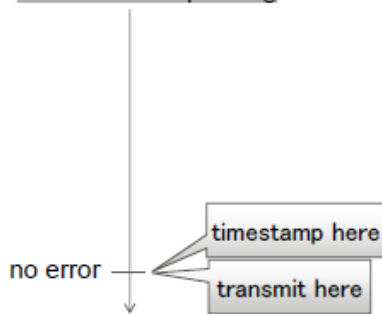
Event Ordering based on Timestamp

Time	Event
2.999	Event 1
5.003	Event 2
6.999	Event 3

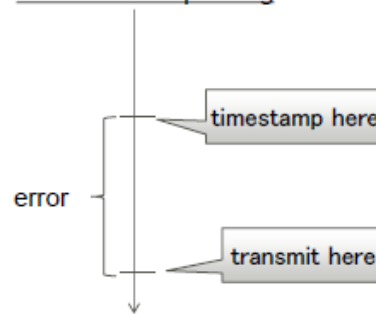
← correct!

Необходимо учитывать не только ошибки передачи данных, но и то, что на саму передачу требуется время, за которое метка уже устаревает.

Ideal timestamp timing

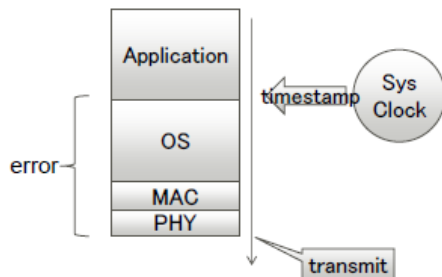


Real timestamp timing



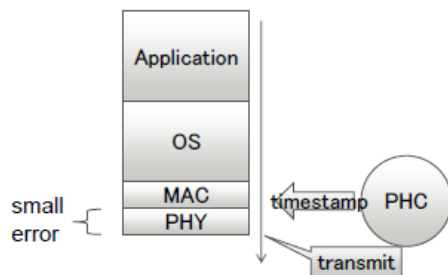
Как можно собирать метки:

- на уровне программного обеспечения (приложения, ОС). Ошибка может быть весьма велика.



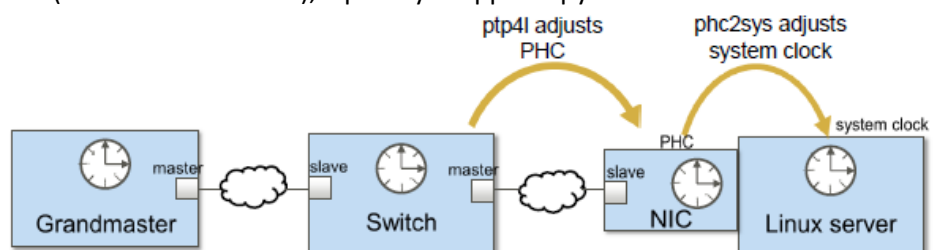
- на уровне аппаратного обеспечения. Ошибка минимальна.

Hardware Timestamping

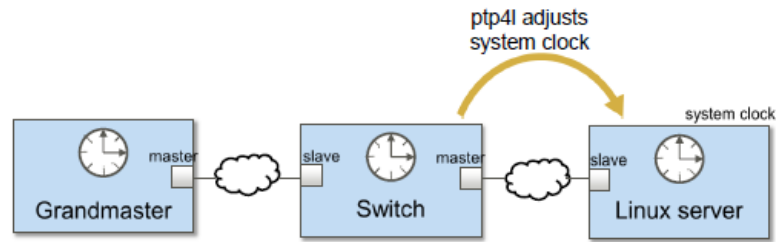


Существует несколько приложений, реализующих *PTP* на *linux*.

- *ptp41* - реализация *PTP*.
 - Обычные / граничные часы
 - Аппаратные / программные метки времени
 - Задержка между запросом-ответом
 - IEEE 802.3 (Ethernet) / UDP IPv4 / UDP IPv6 протоколы
- *phc2sys* - синхронизирует двое часов (*PHC* и системные часы)
 - Когда используются аппаратные временные метки, то *ptp41* корректирует *PHC* (PTP Hardware Clock), а *phc2sys* корректирует системные часы.



- Когда мы используем программные временные метки, то `phc2sys` становится не нужен.



- `pmc` (PTP Management Client) - отправляет *PTP* сообщения на *PTP* узлы.
 - PTP Management messages:
 - GET action: взять текущее значение данных
 - SET action: обновить значения переменных
 - CMD action: вызвать какие-нибудь события
 - Многие PTP устройства еще не поддерживают некоторые сообщения

1) Computer Time Synchronization Concepts

Ссылка: файл "**2 Computer Time Synchronization Concepts - 2014-04-29.pdf**"

При работе с временем необходимо учитывать то, что время может быть летним/зимним, нужно помнить про часовые пояса.

Greenwich Mean Time (GMT)

Coordinated Universal Time (UTC)

Чтобы получать максимальную точность измерения времени, используют атомарные часы, которые работают на переходах атомов цезия между энергетическими уровнями - они способны давать точность до 10^{-15} секунд.

Интересно, что в силу изменений, происходящих с планетой, время на Земле сдвигается каждые несколько десятилетий на секунды. Такие скачки называют "прыгающими секундами".

Современные компьютеры имеют на материнской плате чип Real Time Clock (RTC), который работает от батареи, даже когда компьютер выключен. Программные часы используют механизм прерываний для работы с RTC чипом. Из-за этого может возникнуть проблема на виртуальных машинах, где прерывания эмулируются программно и их регулярность может нарушаться.

Точность системного времени

В Unix-подобных системах время можно получить с точностью до микро- (и даже нано-) секунд.

В Windows точность существенно ниже - даже в Windows 8 можно получить точность лишь до 0.1мкс.

Из-за того, что кристаллы на чипе имеют немного различную частоту, время иногда слегка искажается в большую или меньшую сторону.

Для поддержания точности и исправления ошибок вроде перечисленных выше есть различные реализации NTP и PTP.

Все время в компьютере обычно считается в UTC и уже в приложениях приводится к читаемому виду. В Unix можно даже разные процессы запускать в их собственной временной зоне.

Протоколы передачи времени

NTP и PTP работают, как процессы-демоны. PTP служит для повышения точности NTP.

NTP синхронизирует системное время компьютера с одним или более внешними источниками. Он работает только со временем UTC.

Иерархия синхронизации времени по NTP:

Каждый демон-процесс может быть клиентом, сервером или "коллегой" другого демона на другом устройстве.

- Клиент собирает время со множества серверов
- Сервер сам отдает свое время клиентам
- Равноправное устройство сравнивает свое время с остальными, пока все не согласится, какое правильное.

Ситуация, когда один клиент опирается сразу на несколько серверов, называется избыточностью NTP.

NTP-демон использует конфигурационный файл `ntp.conf`, в котором указаны серверы, откуда брать время, если они есть.

Для динамического выбора серверов может использоваться DHCP (Dynamic Host Configuration Protocol). Эта возможность пригождается при большом количестве клиентов, нуждающихся в источниках времени.

Согласно NTP, сервер не должен просто слать пакеты клиенту, а должен делать это только по требованию клиента - это снижает нагрузку на сеть.

Ошибки

Когда NTP в рабочем состоянии, нельзя менять системное время вручную - это вызовет скачок между тем временем, что было и что стало, а NTP попытается этот скачок компенсировать, и все станет плохо.

Клиент может не синхронизироваться с сервером - тогда надо проверить, что пакеты исправно ходят, для чего есть программа `ntpdump`, являющаяся частью NTP и работающая очень похоже на `ping`: отправляет 4 пакета и ждет 4 ответа.

При работе и отправке пакетов NTP использует 4 метки времени:

- клиент отправил запрос
- сервер получает запрос
- сервер отправляет ответ
- клиент получает ответ

Из этих меток можно узнать время, которое тратится на путешествие пакета по сети.

Но даже тут не все гладко - на этих шагах так же появляются задержки.

- 1) пока отправляющая программа делает метку и упаковывает в пакет
- 2) пакет должен дожидаться очереди перед отправкой в сеть, а это зависит от загруженности сети и производительности компьютера
- 3) сетевые драйвера начинают отправлять пакет, но может случиться коллизия, и тогда отправка будет отменена и возобновиться только через случайный период времени.
- 4) если передача происходит через хаб, которые уже 100 лет как устарели, то еще добавляется постоянная задержка, а если через роутер, то задержка будет случайной
- 5) даже когда пакет пришел - его надо еще прочитать согласно протоколам драйверов после прерывания.
- 6) наконец, пакет дошел до приложения, пройдя все сетевые уровни, и теперь надо делать еще одну метку времени, после чего уже можно посмотреть их разницу с той, что в пакете.
- 7) если приложение было остановлено в силу многозадачности системы, то придется еще ждать, пока процессор его возобновит.

Чтобы узнать задержку после получения пакета на проводах и его получения приложением, в RTP есть time stamp unit (TSU), который может ставить метки времени на пакет сразу после того, как он пришел с провода.

Проблемой остается то, что происходит на посреднических узлах вроде роутеров, где пакет может залежаться на целые миллисекунды.

Сравнение NTP и RTP

Несмотря на высокую точность RTP, NTP может быть вполне достаточно даже через WAN соединения без требования особых аппаратных средств, в то время как RTP без спец.оборудования совершенно бесполезен. Но это требование вполне окупается невероятной точностью.

Аппаратными источниками времени могут быть устройства, подключенные через шину PCI или USB устройства или радио-часы. Качество счета времени по этим устройствам зависит от:

- точности самого источника
- времени на прочтения метки времени
- степени детализации структуры, используемой для пересылки меток
- характеристик операционной системы
- реализации цикла, выполняющего синхронизацию в проложении