Time keeping in a NetWare Environment

Joe R. Doupnik
Dept of ECE
Utah State University
jrd@cc.usu.edu

Agenda

- What the PC motherboard offers for time, and what operating systems use
- Internet standard time protocol NTP
- NetWare NCP+NTP Timesync
- Suggestions for better synchronization

PC motherboard time basics

Time in software

PC clocks

Two clocks:

- **CMOS clock**
  - keeps time and date
  - Keeps machine configuration
  - Data is read when machine cold boots
  - Has its own battery (coin-sized unit)
  - Normally is set via Bios “setup” and DOS Time and Date commands, etc.

PC clocks

Run-time counter chip (hardware clock)

- 18.2 tics/sec DOS/Win, 100 tics/sec Unix, etc
- Generates an interrupt on count down to 0
- Bios counts such ticks, if the Bios is allowed to run (does so for DOS/Windows)
- Has no notion of time nor date, it is just a ticker
- Tick rate is normally fixed once the operating system starts

Operating system time

At startup the o/s consults the CMOS clock to obtain the current time and date

It captures hardware clock interrupt to maintain count of ticks since last read of CMOS clock

It offers time of day service to software. It is here that we “adjust” time. This is a software clock, driven by hardware ticks.

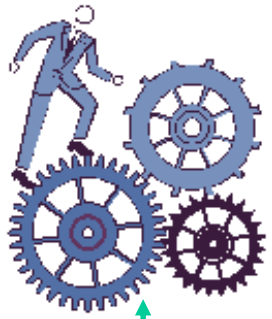
Operating system time

Software clock

Starting point

CMOS time&date

Hardware ticker



Sec/tick motor

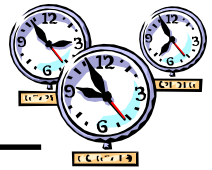
Software time

System time

Speed control: drift adjustment

smoothing filter

difference calculator



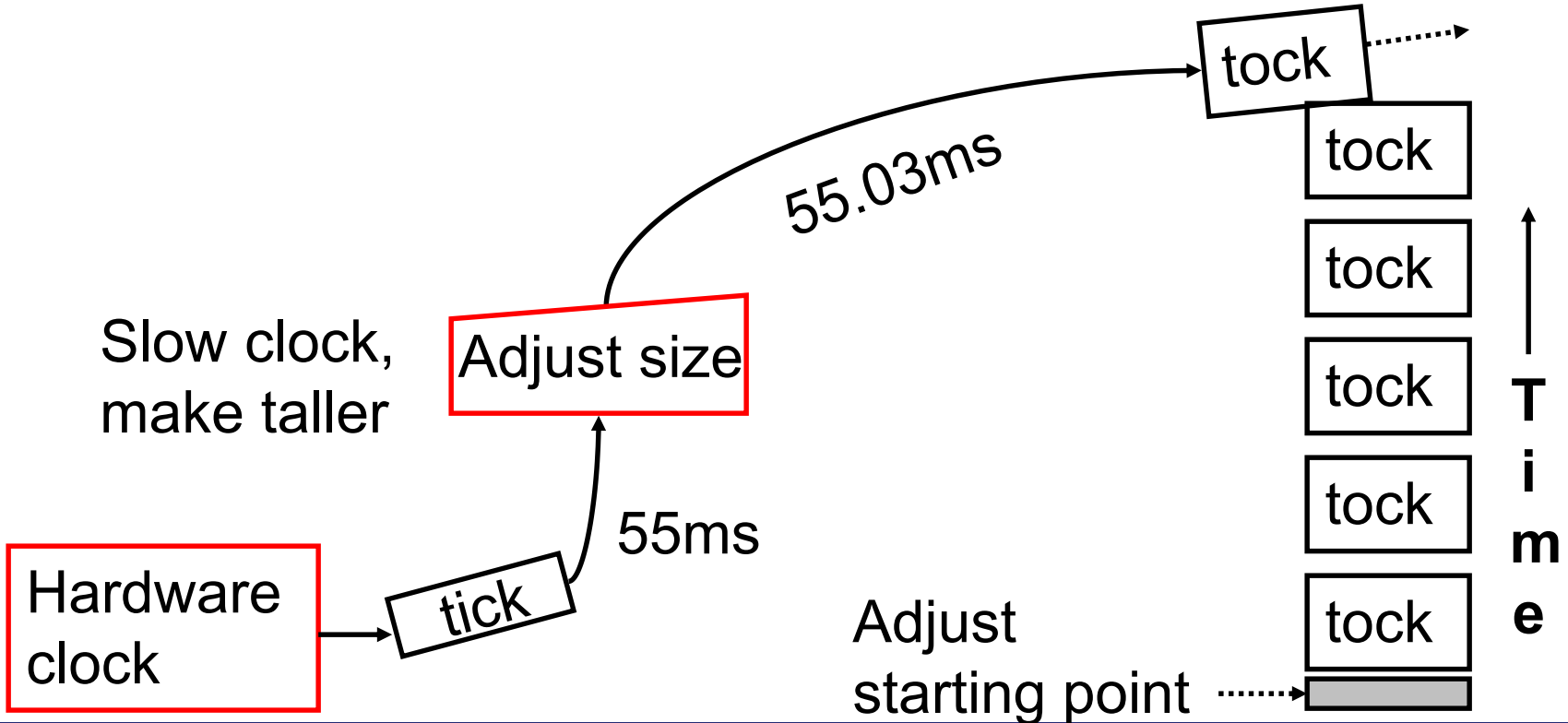
Ext sources

Operating system software clock

- **Hardware clock says when tick count changes**
- **Software computes seconds based on counts (seconds per tick)**
- **Hardware tick rate errors are absorbed into software seconds per tick scaling, thus forming a software clock**

Operating system software clock

Adjust increment of time (tock) for a hardware tick



Operating system software clock

- **Software time must never stop (long flat spots) nor go backward: time is a positive monotone**
- **Software time changes have no finer granularity than the hardware clock (something has to count at a regular rate)**
- **Adjusting seconds/tick needs to be sluggish to reduce jitter and over-compensation waves**

Operating system software clock

At system start the time and date are likely wrong, occasionally grossly so:

- **The software clock needs to be brought to the current epoch forcibly, else no sync**

During operation the hardware ticks are slightly at the wrong pace:

- **Need to scale hardware tick interval to yield proper time rate (seconds/tick)**

Network Time Protocol, NTP

Network Time Protocol

- One of the Internet's very long term use protocols
- De facto network time distribution standard
- Distributes time from many sources to many consumers, a mesh of trees
- Tree-like distribution to reduce mutual deception (rumor mongering, spanning tree routing tactics), uses NTP "stratum" notion

Network Time Protocol

Assumes time givers can be far away and the transmission paths can have much jitter

Collects data from multiple sources and chooses the most reliable as the current true-chimer (best source of UTC). Eliminates false-tickers, keeps viable alternative sources.

Choosing is a sophisticated clustering algorithm, using history of source reliability and closeness to national standards

Network Time Protocol

- **NTP time is Universal Coordinated Time, UTC1**
- **GPS uses international atomic time (TAI)**
- **UTC is for people, tied to solar conditions (86400 seconds in “a day”)**
- **TAI is absolute, and the two progressively differ (leap seconds, etc)**

Network Time Protocol

Robustness requires

- **Multiple independent sources, not one fragile link**
- **Stability in the face of jitter, delay, sources which go bad or disappear completely**
- **Avoidance of mutual and self deception**
- **Time must remain monotonic**
- **Security to defend against bad guys**

Network Time Protocol

- NTP tries to find the best source of UTC under often harrowing conditions
- It does not explicitly force a collection of local clocks to be in lock-step sync
- Very close alignment of local clocks is a consequence of robust sources of UTC
- Very close means milliseconds or better

Network Time Protocol

Version 4 is current

- full v4 is under development, RFC2030 for SNTP

Version 3 is still widely used

- subset of version 4 but lacking security, RFC1305

Version 2 and older, becoming rare

- depreciated but tolerated, RFC1119 for version 2

Novell's timesync.nlm

- is NTP version 1, RFC1059, July 1988, but omitting significant features

Network Time Protocol

"Simple" NTP programs, SNTP

- Use only one source of time
- Omit clustering analysis of sources
- Can act as client or server
- Should provide support for control packets

Simple Client NTP

- Client-only, usually single source of time
- Asks for, but does not offer time

Network Time Protocol

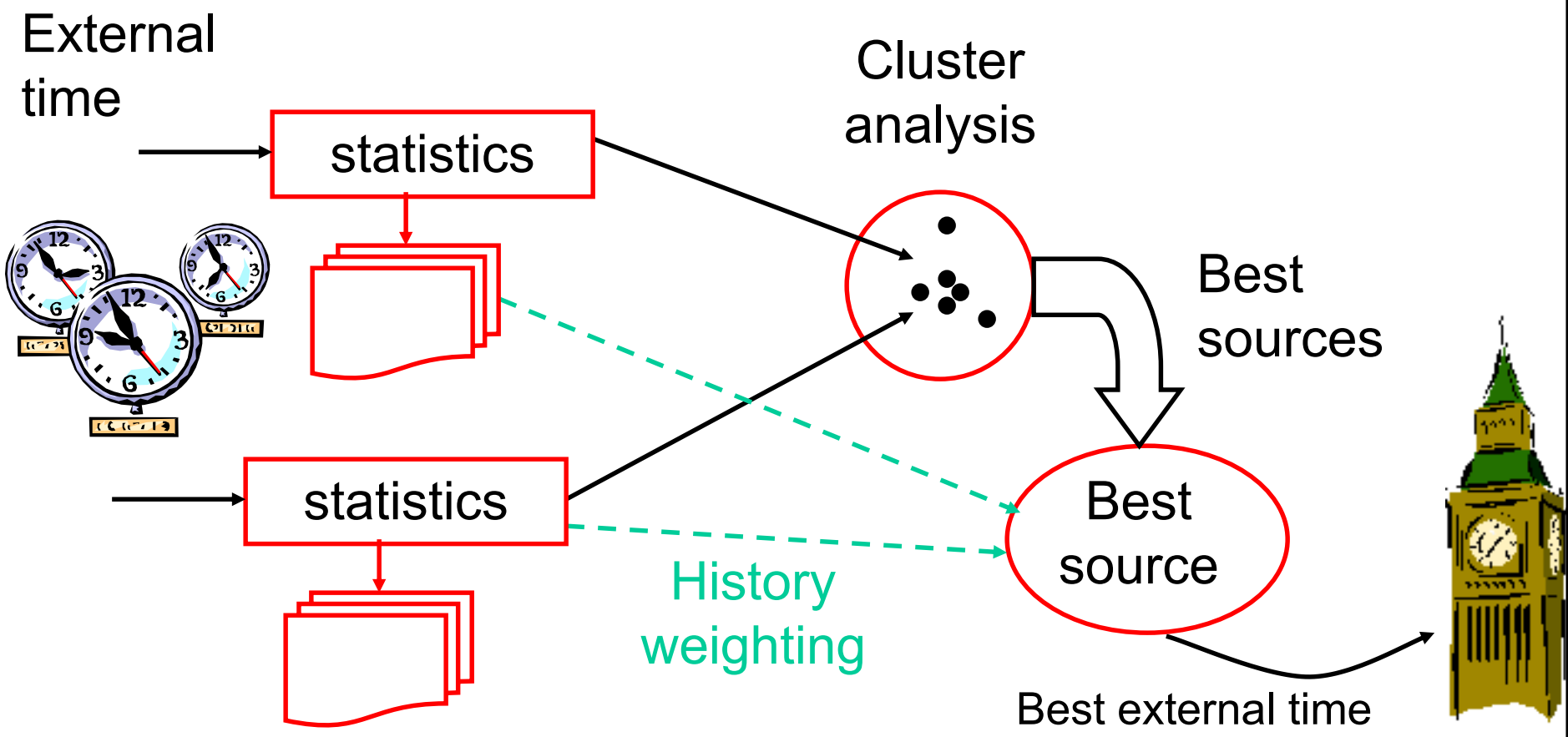
On the wire

- **NTP stations exchange simple timestamp packets (yields server's time and delay)**
- **Server is on UDP port 123**
- **Exchange rate is about three per hour when stable, faster when learning**
- **Control packets permit querying clients and servers about their time statistics, etc**

Network Time Protocol

- Can use local time sources rather than an external network
  - Pulse per second radio clocks
  - Precision oscillators
  - GPS receivers
  - Phone calls to national standards
  - Even the computer's horrid internal clock
- Can use all sources in combination

NTP selection of external time



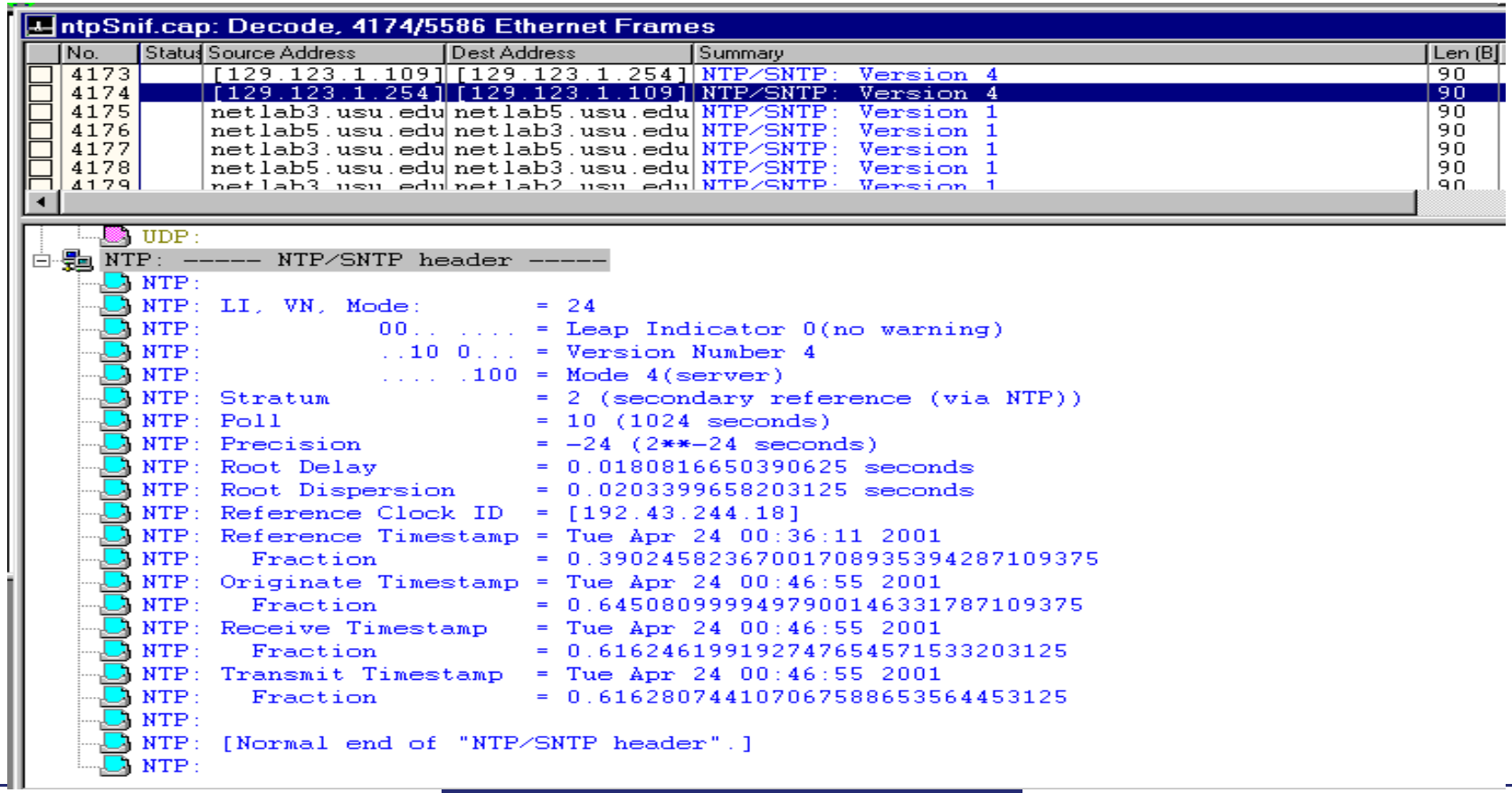
Client request, Unix to Cisco

ntpSnif.cap: Decode, 4173/5586 Ethernet Frames

No.	Status	Source Address	Dest Address	Summary	Len (B)
4173		[129.123.1.109]	[129.123.1.254]	NTP/SNTP: Version 4	90
4174		[129.123.1.254]	[129.123.1.109]	NTP/SNTP: Version 4	90
4175		netlab3.usu.edu	netlab5.usu.edu	NTP/SNTP: Version 1	90
4176		netlab5.usu.edu	netlab3.usu.edu	NTP/SNTP: Version 1	90
4177		netlab3.usu.edu	netlab5.usu.edu	NTP/SNTP: Version 1	90
4178		netlab5.usu.edu	netlab3.usu.edu	NTP/SNTP: Version 1	90
4179		netlab3.usu.edu	netlab2.usu.edu	NTP/SNTP: Version 1	90

```
UDP:
NTP: ----- NTP/SNTP header -----
NTP:
NTP: LI, VN, Mode:          = 23
NTP:          00.. ....     = Leap Indicator 0(no warning)
NTP:          ..10 0...     = Version Number 4
NTP:          .... .011     = Mode 3(client)
NTP: Stratum                = 3 (secondary reference (via NTP))
NTP: Poll                   = 10 (1024 seconds)
NTP: Precision              = -20 (2**-20 seconds)
NTP: Root Delay             = 0.0601806640625 seconds
NTP: Root Dispersion        = 0.11322021484375 seconds
NTP: Reference Clock ID     = [129.123.1.9]
NTP: Reference Timestamp    = Tue Apr 24 00:38:51 2001
NTP:   Fraction             = 0.652538817504733268965148925781 25
NTP: Originate Timestamp    = Tue Apr 24 00:29:51 2001
NTP:   Fraction             = 0.622379075764001337875366210937 5
NTP: Receive Timestamp      = Tue Apr 24 00:29:51 2001
NTP:   Fraction             = 0.654016207262662075193786621093 75
NTP: Transmit Timestamp     = Tue Apr 24 00:46:55 2001
NTP:   Fraction             = 0.645080999949790014633178710937 5
NTP:
NTP: [Normal end of "NTP/SNTP header".]
NTP:
```

Server response, Cisco to Unix



Timesync to timesync, query

ntpSnif.cap: Decode, 4175/5586 Ethernet Frames

No.	Status	Source Address	Dest Address	Summary	Len (B)
4173		[129.123.1.109]	[129.123.1.254]	NTP/SNTP: Version 4	90
4174		[129.123.1.254]	[129.123.1.109]	NTP/SNTP: Version 4	90
4175		netlab3.usu.edu	netlab5.usu.edu	NTP/SNTP: Version 1	90
4176		netlab5.usu.edu	netlab3.usu.edu	NTP/SNTP: Version 1	90
4177		netlab3.usu.edu	netlab5.usu.edu	NTP/SNTP: Version 1	90
4178		netlab5.usu.edu	netlab3.usu.edu	NTP/SNTP: Version 1	90
4179		netlab3.usu.edu	netlab2.usu.edu	NTP/SNTP: Version 1	90

```
UDP:
NTP: ----- NTP/SNTP header -----
NTP:
NTP: LI, VN, Mode:          = 0C
NTP:          00.. ....     = Leap Indicator 0(no warning)
NTP:          ..00 1...     = Version Number 1
NTP:          .... .100     = Mode 4(server)
NTP: Stratum                = 5 (secondary reference (via NTP))
NTP: Poll                   = 6 (64 seconds)
NTP: Precision              = -17 (2**-17 seconds)
NTP: Root Delay             = 0. seconds
NTP: Root Dispersion        = 0. seconds (invalid)
NTP: Reference Clock ID     = [78.79.86.76]
NTP: Reference Timestamp    = Tue Apr 24 00:46:54 2001
NTP:   Fraction             = 0.79913397948593321520843505859375
NTP: Originate Timestamp    = Tue Apr 24 00:54:53 2001
NTP:   Fraction             = 0.72819213596885861433410644531125
NTP: Receive Timestamp      = Tue Apr 24 00:46:54 2001
NTP:   Fraction             = 0.79913397948593321520843505859375
NTP: Transmit Timestamp     = Tue Apr 24 00:46:54 2001
NTP:   Fraction             = 0.79913397948593321520843505859375
NTP:
NTP: [Normal end of "NTP/SNTP header".]
NTP:
```

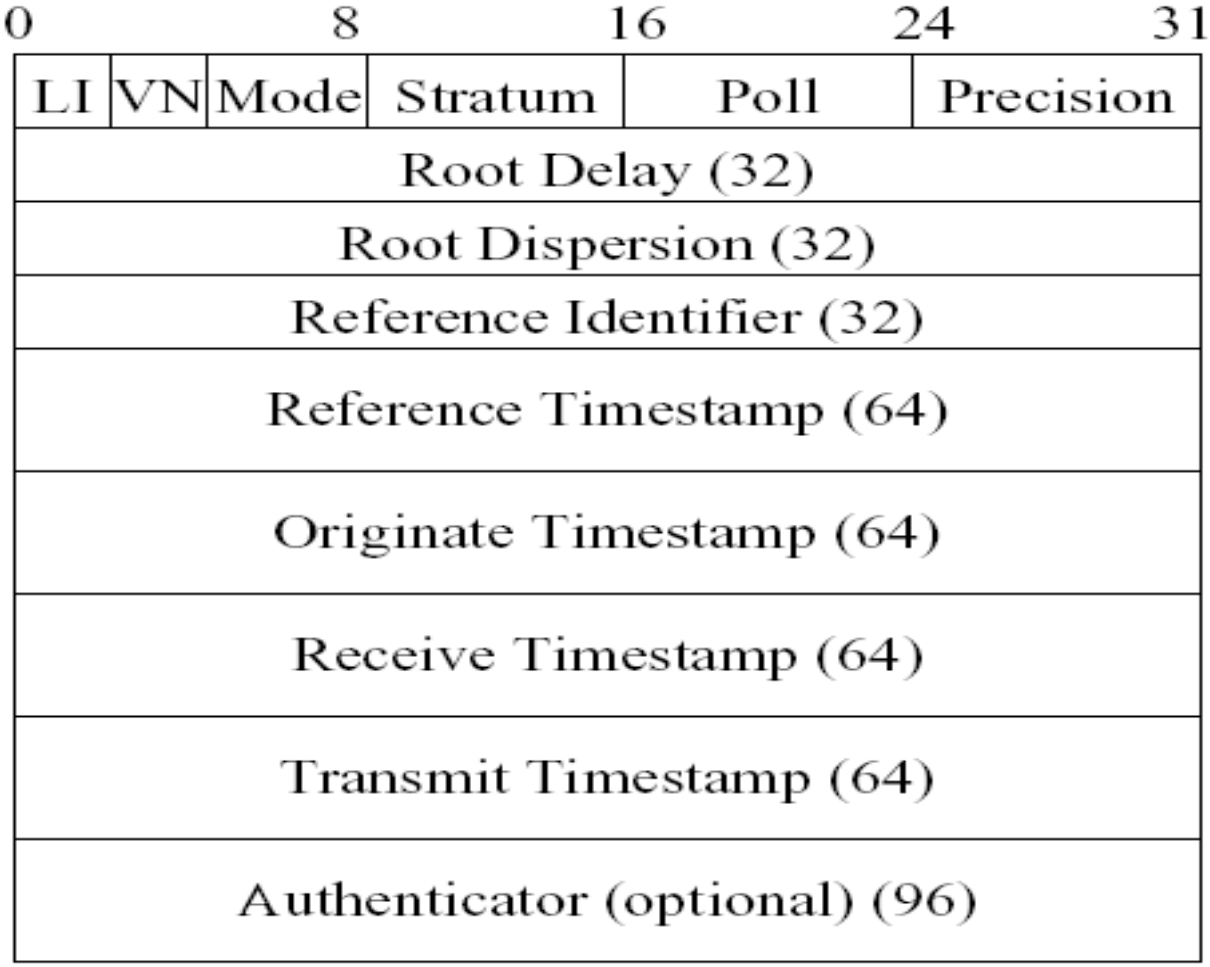
Timesync to timesync, reply

ntpSnif.cap: Decode, 4176/5586 Ethernet Frames

No.	Status	Source Address	Dest Address	Summary	Len (B)
4173		[129.123.1.109]	[129.123.1.254]	NTP/SNTP: Version 4	90
4174		[129.123.1.254]	[129.123.1.109]	NTP/SNTP: Version 4	90
4175		netlab3.usu.edu	netlab5.usu.edu	NTP/SNTP: Version 1	90
4176		netlab5.usu.edu	netlab3.usu.edu	NTP/SNTP: Version 1	90
4177		netlab3.usu.edu	netlab5.usu.edu	NTP/SNTP: Version 1	90
4178		netlab5.usu.edu	netlab3.usu.edu	NTP/SNTP: Version 1	90
4179		netlab3.usu.edu	netlab2.usu.edu	NTP/SNTP: Version 1	90

```
UDP:
NTP: ----- NTP/SNTP header -----
NTP:
NTP: LI, VN, Mode:          = 0C
NTP:          00.. ....     = Leap Indicator 0(no warning)
NTP:          ..00 1...     = Version Number 1
NTP:          .... .100     = Mode 4(server)
NTP: Stratum                = 0 (unspecified)
NTP: Poll                   = 6 (64 seconds)
NTP: Precision              = -17 (2**-17 seconds)
NTP: Root Delay             = 0. seconds
NTP: Root Dispersion        = 0. seconds (invalid)
NTP: Reference Clock ID     = (Unknown)
NTP: Reference Timestamp    = Tue Apr 24 00:54:55 2001
NTP:   Fraction             = 0.00280734810273297215118408203125
NTP: Originate Timestamp    = Tue Apr 24 00:46:54 2001
NTP:   Fraction             = 0.79913397948593321520843505859375
NTP: Receive Timestamp      = Tue Apr 24 00:54:55 2001
NTP:   Fraction             = 0.00280734810273297215118408203125
NTP: Transmit Timestamp     = Tue Apr 24 00:54:55 2001
NTP:   Fraction             = 0.00280734810273297215118408203125
NTP:
NTP: [Normal end of "NTP/SNTP header".]
NTP:
```

NTP packet format, RFC1305



NTP packet format

LI, Leap second Indicator (2 bits)

- **00: no warning**
- **01: last minute has 61 seconds**
- **10: last minute has 59 seconds**
- **11: alarm condition (clock not synchronized) This is set when machine is not time sync'd**

VN (3 bits)

- **Version number of NTP being used, server repeats what client says**

NTP packet format

Mode (3 bits)

- **0: reserved**
- **1: symmetric active (peer always sends to peers)**
- **2: symmetric passive (peer sends if reasonable)**
- **3: client**
- **4: server**
- **5: broadcast**
- **6: control message (list of “tag=value” strings)**
- **7: reserved (NTP-private, used by Unix ntpdc)**

NTP packet format

Stratum (8 bits) distance from prime authority

- **0: unspecified**
- **1: primary reference (radio clock, cesium beam, etc)**
- **2-255: secondary reference**
- **16-255: unreachable, current practice to reduce loops**

Poll (8-bits)

- **Max time to wait before re-asking, 2^{poll} seconds. Typically 1024 seconds when stable**

Precision (8-bits)

- **Of local clock, $2^{precision}$ seconds**

NTP packet format

Root delay (32 bits)

- Round trip delay seconds.fraction (16-bits each)

Root dispersion (32 bits)

- Max relative error seconds.fraction

Reference id (32 bits)

- IP of preferred time provider (was array subscript to identify radio clock kind etc)
- Novell's timesync uses non-spec "NOVL"

NTP packet format

Timestamps (timestamp format)

- **When last talked to our preferred time provider**
- **When query was sent**
- **When query was received**
- **When response was sent**
- **(When response was received)**

Time and delay

Timestamp format (64 bits)

- **32 bits of seconds (span of 136 years)**
- **32 bits of fraction of seconds (resolution of 232ps)**
- **0.0 is 00:00:00 HMS on 1 Jan 1900**
- **Rollover on 2036, a Y2.036K event; civilization ends**

NTP UDP port numbers

Well known server port is 123 (UDP)

Clients can use any port number

NTP v1 is strange, like this

Src port	Dest port	Mode
123	123	symmetric active
123	other	server
other	123	client
other	other	not possible

NTP lore

- Clocks directly attached to a machine are given pseudo-IP numbers for convenience of configuration
- These IP addresses are 127.127.t.u
  - t is the clock type (from a list of such types)
  - u is the unit number
  - 127.anything is local to only this machine.
- NTP multicast address is 224.0.1.1

NTP symmetric active, defined

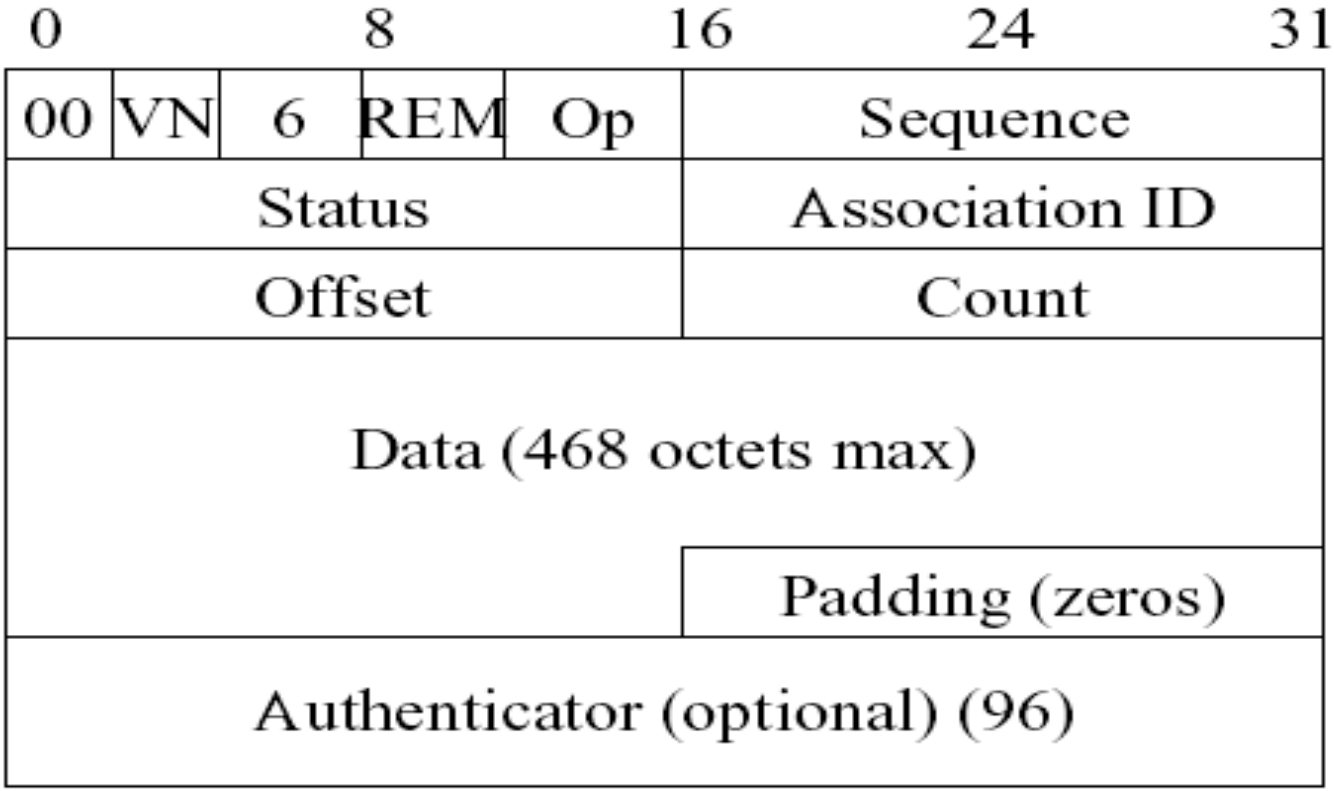
"Symmetric Active (1):

A host operating in this mode sends periodic messages regardless of the reachability state or stratum of its peer. By operating in this mode the host announces its willingness to synchronize and be synchronized by the peer."

Quote above is from RFC1305

Source and destination UDP ports are 123 for NTP v1.0, but for other versions only the destination port need be 123.

Format of control packets



NTP general topology

- NTP servers are also NTP clients, and vice versa (except for simple NTP clients)
- No special names or restrictions for machines acting as client, or server, or both
- Choose to be a client of selected servers, can be picky about serving others
- Don’t use broadcasts (naïve), be cautious about multicasts
- Use a mesh of sources for robustness

Picking and choosing sources

NTP uses sophisticated clustering analysis and filtering techniques to choose a subset of sources which are presumed to be accurate. No voting. One source is adopted as master.

Discard but remember outlying sources

Dispersion and stratum numbers are weights

Path and processing delays, clock granularity are included.

Smoothed results are less noisy that local clock steps, long term accuracy is very good

Picking and choosing sources

- **The complete path from here to the fundamental time sources can be traced and the dispersion etc at each step can be obtained**
- **Unix program ntptrace does the tracing**
- **Unix program ntpq does host querying**
- **Unix program ntpd is the NTP daemon**
- **Unix program ntpdc is daemon control**
- **Unix program ntpdate gets & forces time+date**
  - **Same as Rdate.nlm by Brad Clements**

Getting started on the right foot

- **NTP has a capture window (a "capture basket") within which clocks will track each other. It is normally 1000 seconds, configurable.**
- **Force initial time to be near center of the capture basket (within seconds)**
  - **NTP v4 software can do automatic forcing at server startup**

Getting started on the right foot

- **Use ntpdate or equivalent to initially force both time AND date to proper values**
- **At startup use previous "drift file" to correct hardware clock's rate and hence accelerate synchronization**
- **Frequency tolerance of hardware clock:**
  - **100 ppm (V1-3), 500 parts per million (V4)**
  - **100 ppm is an error of 8.64 seconds/day**

Client-server or peer to peer?

Client-server is the overall best approach

- **Several servers to a client for robustness**
- **Areas of a site use other areas as sources**

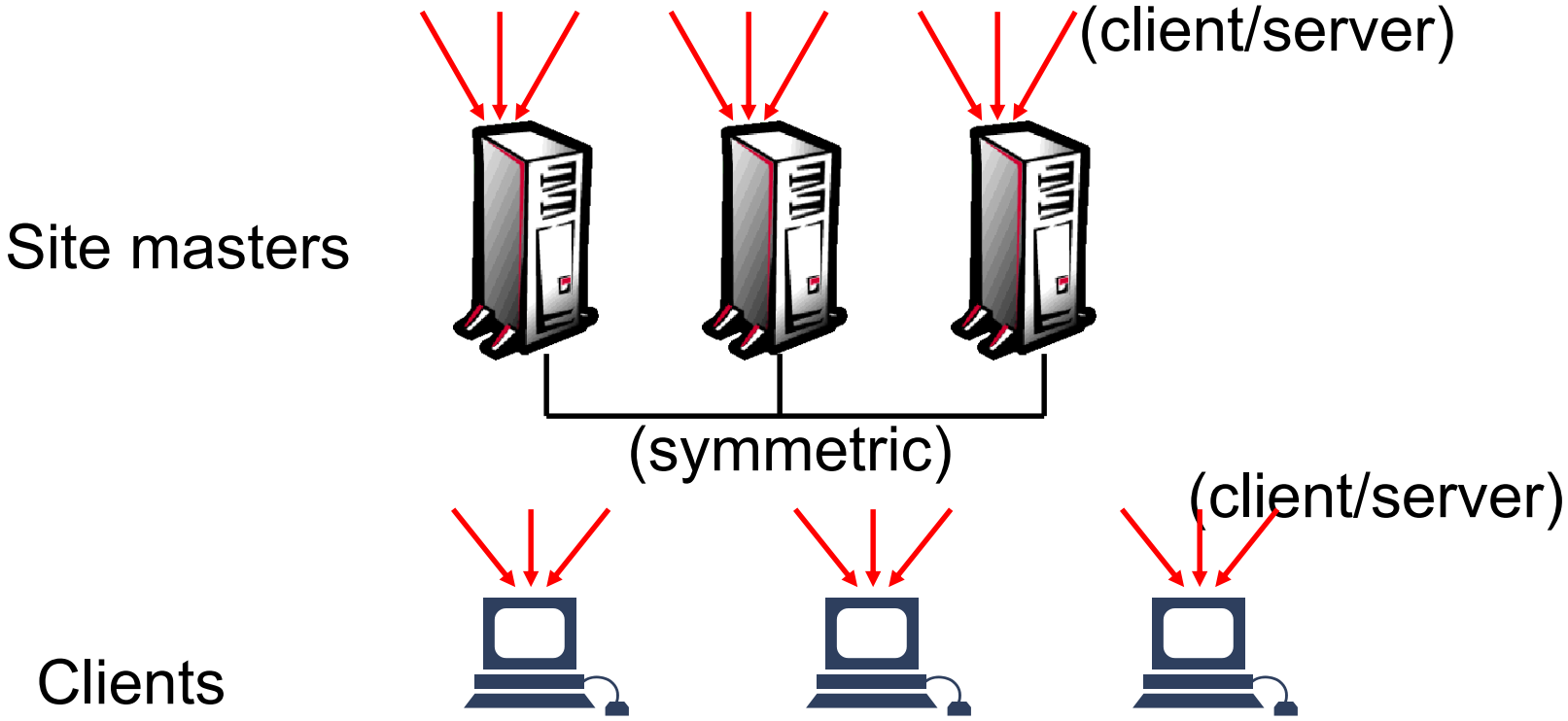
Peer to peer

- **Shares time between equals, for site robustness and load sharing, recommended**

Proper servers & clients can be both at once

A normal medium site layout

Internet. At least 3 sources, minimal overlap



Choosing a server plan

Best approach for a site is

- **Use several local servers as site masters**
- **Each site master server uses a spread of time givers slightly overlapping other masters**
- **Each client talks to several site servers**
- **Robustness involves redundancy and avoidance of single points of failure**
- **Avoid national standards if other sites will offer service (spreads the load). Do ask.**

Choosing a server plan

Tinkering with sources

- **Don’t believe your internal clock as an external source! See next point.**
- **If external time sources vanish NTP will leave the local clock unsynchronized. Keep that drift file.**
- **We can force the stratum value of external sources, if necessary. Most useful if external sources are of primary kind (radio, etc).**
- **We can use internal clock at high numbered stratum to assist clients when all external sources vanish. Suggest using stratum 10 or more for internal clock.**

Choosing a server plan

For NetWare servers, and multitasking machines in general, it is best to avoid Pulse Per Second radio clocks attached through the serial port. These require serial port software with little delay and small jitter.

Better is to use today's external clocks providing NTP service over lans

Unix "ntpq" to query servers

```
$ ntpq
ntpq> lpeer
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+ser2-FE3-11-12. 132.163.135.130  2 u  371 1024  377    46.812    9.236  11.325
+tock.usno.navy. .USNO.           1 u  487 1024  377    61.162    3.732   9.737
*time.nist.gov   .ACTS.           1 u  493 1024  377    12.059   -2.419   8.771
```

milliseconds

remote = server

refid = IP, or NTP index

st = stratum of source

t = up (or reachable)

when = seconds since last poll

poll = seconds between polls

reach = bit-field of last 8 polls, octal, 1=OK

* best source

+ viable alternate

- discarded

Unix "ntpq" queries servers

```
$ ntpq
ntpq> lpeer
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+ser2-FE3-11-12. 132.163.135.130  2 u  371 1024  377   46.812    9.236  11.325
+tock.usno.navy. .USNO.           1 u  487 1024  377   61.162    3.732   9.737
*time.nist.gov   .ACTS.           1 u  493 1024  377   12.059   -2.419   8.771
ntpq> host 129.123.1.254
current host set to 129.123.1.254
ntpq> lpeer
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+ntp1.usno.navy. .USNO.           1 u   18 1024  377   60.700   27.513  29.680
+ntp0.usno.navy. .USNO.           1 u  768 1024  367  107.850    2.785  15.140
-132.163.135.131 .ACTS.           1 u  288 1024  377   15.750   35.112  24.350
-132.163.135.130 .ACTS.           1 u  986 1024  377  129.460  -29.155  23.790
*time.nist.gov   .ACTS.           1 u  756 1024  377   84.030   -6.437   6.290
 main-Vl100.gw.u CHU_AUDIO(1)     2 u   9d 1024    0    7.250   -1.523 16000.0
```

Stats about sources

Query another host

(9 days!)

NetWare, timesync.nlm

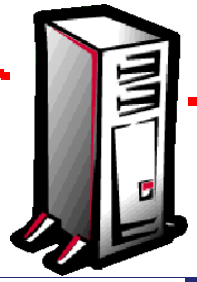
NetWare uses of time

- **Time is needed for file date stamps, workstations, other ordinary things**
- **Time is vital to NDS so that events are labeled in order of occurrence (without negotiating a tree-wide unique sequence number)**
- **Loosely consistent NDS data needs time synchronized amongst servers within a few seconds**

NetWare view of time

The Internet → Tenuous use of NTP version 1

NetWare servers exchange time using only NetWare Core Protocol packets. Main source of time is PC motherboard



NetWare time, 2 servers + world

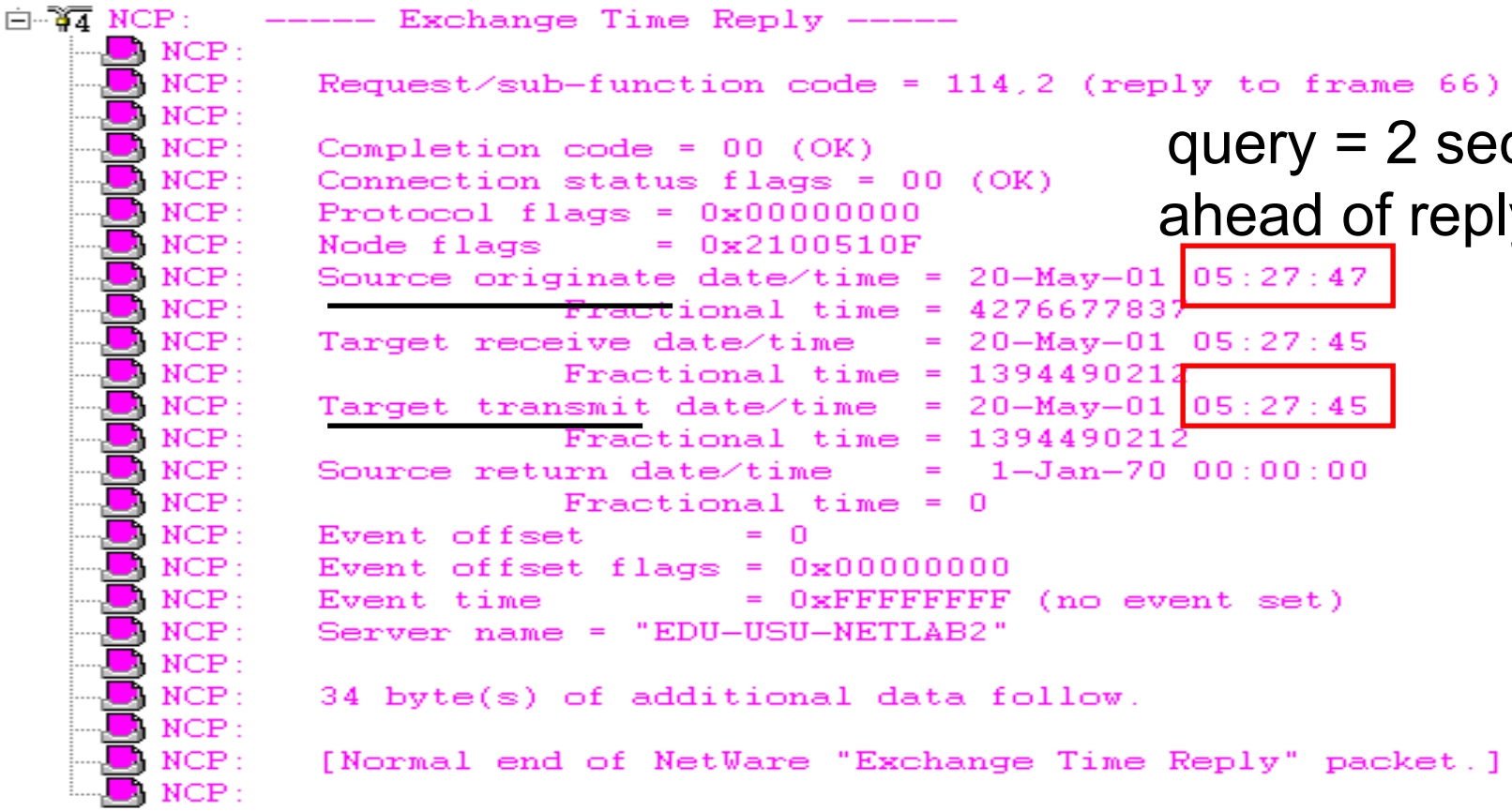
No.	Status	Source Address	Dest Address	Summary	Len (B)	Rel. Time	Delta Time	
52		netlab2.usu.edu	netlab1	NTP/SNTP: Version 1	90	0:00:45.553	8.064.044	NTP 123:123
53		netlab1	netlab2.usu.edu	NTP/SNTP: Version 1	90	0:00:45.553	0.000.265	
54		netlab2.usu.edu	netlab1	NTP/SNTP: Version 1	90	0:00:45.553	0.000.229	
55		netlab1	netlab2.usu.edu	NTP/SNTP: Version 1	90	0:00:45.554	0.000.773	
56		netlab2.usu.edu	netlab1	NTP/SNTP: Version 1	90	0:00:45.615	0.060.494	
57		netlab1	netlab2.usu.edu	NTP/SNTP: Version 1	90	0:00:45.615	0.000.156	
58		netlab2.usu.edu	[129.123.1.254]	NTP/SNTP: Version 1	90	0:00:45.677	0.061.712	
59		[129.123.1.254]	netlab2.usu.edu	NTP/SNTP: Version 1	90	0:00:45.678	0.001.080	
60		netlab2.usu.edu	[129.123.1.254]	NTP/SNTP: Version 1	90	0:00:45.738	0.060.627	
61		[129.123.1.254]	netlab2.usu.edu	NTP/SNTP: Version 1	90	0:00:45.740	0.001.399	
62		netlab2.usu.edu	[129.123.1.254]	NTP/SNTP: Version 1	90	0:00:45.800	0.060.381	
63		[129.123.1.254]	netlab2.usu.edu	NTP/SNTP: Version 1	90	0:00:45.814	0.014.161	
64		817B0100.009027	817B0100.0060B0	NSAP: R EDU-USU-NETLAB2	110	0:00:46.500	0.685.504	
65		817B0100.009027	817B0100.080009	NSAP: R EDU-USU-NETLAB2	110	0:00:46.955	0.455.386	
66		netlab5.usu.edu	netlab2.usu.edu	NCP: C Exchange time	154	0:00:48.202	1.246.983	NCP
67		netlab2.usu.edu	netlab5.usu.edu	NCP: R OK	156	0:00:48.202	0.000.217	
68		netlab5.usu.edu	netlab2.usu.edu	NCP: C Exchange time	154	0:00:48.203	0.000.485	
69		netlab2.usu.edu	netlab5.usu.edu	NCP: R OK	156	0:00:48.203	0.000.178	
70		netlab5.usu.edu	netlab2.usu.edu	NCP: C Exchange time	154	0:00:48.203	0.000.446	
71		netlab2.usu.edu	netlab5.usu.edu	NCP: R OK	156	0:00:48.204	0.000.178	
72		817B0100.009027	817B0100.080009	NSAP: R EDU-USU-NETLAB2	110	0:00:48.211	0.007.021	
73		netlab5.usu.edu	netlab2.usu.edu	NCP: C Exchange time	154	0:00:49.191	0.980.036	NCP
74		netlab2.usu.edu	netlab5.usu.edu	NCP: R OK	156	0:00:49.191	0.000.232	
75		netlab5.usu.edu	netlab2.usu.edu	NCP: C Exchange time	154	0:00:49.191	0.000.470	
76		netlab2.usu.edu	netlab5.usu.edu	NCP: R OK	156	0:00:49.192	0.000.183	
77		netlab5.usu.edu	netlab2.usu.edu	NCP: C Exchange time	154	0:00:49.192	0.000.460	
78		netlab2.usu.edu	netlab5.usu.edu	NCP: R OK	156	0:00:49.192	0.000.174	
79		817B0100.009027	817B0100.080009	NSAP: R EDU-USU-NETLAB2	110	0:00:49.468	0.275.909	
80		817B0100.009027	817B0100.080009	NSAP: R EDU-USU-NETLAB2	110	0:00:50.727	1.258.843	
81		netlab5.usu.edu	netlab2.usu.edu	NCP: C Exchange time	154	0:00:59.080	8.353.445	NCP
82		netlab2.usu.edu	netlab5.usu.edu	NCP: R OK	156	0:00:59.081	0.000.232	
83		netlab5.usu.edu	netlab2.usu.edu	NCP: C Exchange time	154	0:00:59.084	0.003.522	
84		netlab2.usu.edu	netlab5.usu.edu	NCP: R OK	156	0:00:59.084	0.000.183	
85		netlab5.usu.edu	netlab2.usu.edu	NCP: C Exchange time	154	0:00:59.085	0.000.491	
86		netlab2.usu.edu	netlab5.usu.edu	NCP: R OK	156	0:00:59.085	0.000.174	

15 seconds of an active sync interval

NCP time (code 114) request

```
NCP:    ----- Exchange Time Request -----
NCP:
NCP:    Request/sub-function code = 114,2
NCP:
NCP:    Protocol flags = 0x00000000
NCP:    Node flags     = 0x21000000
NCP:    Source originate time/date = 20-May-01 05:27:47
NCP:                 Fractional time = 4276677837
NCP:    Target receive date/time   =  1-Jan-70 00:00:00
NCP:                 Fractional time = 0
NCP:    Target transmit date/time  =  1-Jan-70 00:00:00
NCP:                 Fractional time = 0
NCP:    Source return date/time    =  1-Jan-70 00:00:00
NCP:                 Fractional time = 0
NCP:    Event offset        = 0
NCP:    Event offset flags  = 0x00000000
NCP:    Event time          = 0xFFFFFFFF (no event set)
NCP:    Server name = "EDU-USU-NETLAB5"
NCP:
NCP:    34 byte(s) of additional data follow.
NCP:
NCP:    [Normal end of NetWare "Exchange Time Request" packet.]
NCP:
```

NCP time reply



query = 2 sec ahead of reply

TID 10057202 *If it hurts, don't do that*

Symptom

Server time out of sync with the Network

Client displays the time which is the same as the time from the Server BIOS

Change

Server time is in sync with the Network the correct time is displayed on the client

Cause

Server.nlm determines which time to give to the client asking via Timesync Get Time NCP 114 01.
Timesync sets a bit on the server depending if the server is in sync with the network

The server determines which event offset value to send back for Daylight Saving Time

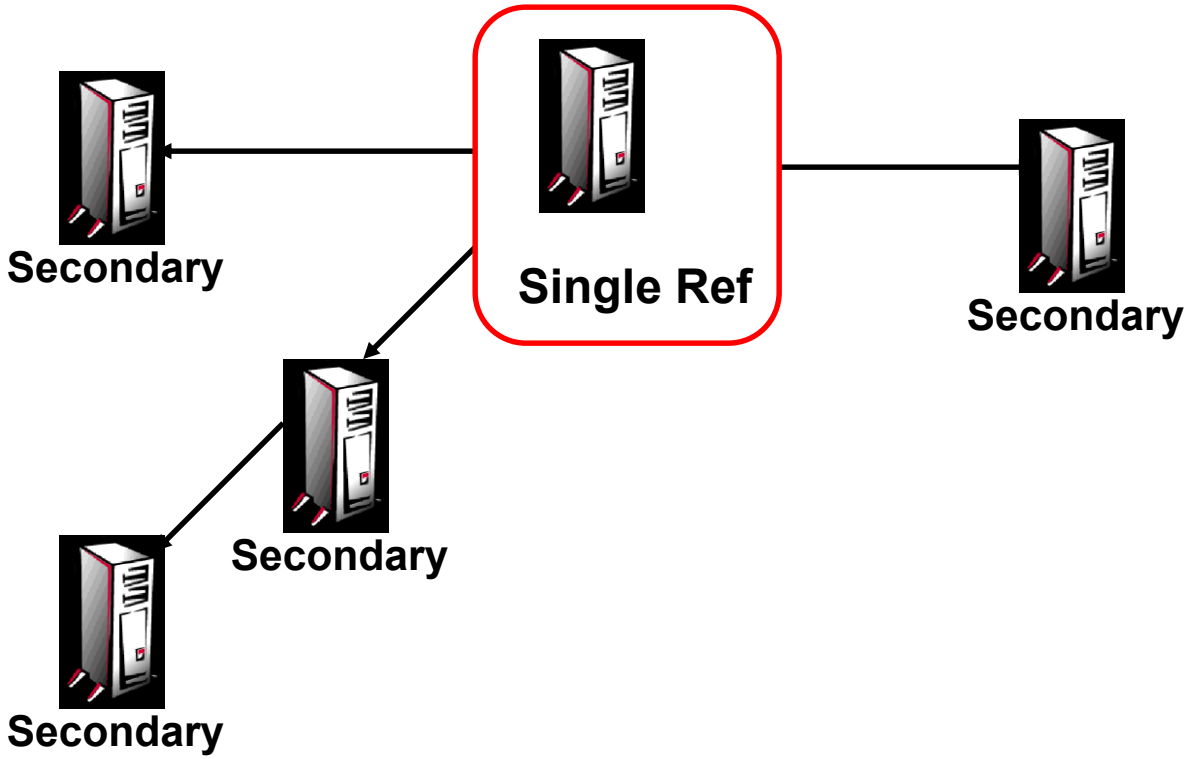
The server sends back the BIOS time if the server is out of sync with no DST adjustments

For the client it only matters what the server sends back for the NCP 114 01 reply

Fix

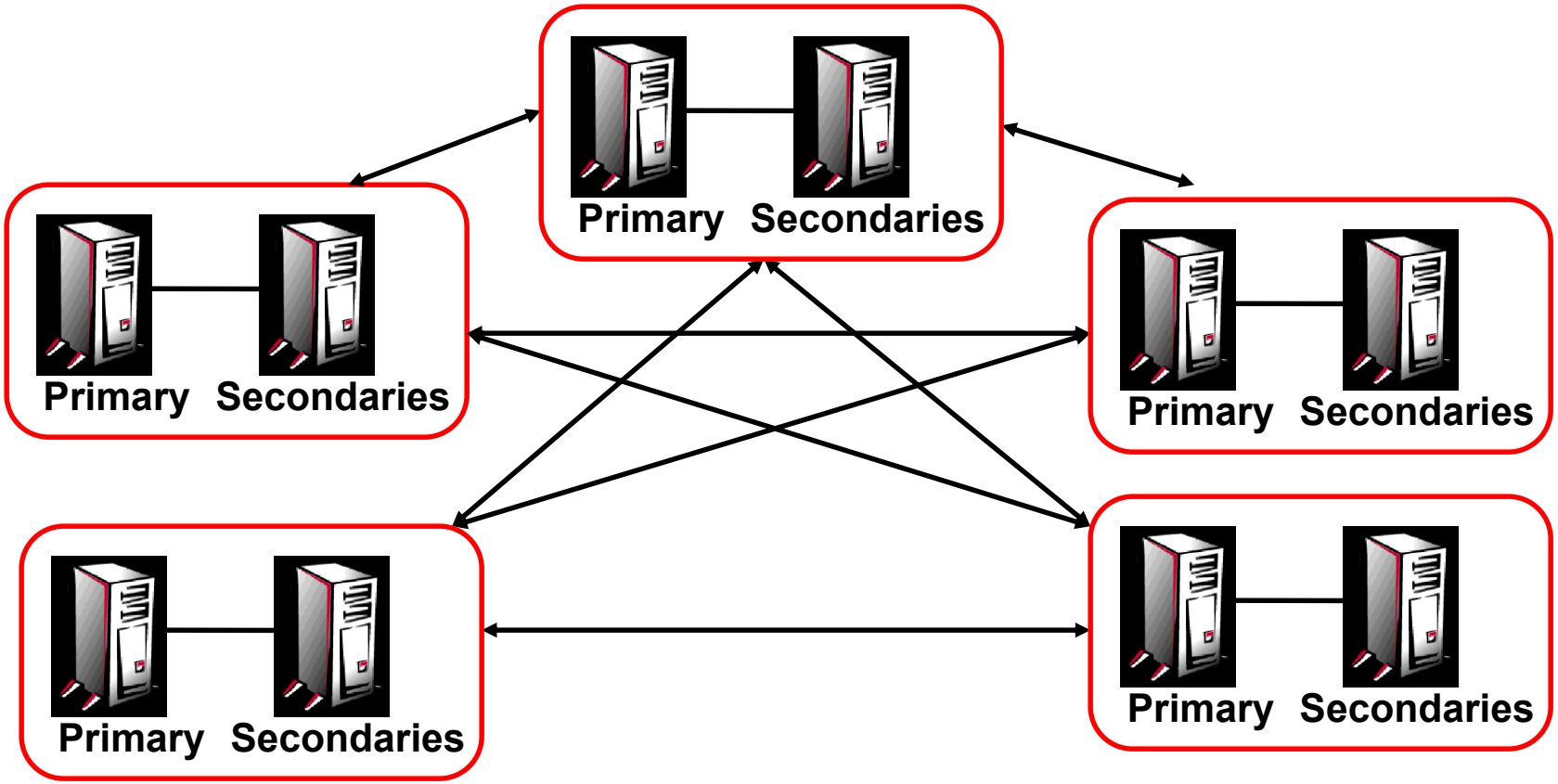
Keep time in sync with the one on the network

NW 5 docs, SINGLE Reference



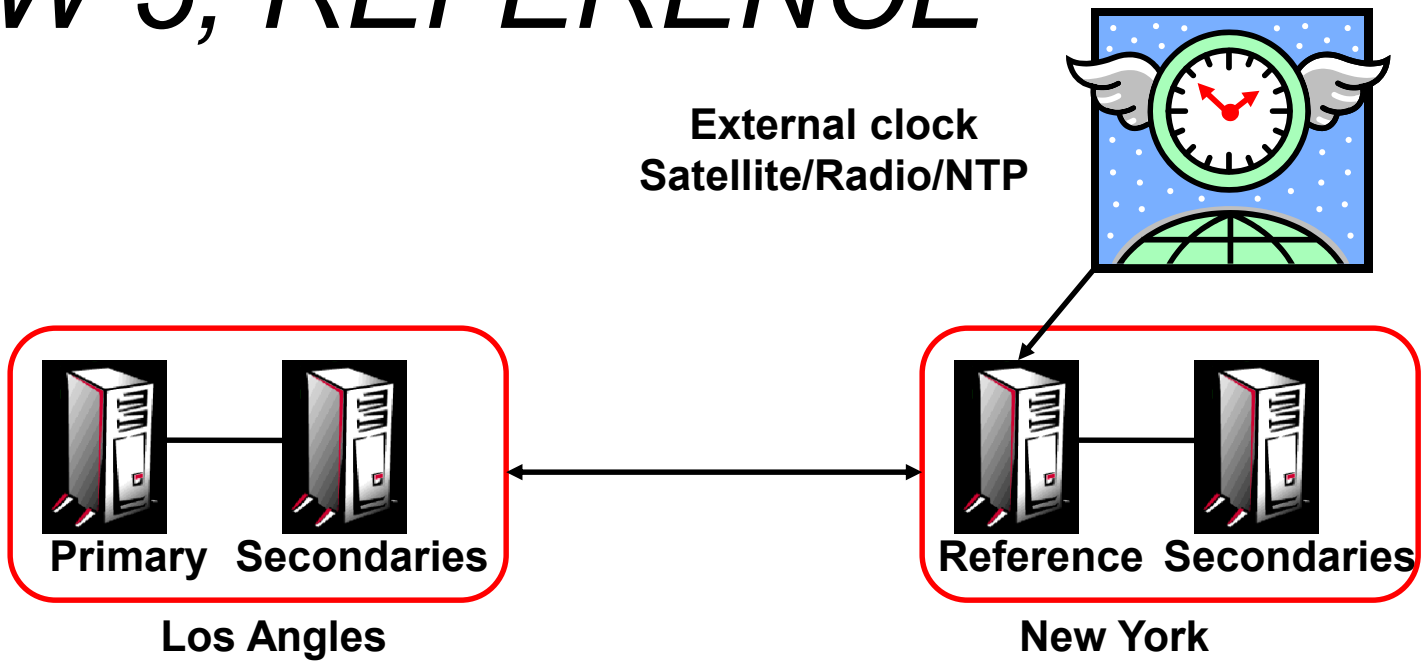
IMPORTANT: *If you use a single reference time server, don't use any primary or reference time servers on the network.*

NW 5, PRIMARY + SECONDARIES



Use the primary time server on larger networks to increase fault tolerance by providing redundant paths for secondary time servers.

NW 5, REFERENCE

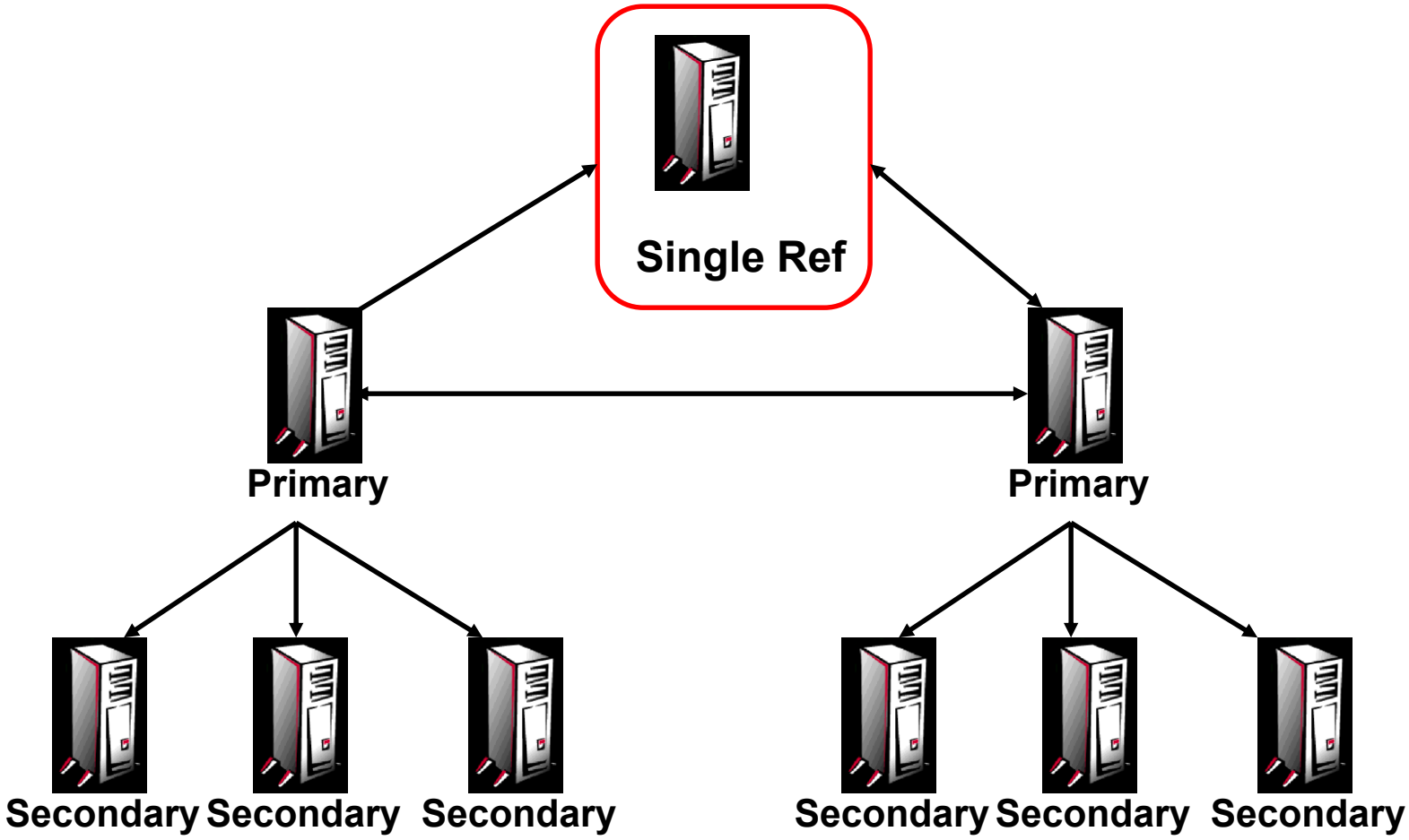


Use a reference time server when you need a central point to control time on the network. Usually only one reference time server is installed on a network.

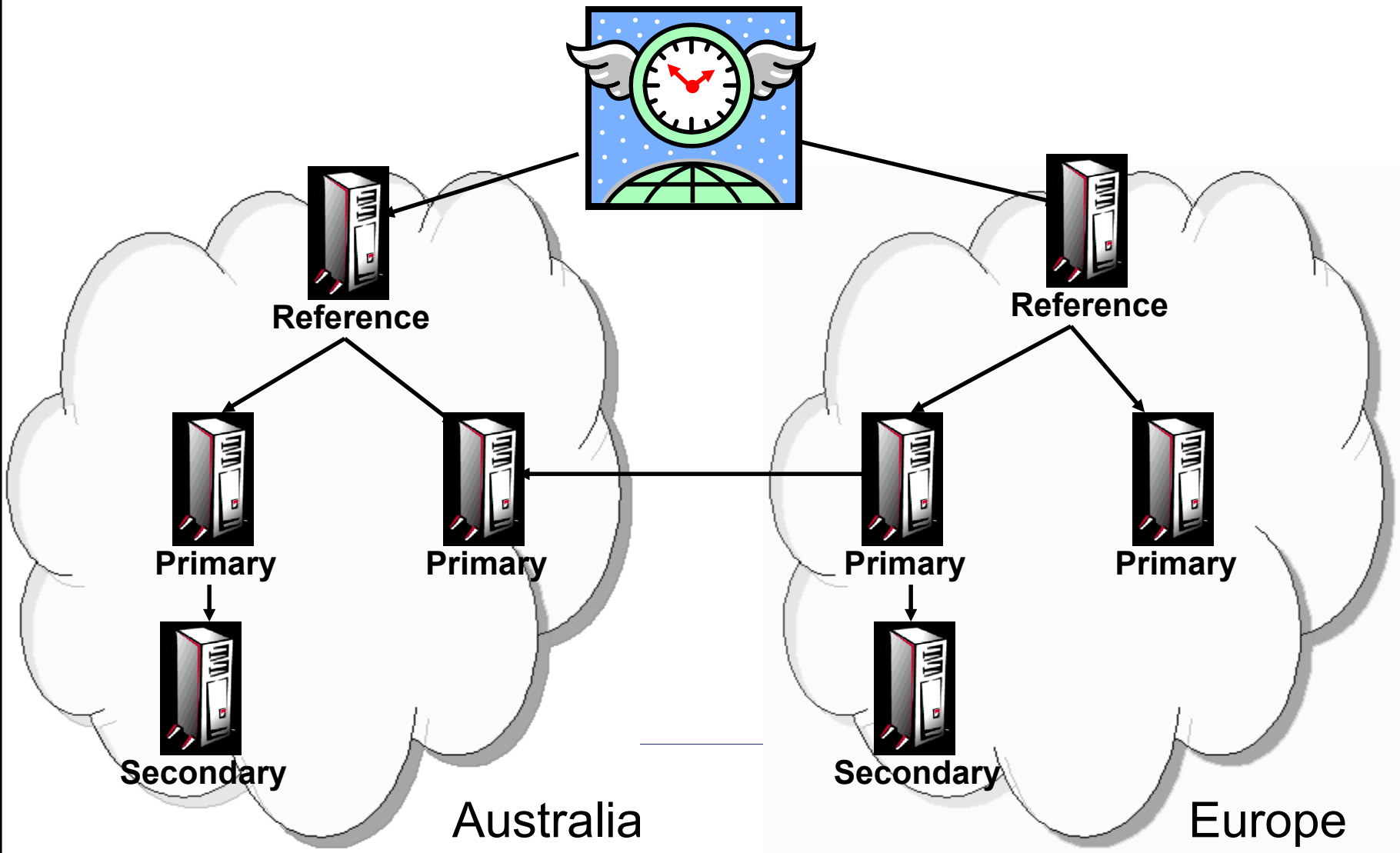
You can use more than one reference time server on a network, but you must synchronize each reference server with an external time source, such as a radio clock

IMPORTANT: You must have at least one other primary time server that the reference time server can contact. Whenever primary and reference time servers are on the network, they must be able to contact each other for polling.

NW 5, SINGLE+PRIMARIES



NW 5, multiple REFERENCES



Timesync "layers" versus kinds

1	Single	Reference(s)
2	<none>	Primaries
3	Primary or Secs	Secondaries
4…	Secondaries	

Single and Refs can talk externally.

Primaries are peer to peer time voting machines when there is more than one source of external time (Ref's).

NW over-specialization puzzle

SINGLE (Single Reference)

- **Actually it speaks NTP v1 to read external time**
- **Reads the hardware clock for every time sample**
  - We think it reads only one or the other
- **Offers time to PRIMARYs and SECONDARYs**
- **Does not read time from them**

REFERENCE

- **Reads time from outside (NTP v1)**
- **Reads the hardware clock for every time sample**
  - We think it reads only one or the other
- **Offers time only to PRIMARYs**

NW over-specialization puzzle

PRIMARY

- **Reads time from SINGLE/REFs or internal clock**
- **Coordinates (votes) time with other PRIMARYs**
- **Offers time to SECONDARYs**
- **Writes the hardware clock for every time sample**
  - We think it should not both read and write its internal clock

SECONDARY

- **Reads time from SINGLE and PRIMARYs**
- **Does not read time from REFERENCEs**
- **Writes the hardware clock for every time sample**

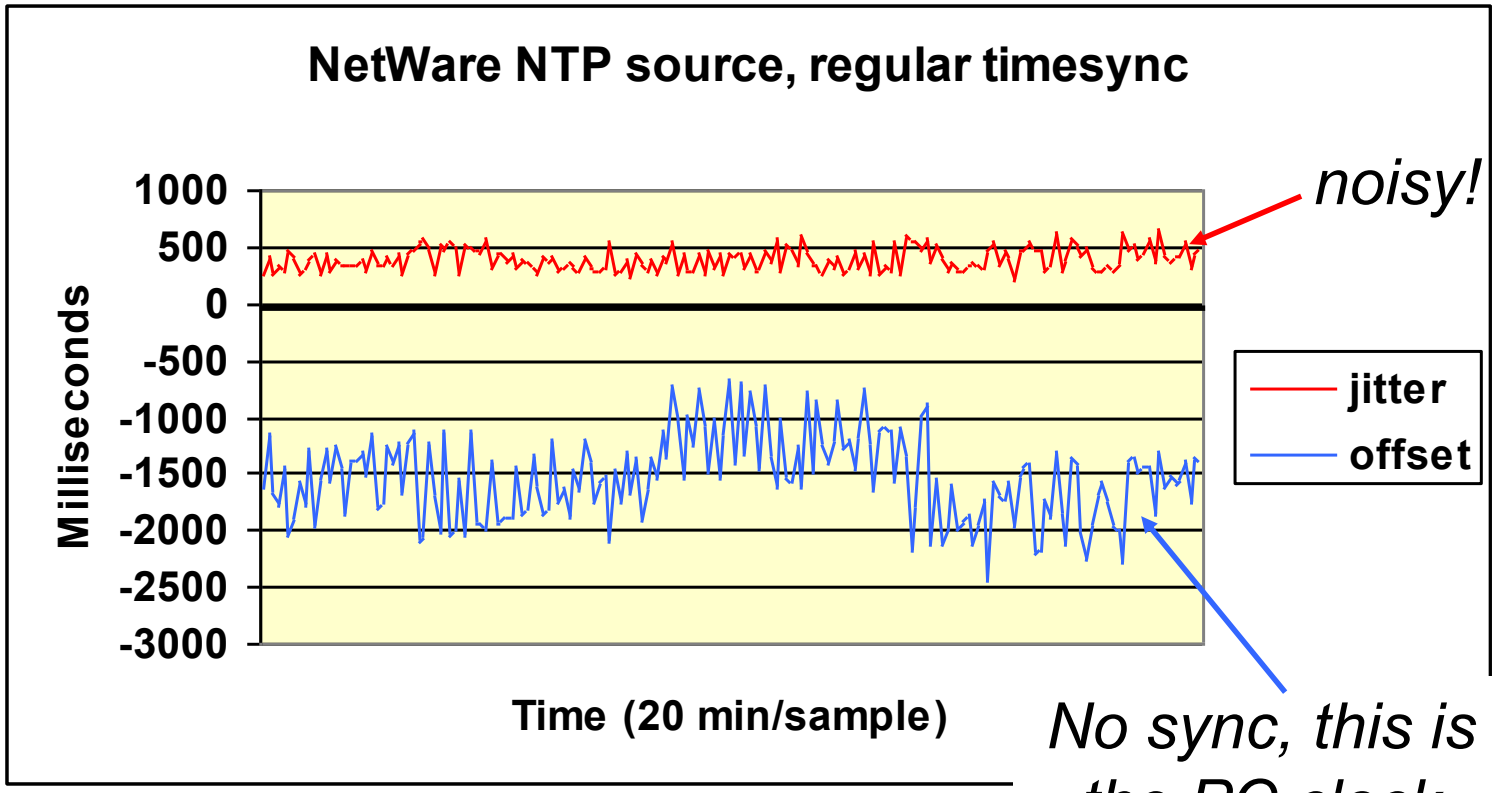
NW over-specialization puzzle

Writing the hardware clock should be done rarely, say at server start, and only if there is a remote reliable source of time

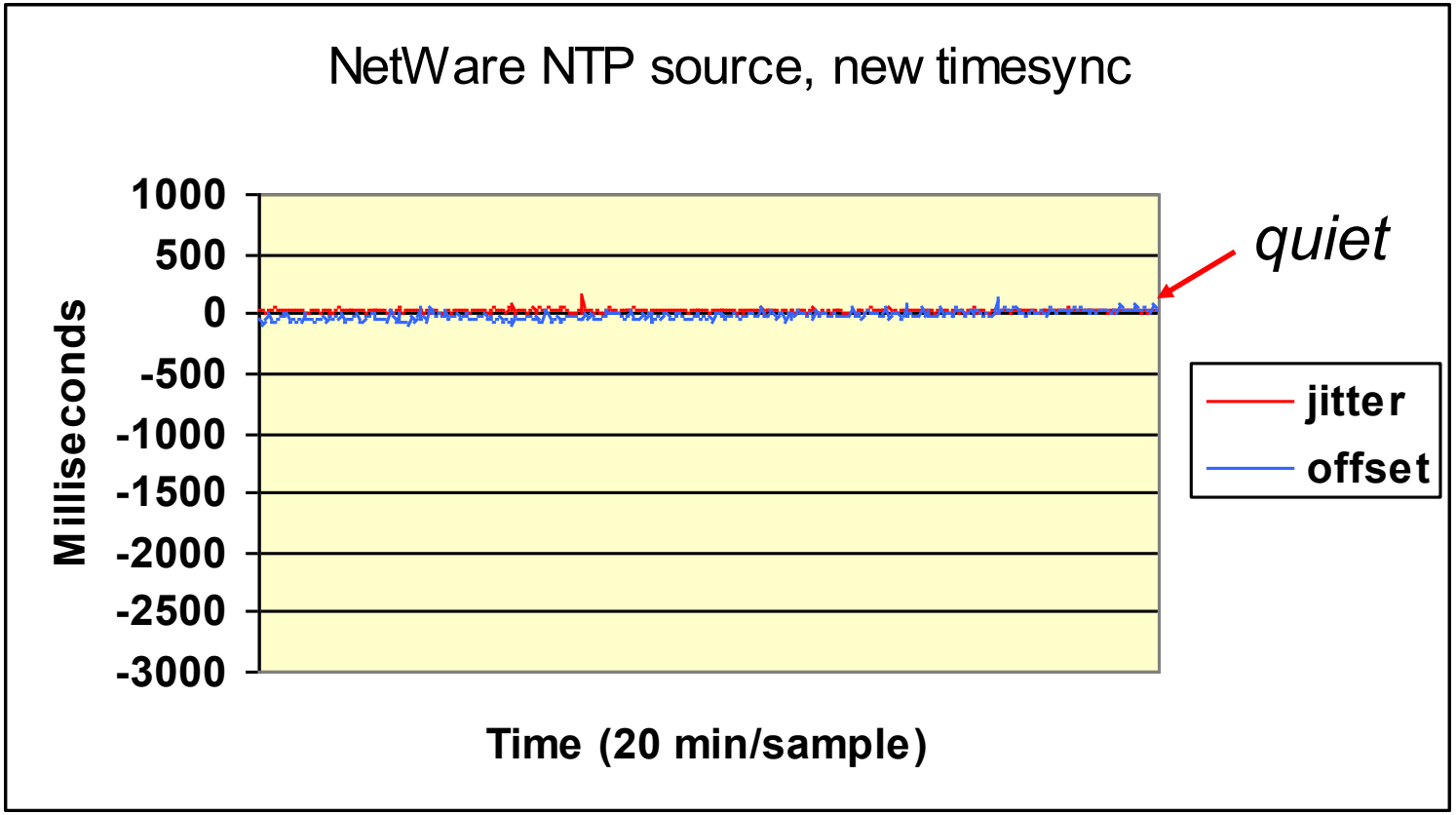
Reading the "hardware clock" is unnecessary unless there is no source of network time

"Hardware clock" in Novell terms means the PC CMOS clock which has time and date information (real hardware clock has neither)

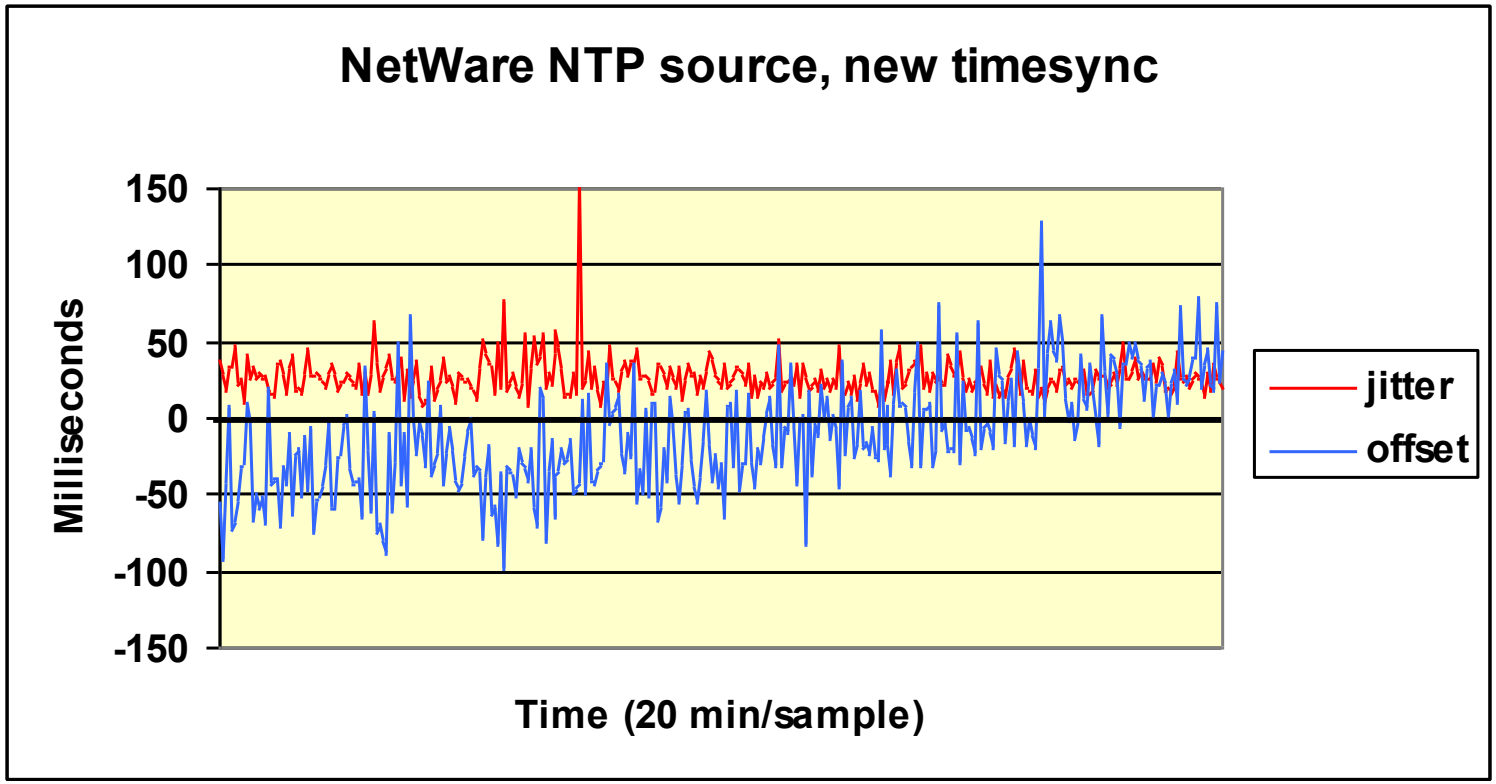
NetWare timesync as NTP client, regular timesync.nlm



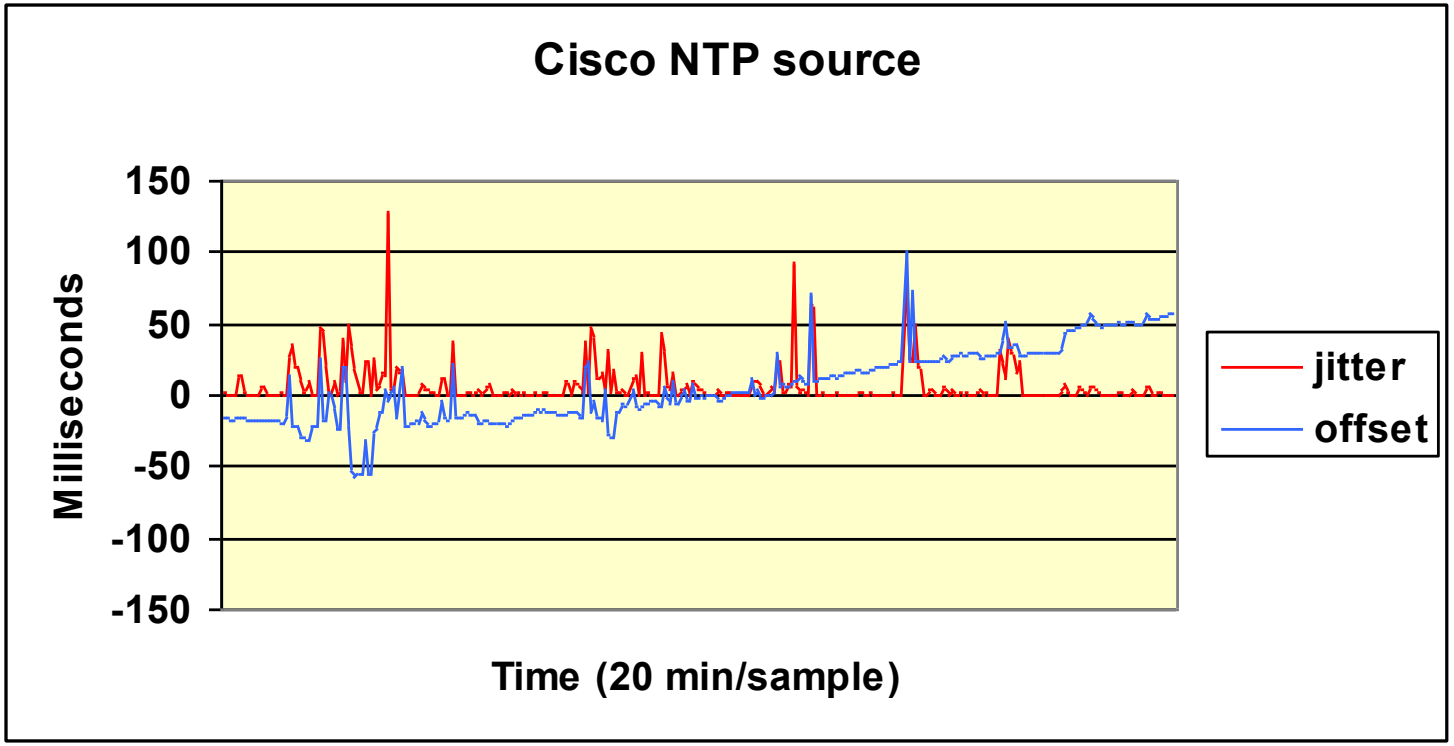
NetWare timesync as NTP client, new timesync.nlm



NetWare timesync as NTP client, new timesync.nlm



Cisco router as NTP source



Timesync experimental results

NW 6 beta 2 refresh

- **Type Primary**
  - Server tries NTP sync. This will fail if time is outside the 600 second basket. Change capture basket size via TIMESYNC MAXIMUM OFFSET = < many seconds> in Monitor | Server Params | Time. Sync will occur if the value is large enough, but watch for xST/xDT loss.
  - CMOS clock is updated during server startup
  - If timesync fails then time is left at unknown value
- **For NW 5.0 use TIMESYNC OFFSET CEILING which is already very large**
- **Both settings are invisible (not even in SET) unless we use *LOAD MONITOR !H***

Timesync experimental results

NW 6 beta 2 refresh

Type SINGLE and REFERENCE

- No sync unless within capture basket
- When captured MDT/MST is wrong and time stabilizes at 45-60min or so into the FUTURE
- Must reload timesync.nlm to regain proper time
- CMOS clock is updated

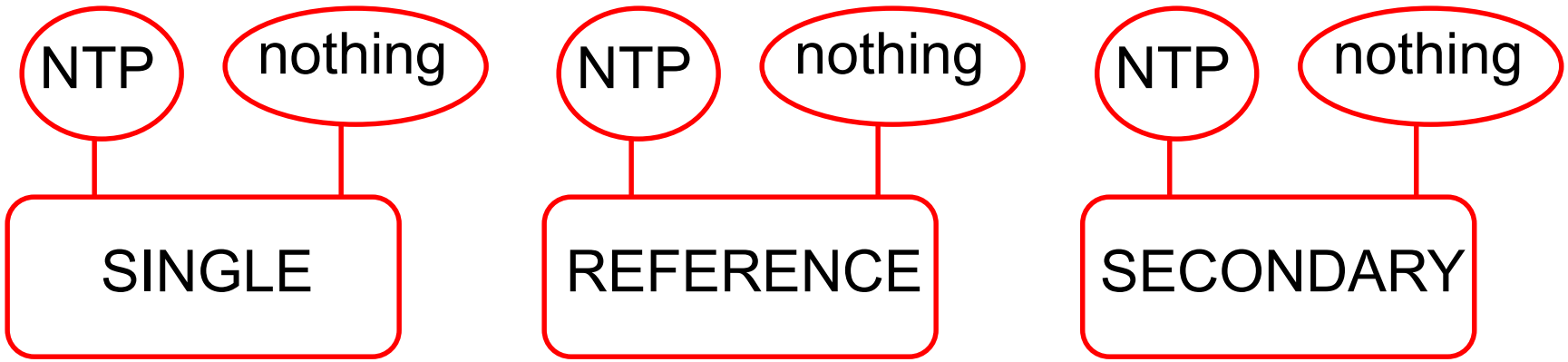
Timesync experimental results

NW 6 beta 2 refresh

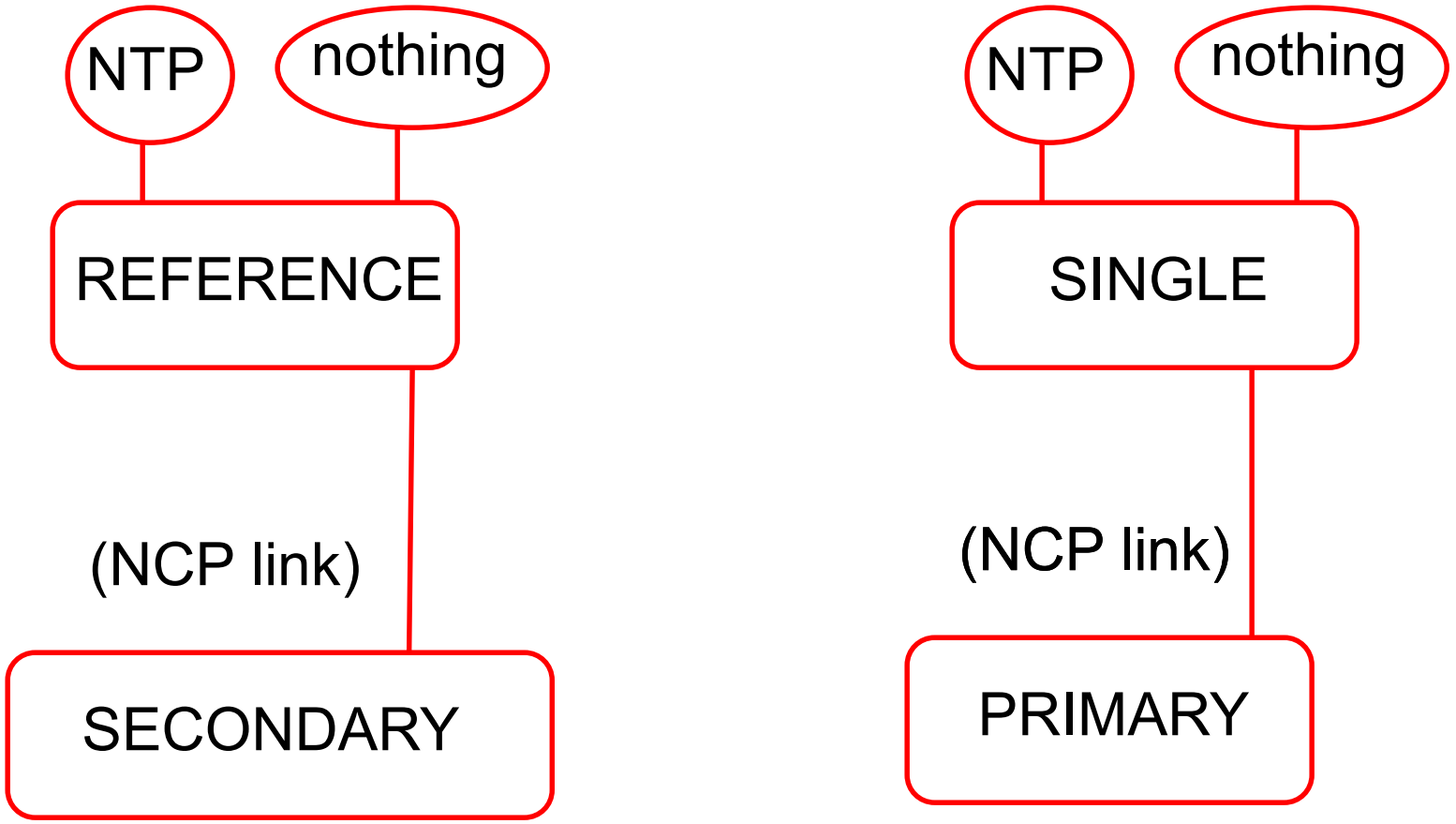
Type SECONDARY

- Syncs on first NTP exchange, but is ahead by one hour (as well as having MST/MDT wrong)
- Reload timesync to get on correct time (but MST/MDT flag is still wrong)
- REFERENCE server complained about syncing to SECONDARY, said it would not, but timeysync debug display shows it did anyway, referring to source as (primary/chained secondary)
- CMOS clock is updated

What seems to work

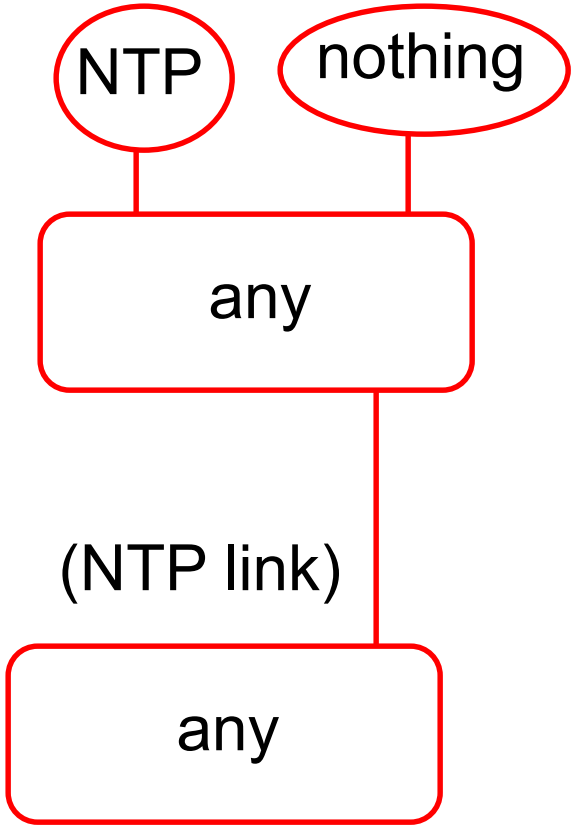


What seems to work



What does not work

Internal NTP link causes NTP query and responses continuously every 62ms



The local clock

- **Despite what the docs and Monitor say, experiments indicate the PC CMOS clock is SET by every kind of timesync**
- **This is actually a good thing if network time is sane, because most often PC clocks are way off, but a little good goes a long way.**
- **If the clock is behind over a Daylight transition the clock can/will be mis-set by about an hour, and the Daylight Time parameter lost,. Internal time can be ahead of proper UTC.**

Timesync, adjustables

Set Timesync Service Advertising = on

- Registers with SLP as timesync.novell.service

Set Timesync Directory Tree Mode = on

- Confines synchronization traffic to one tree

Set Timesync Configured Sources = on

- Use list of NCP or IP names
- Use :123 to invoke NTP with IP, else uses NCP over IP

Timesync, adjustables

Set Timesync Hardware Clock = on/off

- **Primary and Secondary set the clock**
- **Single and Reference read the clock upon every polling interval (meaning time is from PC clock)**

Set Timesync Polling Interval = seconds

- **Defaults to 600 seconds**

Set Timesync Synchronization Radius = msec

- **Defaults to 2000 msec (allowed time error while declaring NDS time is still synchronized)**

NW 5, what it does not do

(Quoting NW 5 documentation:)

TimeSync is an adaptation of NTP and uses algorithms similar to NTP for network delay when obtaining time from a time source.

However, NTP is heavily weighted so some NTP features for TimeSync were omitted for NetWare, such as:

NW 5, what it does not do

Leap second insertion. At regular intervals, clocks need to be compensated by inserting a leap second to the clock time

Drift compensation. NTP keeps track of the pattern of changes to the clock, and over a period of time, it computes a drift compensation factor that is applied to the clock

NW 5, what it does not do

- **Clock filtering and clock selection of algorithms. NTP uses complex algorithms to assess the reliability of each source in its list of time sources and selects only those which are considered to be reliable.**
- **Authentication. NTP uses authentication to ensure the integrety of the time source. TIMESYNC.NLM trusts all time sources.**

NW 5, what it does not do

NTP is strict when considering time sources. If a time source is more than 1000 seconds (about 17 minutes) away for the local clock, NTP rejects the time source as unreliable. The time source is labeled as unreliable and is not used in NTP exchanges.

NW 5, what it does not do

NTP never corrects time directly. The time polled is given as feedback to the clock and the clock corrects its time.

In TimeSync, the time is gradually applied to the local clock. The TIMESYNC.NLM module attempts to correct time within one polling interval, so TimeSync is able to synchronize time in a network more quickly.

(End of quotations)

NW 5 Timesync

The normal time reference for NW is the PC motherboard hardware clock (a poor clock)

We observe servers drift apart in time and lose synchronization, despite timesync. NDS sync ceases.

One remedy is to run Rdate.nlm on servers to periodically force server time to be close to UTC, within seconds.

Load rdate /u /v 1 /p 120 IP-time-giver

Best, use timesync v5.20j or v6.00e plus NTP

NW 5 Timesync

The internal workings of Timesync are proprietary

Various kinds of time serving is confusing

A suggestion is to use NTP V4 amongst servers to sync within a few milliseconds and to keep time aligned even in temporary absence of external sources

Questions?