УДК 006.92+521.934

ТОЧНОЕ ВРЕМЯ В ГЛОБАЛЬНОЙ СЕТИ ИНТЕРНЕТ

С.Н. Каган

ФГУП «ВНИИФТРИ», Менделеево, Московская обл.
skagan@vniiftri.ru

Описываются принципы работы Протокола передачи точного времени через глобальную сеть Интернет NTP

The article describes principles of operation of the Protocol of accurate time transmission using the wide-area network of NTR

Ключевые слова: сервер, время, синхронизация, смещение времени

Введение

Широкое распространение компьютерных систем привело к появлению нового класса потребителей точного времени. В их число входят самые различные технические системы, использующие компьютеры и требующие синхронизации шкалы их системного времени с точностью порядка от нескольких миллисекунд до секунды.

Это, прежде всего, системы, использующие электронную регистрацию финансовых операций, требующие обязательную оцифровку времени совершения сделки, системы синхронизации многочисленных локальных компьютерных сетей, регистрации продолжительности телефонных переговоров в обычных и сотовых телефонных системах, регистрации временного трафика в сети Интернет и др.

Для лучшего понимания проблемы рассмотрим, каким образом в компьютерах формируется и хранится время, и какие методы возможно использовать для его подстройки.

Хранение шкалы времени в персональном компьютере

Со времени появления первых персональных компьютеров IBM PC/AT в 1984 г. все PC совместимые компьютеры формируют время одним и тем же способом. Каждый PC компьютер содержит двое часов, независимо от того, на каком процессоре он работает - 286, 386, 486 или Pentium. Эти часы называют различными именами, но для простоты назовем их программными и аппаратными часами. Программные часы работают, когда компьютер включен, и прекращают работу при его выключении. Аппаратные часы питаются от литиевой батарейки и функционируют даже тогда, когда компьютер выключен.

Работа программных часов обеспечивается использованием микросхемы таймера (счетчика) Intel 8254 (или функционально эквивалентного устройства). Этот таймер генерирует прерывания каждые 54,936 миллисекунды, или около 18,2 раз в секунду. Компьютерный BIOS (базовая система ввода вывода) содержит программу, которая считает количество прерываний и формирует данные в формате времени, которые могут быть прочитаны или установлены другими программными средствами. Например, операционная система может использовать информацию о времени с программных часов для фиксации времени создания или модификации файлов.

Интервал времени, за который было осуществлено n таймерных прерываний, можно представить как:

$$T = n \text{ x } \Delta T_{\text{тик}} \text{ (с)},$$

где $\Delta T_{\text{тик}}$ - интервал таймерных прерываний.

Программные часы не являются хорошим хранителем времени. Точность отсчета времени с их помощью ограничивается стабильностью формируемых таймером запросов на прерывание. Любые изменения в параметрах этих запросов могут привести к опережению или отставанию программных часов относительно верного времени. При длительной непрерывной работе компьютера программные часы могут уйти на значительную величину, возможно, на минуту или более за день. Программные часы имеют также ограниченное разрешение. Они могут формировать отсчет времени только с дискретностью, кратной интервалу времени между прерываниями (~55 миллисекунд). Недостатком программных часов является также прекращение их работы и, следовательно, потеря информации о времени при выключении компьютера. По этой причине необходимы также и аппаратные часы.

Работа аппаратных часов базируется на использовании специальной микросхемы Motorola 146818 Real Time Clock или ей подобной. Когда компьютер выключается, аппаратные часы устанавливаются по данным программных часов и продолжают работать от батарейки. Когда компьютер включается вновь, программные часы начинают работать и в пределах 1 секунды устанавливаются по аппаратным часам.

Хотя аппаратные и программные часы синхронизируются при включении питания, они работают от разных кварцевых генераторов и со временем расходятся друг относительно друга. Аппаратные часы изменяют свои показания только раз в секунду и не могут отображать доли секунды. Точность

их хода определяется качеством кварцевого генератора, от которого они работают. Обычно используются дешевые кварцы, чувствительные к изменениям температуры и другим воздействиям, и кварцевые генераторы имеют суточную погрешность по частоте порядка $1 \cdot 10^{-5}$ (~ 1 секунда за день). Реально аппаратные часы могут уходить за день на 5 - 15 секунд.

Как можно видеть из вышесказанного, ни программные, ни аппаратные часы не пригодны для точного хронометрирования. Но все-таки существуют пути решения проблемы хронометрирования персональных компьютеров.

Во-первых, величина «тика», т.е. интервала между таймерными прерываниями, зависит как от аппаратной платформы, на основе которой выполнен компьютер, так и от установленной на нем операционной системы. Величина «тика» 54.936 миллисекунд относится к платформе IBM PC/AT и операционным системам DOS и Windows.

Количество “тиков” в секунду в операционной системе Unix и подобной ей операционной системе Linux для различных аппаратных платформ показано в таблице 1.

Таблица 1

Архитектура	"Тики" в секунду
Alpha	1024
ARM	100
CRIS	100
i386	100
ia64	1024
m68k	100
MIPS	100
MIPS64	100
PA-RISC	100
PPC	100
PPC-64	100
S390	100
S390X	100
SH	100
Sparc	100
Sparc64	100

Как видно из таблицы, существует достаточное количество аппаратных

платформ, у которых интервал таймерных прерываний кратен секунде.

Рассмотрим более подробно, как в большинстве разновидностей ядер операционной системы Unix поддерживается системное время.

В Unix прерывания ядра аппаратным счетчиком происходят с фиксированной частотой: 100 Гц в ядре SunOS, 256 Гц в ядре Ultrix и 1024 Гц в ядре OSF/1.

Так как таймерный интервал Ultrix (величина, обратная частоте) не кратен секунде с точностью до микросекунд, ядро добавляет 64 мкс каждую секунду, так что шкала времени состоит из 255 циклов по 3906 мкс плюс один 3970 мкс. Аналогично, ядро OSF/1 добавляет 576 мкс каждую секунду, так что шкала времени состоит из 1023 циклов по 976 мкс плюс один 1552 мкс.

Во-вторых, в операционной системе Unix существует ряд механизмов подстройки времени и хода системных часов.

В таблице 2 представлены общие команды UNIX, связанные со временем.

Таблица 2

Команда	Описание
date	Показывает или устанавливает системные дату и время
adjtime ()	Подстраивает время системных часов дискретно, на определенную величину
settimeofday ()	Устанавливает время системных часов на указанное значение
gettimeofday ()	Возвращает данные часового пояса и системное время

В большинстве ядер Unix возможно изменение смещения часов относительно текущего времени подстройкой их частоты при помощи системного вызова adjtime (). Для ввода некоторого смещения частота часов изменяется добавлением или вычитанием фиксированной величины (tickadj) в каждом таймерном прерывании такое количество раз, которое необходимо для получения данного смещения. Величина смещения при этом подсчитывается по формуле:

$$T\text{см} = N\ x\ (\Delta T_{\text{тик}} \pm \text{tickadj})\ \ (\text{с}),$$

где N – количество таймерных прерываний с измененным периодом, требуемых для получения данного смещения;

tickadj - фиксированная величина, на которую изменяется интервал таймерных прерываний при расчете системного времени.

Точность вводимого смещения определяется возможностями tickadj и обычно характеризуется величиной порядка 5 мкс. Для реализации функции частотной подстройки необходимо периодическое выполнение системного вызова adjtime () . Для уменьшения джиттера системных часов в таком режиме необходимо, чтобы интервал вызова adjtime () был сравнительно небольшим, в районе 1 с.

Разрешение системных часов можно увеличить чтением имеющихся в компьютере аппаратных счетчиков для интерполяции системного времени между прерываниями по таймеру. В компьютерах DEC 5000/240 и в других машинах этого семейства имеется аппаратный регистр, который считает системные шинные циклы с частотой 25 МГц. Новая подпрограмма для ядра Ultrix microtime () использует этот регистр, чтобы интерполировать системное время между прерываниями по таймеру. Этим достигается точность порядка 1 мкс для всех значений времени, полученных через системные вызовы gettimeofday (). Для машин на платформе 3000 AXP Alpha архитектурой предусмотрен аппаратный счетчик циклов и машинная команда (rpcc) для его чтения. Этот счетчик работает на основной частоте часов центрального процессора или некоторой ее субгармонике. Новая подпрограмма microtime () для ядра OSF/1 автоматически определяет частоту счета и использует это тем же самым способом, как и подпрограмма Ultrix. Ядро SunOS уже включает системные часы с 1 мкс разрешающей способностью; так что в подпрограмме microtime () нет необходимости.

Синхронизация системного времени компьютеров

Для синхронизации шкалы системного времени компьютеров можно использовать различные методы и средства, опирающиеся на систему передач эталонных сигналов частоты и времени по различным каналам связи. Но для компьютеров, подключенных к глобальной сети Интернет, наиболее простым и удобным является использование синхронизации по имеющимся в сети эталонным серверам времени – тайм-серверам.

На сегодняшний день синхронизация шкал времени в компьютерных сетях происходит с использованием специально разработанных стандартных протоколов. В настоящее время самым популярным протоколом синхронизации шкал времени компьютеров является утвержденный международным документом (Request for Comments RFC-5905, June 2010) протокол NTP (Network Time Protokol) версии v.4, который обеспечивает наибольшую точность и надежность передачи времени.

Впервые в виде программного обеспечения этот протокол реализовал профессор Делавэрского университета Дэвид Миллз. Сетевой протокол времени NTP служит для осуществления синхронизации работы различных процессов в серверах и программах клиента. Он определяет архитектуру, алгоритмы, объекты и протоколы, используемые для указанных целей. Целью протокола является обеспечение максимально возможной точности и надежности, несмотря на значительный разброс задержек при прохождении через большое число промежуточных маршрутизаторов.

Протокол предлагает средства для определения характеристик и оценки ошибок локальных часов и временного сервера, который осуществляет синхронизацию. Предусмотрены возможности работы с иерархически распределенными источниками точного времени, такими как синхронизуемые радиочасы.

Точность, достижимая с помощью NTP, в значительной степени зависит от точности локальных часов и характерных скрытых задержек. Алгоритм коррекции временной шкалы включает учет задержек, коррекцию частоты часов и ряд механизмов, позволяющих достичь точности порядка нескольких миллисекунд, даже после длительных периодов, когда потеряна связь с синхронизирующими источниками.

Разработан одноименный программный пакет, реализующий этот протокол и входящий в состав большинства UNIX-подобных операционных систем. Для реализации передачи времени через Интернет организуются так называемые тайм-серверы. Тайм-сервер - это компьютер, системное время которого синхронизируется непосредственно от высокоточного источника сигналов времени, такого как, например, атомные часы. Для синхронизации своих компьютеров клиентам достаточно быть подключенными к сети Интернет и иметь клиентскую программу.

Использование NTP для управления и синхронизации системных часов

В модели NTP предполагается некоторое количество первичных серве-

ров, синхронизированных с помощью национальных служб времени. Целью NTP является передача информации о точном времени от этих серверов к клиентам через Интернет и коррекция ошибок, связанных с флуктуациями задержек в сети. Некоторое число локальных компьютеров могут выполнять функции вторичных серверов времени, общающихся с первичными на основе протокола NTP. В целях обеспечения надежности выбранные вторичные источники могут быть снабжены менее точными, зато более дешевыми радиочасами, используемыми в ситуациях, когда откажет первичный сервер или выйдет из строя ведущий к нему канал.

NTP работает по иерархической модели, в которой небольшое количество серверов дает время большому количеству клиентов. Клиенты на каждом уровне или слое (stratum) являются в свою очередь серверами большему количеству клиентов с более высоким номером слоя. Номера слоя увеличиваются от первичных (stratum 1) серверов к более низким пронумерованным слоям древовидной структуры. Клиенты могут использовать информацию от множества серверов, чтобы автоматически определить лучший источник времени.

Рисунок 1 поясняет иерархическую модель слоев серверов, используемую в NTP.

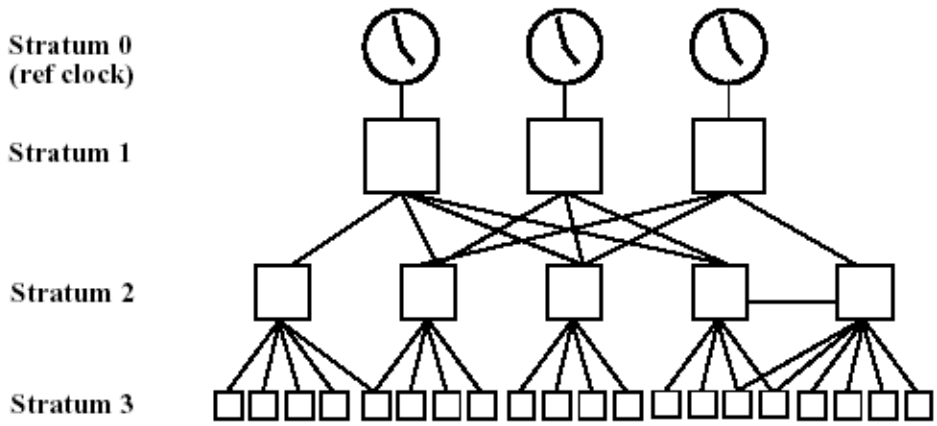


Рис. 1. Иерархическая модель слоев серверов

Серверы, которые непосредственно подключены к эталонным часам, называют stratum 1 серверами. Сами эталонные часы в терминологии NTP при этом являются stratum 0 серверами. Клиенты stratum 1 серверов являются

stratum 2 клиентами. Если они являются источником времени для других клиентов, они являются также stratum 2 серверами, а клиенты, которых они обслуживают, - stratum 3 клиентами и т.д. Номер слоя фактически означает дистанцию (количество промежуточных серверов времени) от данного компьютера до эталонного источника точного времени. Максимальный номер слоя NTP для клиента - 15; однако, практически редко можно найти клиентов с номером слоя более чем 4 или 5 в большинстве реальных топологий. Согласно обзору серверов NTP, сделанных Дэвидом Миллсом, больше чем половина клиентов NTP находится в слое 3, а почти все остальные - в слоях 2 и 4.

Отношения между серверами NTP и клиентами могут быть сконфигурированы различным образом. В широковещательном режиме - сервер посылает периодические широковещательные пакеты NTP любым клиентам, конфигурированным только на прием. Клиенты определяют время исходя из предположения, что задержка составляет несколько миллисекунд; сервер не принимает ответных NTP сообщений. В нешироковещательном режиме клиент NTP конфигурирован так, чтобы посылать периодические запросы времени одному или более серверам NTP. В симметричном режиме взаимодействуют два или несколько компьютеров, работающих с NTP, при этом каждый определяет, который компьютер среди них является наиболее точным, и синхронизируется по нему.

Протокол NTP создан с целью определения трех величин: смещения часов (clock offset), полной задержки от клиента до сервера и обратно и дисперсии. Все они вычисляются по отношению к часам выбранного сервера. Смещение часов определяет поправку, которую необходимо внести в показания местных часов, чтобы результат совпал показанием эталонных часов. Дисперсия характеризует максимальную ошибку локальных часов по отношению к эталонным.

Компоненты NTP

Работа NTP сопровождается использованием ряда специальных программ:

ntpd - принято называть демоном синхронизации времени. Один и тот же демон используется как для клиента, так и для сервера. Алгоритмы работы демона синхронизации описаны ниже.

ntpdate - программа ntpdate позволяет клиенту устанавливать дату от

сервера NTP без стационарной установки демона NTP. Для выбора лучшего источника времени может быть использована группа NTP серверов. По существу, допускается и разовая синхронизация по NTP. Но поскольку это взаимодействие происходит только однократно, часы, в конечном счете, вновь будут иметь неверный отсчет. Регулярное использование ntpdate может преодолеть этот недостаток, но это - далеко не идеальное решение.

ntpq - программа ntpq позволяет запрашивать состояние демона NTP с локальной или удаленной машины. Используя ntpq, администратор может проверить конфигурацию удаленного главного компьютера. Если такие запросы разрешены на главном компьютере, это может быть полезным для выбора сервера для синхронизации, потому что может быть определена информация о типе сервера и опорных часов.

ntptrace - информационная команда, которая отслеживает источник, от которого клиент получает время. Выводят список имен клиентов, номеров их слоя, смещение их времени от времени сервера, расстояние синхронизации и идентификатор опорных часов, подключенных к серверу, если они существуют. Расстояние синхронизации является критерием точности часов, предполагая, что источник времени корректен.

Алгоритмы NTP

Алгоритмы, используемые в NTP, тщательно и длительно совершенствовались с целью достижения максимально возможной степени точности и надежности.

Временной алгоритм NTP состоит из ряда последовательных операций. Эти операции осуществляются в следующем порядке:

1. Проверка признака целостности пакетов, подтверждающих возможность их использования.

2. Осуществление фильтрации с использованием "истории" измерений для данных часов, чтобы уменьшать jitter.

3. "Алгоритм пересечений". Позволяет исключить удаленные часы с неверно установленным временем.

4. Проверка кандидатов для синхронизации с помощью алгоритма кластеризации. Определяет и исключает наихудшие часы. В списке остаются только 10 (максимально возможное количество для обработки)

5. Синхронизация по источнику, выбранному с помощью алгоритма кластеризации. Выбирается лучший источник из оставшихся в списке.

6. Комбинация часов. Проверяет время от наилучшего источника, опираясь на время и ошибку его определения от других часов.

Проверка целостности

NTP проверяет целостность пакетов прежде, чем они будут обработаны. Есть несколько шагов для определения того, что пакет пригоден для использования. Проверка целостности является очень важной и необходимой задачей, поскольку, если все пакеты, посланные сервером, имеют нарушенный признак целостности, ntptrace исключит информацию из обработки. Подкоманда ntpq "peers" покажет серверы с нарушенным признаком целостности пакетов, отсутствием какого-либо символа перед их наименованием (у серверов с ненарушенным признаком целостности будет перед их наименованием иметься один из символов: "x" , " ." , " * ", "o" , "#" , или "+").

В каждом пакете формируются следующие признаки целостности:

1. Пакет не должен быть дубликатом другого пакета от того же сервера. Это устраняет возможность обработки одного и того же пакета, прибывающего по 2 разным путям.

2. Пакет должен содержать ту же метку времени клиента, как последний пакет, который был послан клиентом. Эта проверка подтверждает, что сервер отвечает на самый последний запрос клиента, а также обеспечивает некоторую защиту против фальсифицированных пакетов.

3. Сервер - достижим. Если в течение 8 последовательных интервалов опроса от сервера не было сообщений, то он считается недостижимым.

4. Рассчитывается задержка туда и обратно и дисперсия этой задержки. Если дисперсия - более, чем 16 секунд, тогда ntpd считает, что часы неточно синхронизированы. Это может случиться в некоторых чрезвычайно перегруженных сетях.

5. Должна быть реализована идентификация клиента (если она заложена в конфигурации).

Если клиент сконфигурирован на прием пакетов, содержащих биты идентификации, то пакеты, не содержащие этих битов, будут исключены из обработки во время проверки их целостности.

6. Часы сервера должны быть синхронизированы к внешнему источнику, который был модифицирован в течение последнего дня. NTP будет игнорировать пакеты от сервера, который не синхронизирован по серверу слоя «0»

(непосредственно или через дерево серверов). Ограничение одним днем позволяет серверу терять обеспечение собственной синхронизации на срок до 24 часов и все еще обеспечивать обслуживание NTP своих клиентов. Ограничение одним днем также позволяет обслуживать опорные часы. Эта операция защищает клиента от соединения с сервером, который никогда не предназначался для работы в качестве сервера времени, например, с неправильно сконфигурированной рабочей станцией.

7. Сервер должен иметь в более низкий или равный номер слоя по сравнению с клиентом.

Это предотвращает зацикливание в NTP.

8. Полная задержка от корневых часов и предельная ошибка должны быть менее 16 секунд.

Это предотвращает неточные конфигурации NTP, где существуют довольно большие задержки на многочисленных уровнях иерархии.

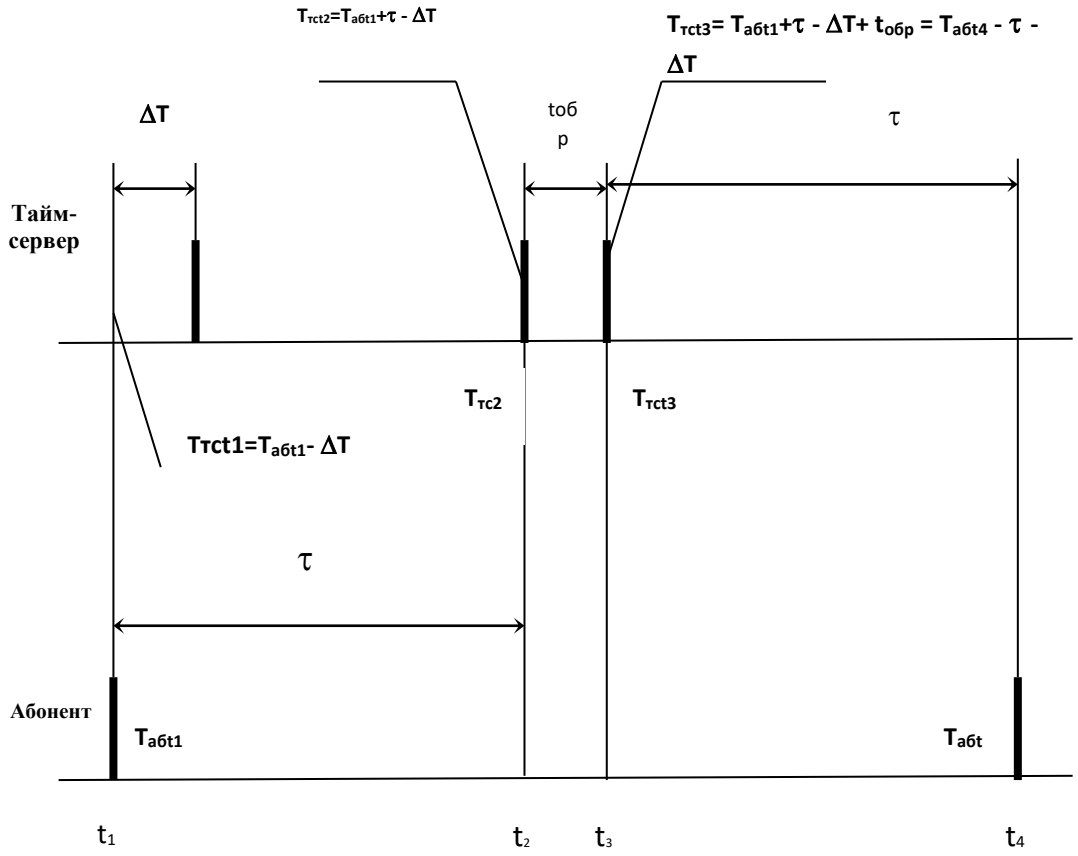
Оценка смещения и погрешности

Клиент NTP может иметь на связи до 64 серверов, но только 10 из них могут служить потенциальными источниками синхронизации. Увеличение количества серверов, с которыми клиент NTP может соединиться, хорошо по двум причинам: это позволяет установить более точное время и понижает шанс получить время от случайно рассинхронизированного или преднамеренно расконфигурированного источника времени. Эти преимущества реализованы в алгоритме кластеризации и алгоритме комбинации часов.

Принцип определения величины смещения ΔT местных часов относительно часов тайм-сервера с использованием NTP протокола поясняется диаграммой, представленной на рис. 2.

В момент времени t_1 абонент посылает в адрес тайм-сервера запрос на синхронизацию, причем в отправляемом пакете содержится информация о времени отправки запроса в шкале абонента $T_{абт1}$. Тайм-сервер получает запрос в момент времени t_2, определяемый задержкой прохождения запроса в сети τ, и регистрирует его в своей шкале времени $T_{тст2}$.

Через интервал времени $t_{обр}$, необходимый на обработку запроса, в момент времени t_3 запрос направляется обратно абоненту с информацией о времени его получения и отправки $T_{тст2}$ и $T_{тст3}$ в шкале тайм-сервера. С задержкой τ в момент времени t_4 абонент получает возвращенный запрос и регистрирует момент его прихода в своей шкале времени $T_{абт4}$.



$\Delta T_t = T_{абt} - T_{тст}$

$\tau = (T_{абt4} - T_{абt1} - t_{обр})/2$, где $t_{обр} = T_{тст3} - T_{тст2}$

$\Delta T = T_{абt1} + \tau - T_{тст2}$

$\Delta T = T_{абt4} - \tau - T_{тст3}$

$$\Delta T = \frac{(T_{абt4} - T_{тст3}) - (T_{тст2} - T_{абt1})}{2}$$

$T_{тст3} - T_{тст2} = t_{обр}$

Рис. 2

Из формул, приведенных на диаграмме, видно, что из содержащейся в полученном пакете информации можно вычислить как величину смещения ΔT часов абонента относительно часов тайм-сервера, так и время задержки в сети от абонента до тайм-сервера. Основным источником ошибки при этом является предположение, что задержки в прямом и в обратном направлении

равны, что часто не соответствует действительности.

Пример возникновения такой ошибки продемонстрирован на рис. 3.

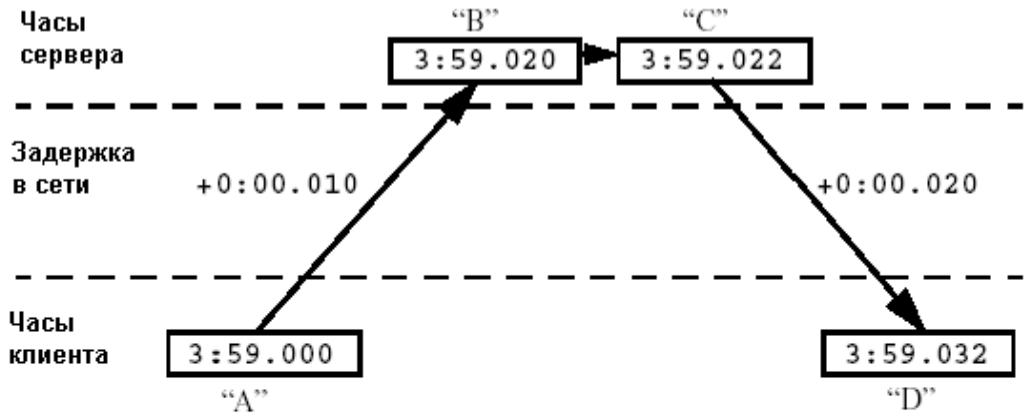


Рис. 3. Пример запроса клиент/сервер

В этом примере суммарное время задержки:

$$2\tau = (D - C) + (B - A) = 0.03 \text{ с.}$$

NTP протокол примет за время задержки $\tau = 0{,}015$ с и рассчитает смещение:

$$\Delta T = ((T_{абт4} - T_{тст3}) - (T_{тст2} - T_{абт1}))/2 = ((D - C) - (B - A))/2 = 0.005 \text{ с}$$

Действительное же смещение, как это видно из рисунка, равно 0,01 с. Ошибка, определяемая предположением равенства задержки в прямом и обратном направлениях, составляет в данном примере 0,005 с.

Фактор погрешности включает погрешности, связанные с неточностью считывания времени и ходом часов.

Фильтрация и алгоритм пересечения

Как только пакет получен, данные пакета ставятся в очередь пакетов от того же источника. Очередь содержит 8 позиций, и когда прибывает новый пакет, самый старый пакет исключается. NTP использует алгоритм фильтрации, чтобы уменьшать влияние ошибок на точность часов. Результатом работы фильтрующего алгоритма являются величины, которые наилучшим образом представляют текущее смещение и максимальную ошибку данных

часов. Поскольку очереди требуется мажоритарность для определения правильной информации, сервер недействителен для синхронизации, пока он не пошлет 5 правильных пакетов. Это требование является причиной, по которой ntpd требуется приблизительно 5 минут, чтобы синхронизироваться по новому серверу.

Как только данные будут отфильтрованы, ntpd сравнивает отфильтрованные данные с различных часов, чтобы определять "пересечение", в котором должны лежать истинные данные о времени. Алгоритм пересечения выполнен с использованием данных о смещении вместе с максимально возможной ошибкой для каждых часов. Если двое часов не укладываются в зону ошибки, то одни из них являются неверными. Если имеется группа часов, наиболее вероятно пересечение зон ошибки большинства из них. В документации NTP, часы, которые попадают в зону пересечения, названы “truechimers”, а часы вне этой зоны - "falsetickers". Рисунок 4 - диаграмма truechimers и falcetickers в алгоритме пересечения.

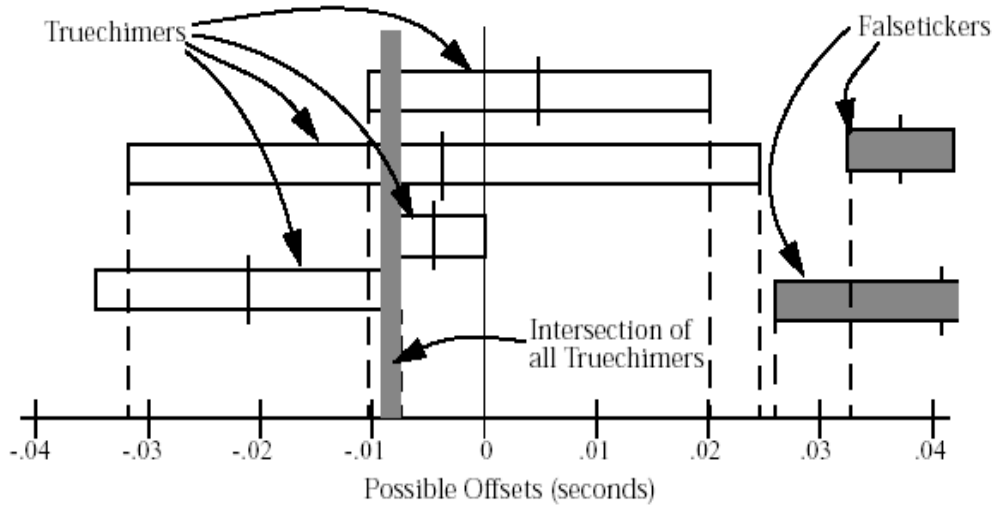


Рис. 4. Алгоритм пересечения

Если сервер - falseticker, он будет иметь знак “x” перед именем сервера при выводе списка серверов командой подсистемы ntpq .

Алгоритм кластеризации

Алгоритм кластеризации сортирует все truechimers, основываясь на взвешенной комбинации нескольких факторов, определяющих качество сер-

вера (слой, дисперсия относительно корневых часов, дисперсия по отношению к клиенту, задержка до корневых часов, задержка до клиента и, возможно, погрешность по частоте). Слой при этом выборе почти всегда доминирует. Список урезается до того момента, когда остаются только 10 "лучших" часов. Серверы, которые исключены из списка этим способом, обозначаются знаком “·” перед их именами при выводе списка серверов командой ntpq . Ниже показан вид экрана при выводе списка обрабатываемых серверов командой ntpq,

```
ntpq> peers
.ahau.mex.sun.co  matins.hours.su   4 u   88  512  377    52.28  -15.444   11.23
-imix.palenque.s  lauds.hours.sun   4 u    -  128  376    82.72   33.854    9.02
+ik.tikal.sun.co  prime.hours.sun   4 u  656  512  324    82.44  -10.320    7.34
.akbal.tulum.sun  terce.hours.sun   4 u  721 1024  377   132.80  -26.504    9.37
-kan.sayil.sun.c  sext.hours.sun.   4 u    -   64  376    72.14  -59.963    2.18
+chicchan.bonamp  nones.hours.sun   4 u    -   64  333    74.94    0.308    0.85
 cimi.seibal.sun  compline.hours.   4 u    3  256  171    76.00   -2.084 8000.53
+manik.uxmal.sun  vespers.hours.s   4 u    -   64  336    75.87    7.717    1.16
 lamat.uxmal.sun  compline.hours.   4 u    -   64    2     0.00    0.000 16000.0
 muluc.tikal.sun  0.0.0.0          16 -    -   64    0     0.00    0.000 16000.0
 oc.palenque.sun  0.0.0.0          16 -    -   64    0     0.00    0.000 16000.0
 224.0.1.1        0.0.0.0          16 -    -   64    0     0.00    0.000 16000.0
+chuen.tikal.sun  lauds.hours.sun   4 u  664 1024  377    56.21   -0.620    5.17
*eb.copan.sun.co  prime.hours.sun   3 u  639 1024  377    54.76   -1.608    3.07
xben.palenque.su  lauds.hours.sun   3 u  459 1024  367    80.84  -208.84  100.78
+ix.caracol.sun.  prime.hours.sun   3 u  682 1024  377    92.48    1.055    4.14
+men.yaxchilan.s  terce.hours.sun   4 u   28  128  352    67.50   -6.308    1.80
.cib.copan.sun.c  terce.hours.sun   4 u   52   64  174    68.45   -1.023  126.74
 caban.tulum.sun  nones.hours.sun   5 u    -   64  237    61.26   -2.177  500.79
-etznab.seibal.s  compline.hours.   4 u  602  512  372    72.02  -24.098    6.68
```

Рис. 5

Отбор часов для дальнейшей сортировки и урезания списка базируется на анализе данных о дисперсии и погрешности от каждых часов за предыдущие и текущие запросы на синхронизацию. Если текущий источник синхронизации победил в этом отборе, он автоматически остается для синхронизации, иначе в качестве нового источника синхронизации выбираются лучшие часы из списка. Источник синхронизации помечается звездочкой ” * ” или символом “ ° ” перед его именем командой ntpq . Если в пределах максимально допустимого расстояния нет никаких серверов, потенциальный источник синхронизации имеет знак “#” перед своим именем. Все остальные серверы,

которые прошли алгоритм кластеризации, имеют знак “+” перед именем сервера при выводе командой ntpq .

Комбинирование часов

Хотя в качестве источника синхронизации выбраны только одни часы, на коррекцию часов клиента влияют все оставшиеся после алгоритма кластеризации. Смещение от каждых часов взвешено в соответствии с их качеством (определенными факторами, описанными в разделе алгоритма кластеризации). Взвешенные смещения суммируются, и среднее значение используется как величина коррекции часов клиента.

Временная и частотная синхронизация системных часов

NTP используют два различных типа корректировки часов. Если разница между эталонным и текущим системным временем не превышает 128 мс, то NTP подстраивает частоту системных часов медленно и плавно. Например, для Linux стандартная скорость подстройки составляет 0,5 миллисекунды в секунду. При смещении времени системных часов клиента относительно сервера от 128 мс до 1000 с часы корректируются скачком к правильному значению. Если разница превышает 1000 секунд, то NTP полагает, что имеет место фатальный сбой времени на локальной машине и выходит из клиентской программы с соответствующим сообщением (см. рис. 6).

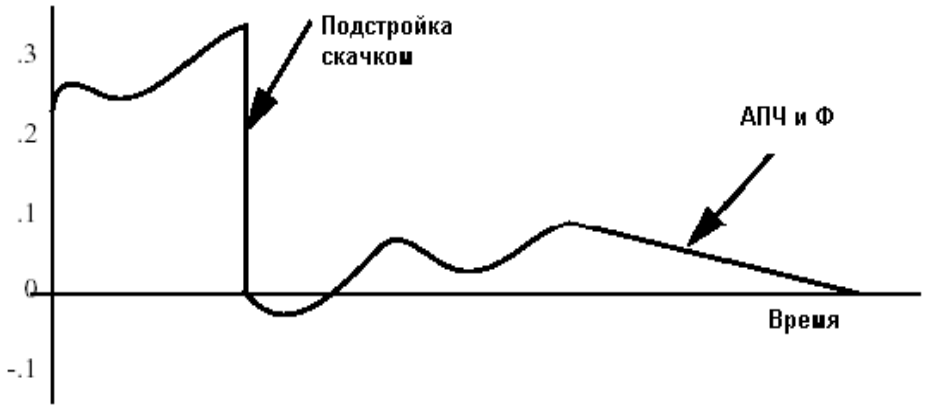


Рис. 6

Плавная подстройка по принципу подобна автоматической подстройке частоты и фазы. Структурная схема ее реализации показана на рис. 7.

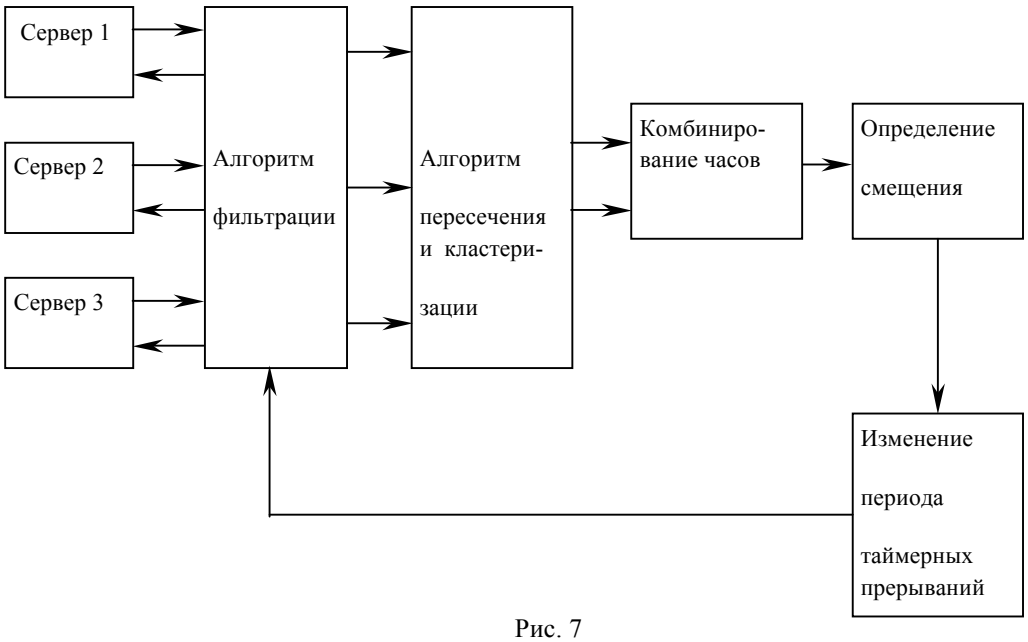


Рис. 7

Компьютер клиента NTP периодически посылает запросы на синхронизацию на несколько доступных тайм-серверов. С использованием алгоритмов фильтрации, пересечения и кластеризации определяется лучший из серверов, а алгоритм комбинирования часов дает весовые коэффициенты оставшихся. Затем вычисляется смещение системных часов клиентского компьютера. Если это смещение не превышает 128 мс, производится изменение периода таймерных прерываний, направленное на уменьшение смещения, что фактически соответствует изменению частоты системных часов. При достижении смещения, близкого к нулю, система переходит в режим фазового слежения.

Как уже упоминалось, плавная подстройка системных часов в компьютерах с операционными системами Unix или Linux осуществляется периодическим вызовом системной программы adjtime (), добавляющей фиксированную величину tickadj к периоду таймерных прерываний. Точность подстройки при этом ограничивается возможностями tickadj и определяется величиной порядка 5 мкс. При использовании NTP в качестве системных часов компьютера, формирующих расчетные значения текущего времени, исполь-

зуется 64 разрядных слова, 32 младших разряда которого используются для формирования дробной части секунд. Максимально возможное разрешение таких часов - ~200 пс и ограничивается в настоящее время только аппаратной реализацией компьютера. Точность ввода разового смещения времени таких системных часов при использовании операционных систем с наносекундными ядрами по данным, приведенным в публикациях Д. Миллса, составляет единицы наносекунд. При использовании NTP системный вызов adjtime () заменяется на более прецизионный - ntp_adjtime ().

Литература

1. David Deeths, Glenn Brunette. Using NTP to Control and Synchronize System Clocks, Copyright 2001 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303 U.S.A.
2. David L. Mills. NTP Clock Discipline Principles, http://www.eecis.udel.edu/~mills, 2003 г.
3. David L. Mills. Network Working Group, Request for Comments: 1305 «Network Time Protocol (Version 3) Specification, Implementation and Analysis», March 1992.