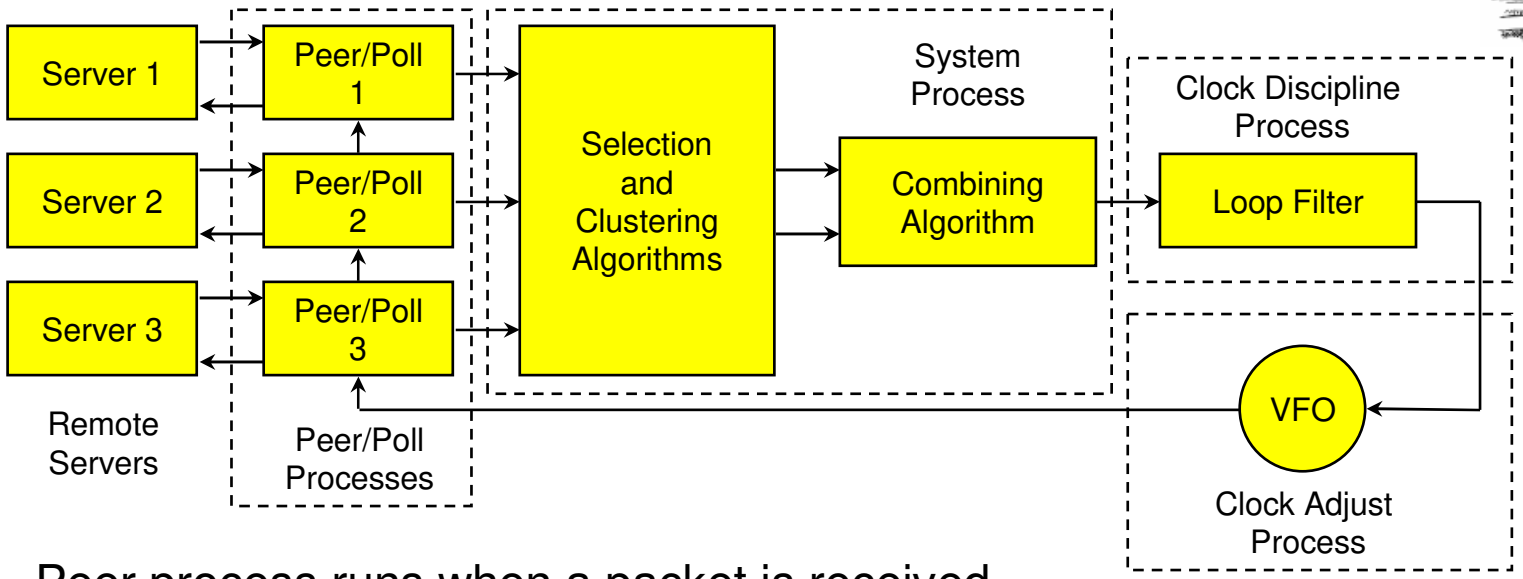
NTP Procedure Descriptions and Flow Diagrams

David L. Mills
University of Delaware
http://www.eecis.udel.edu/~mills
mailto:mills@udel.edu

Sir John Tenniel; *Alice's Adventures in Wonderland,*Lewis Carroll

NTP process overview

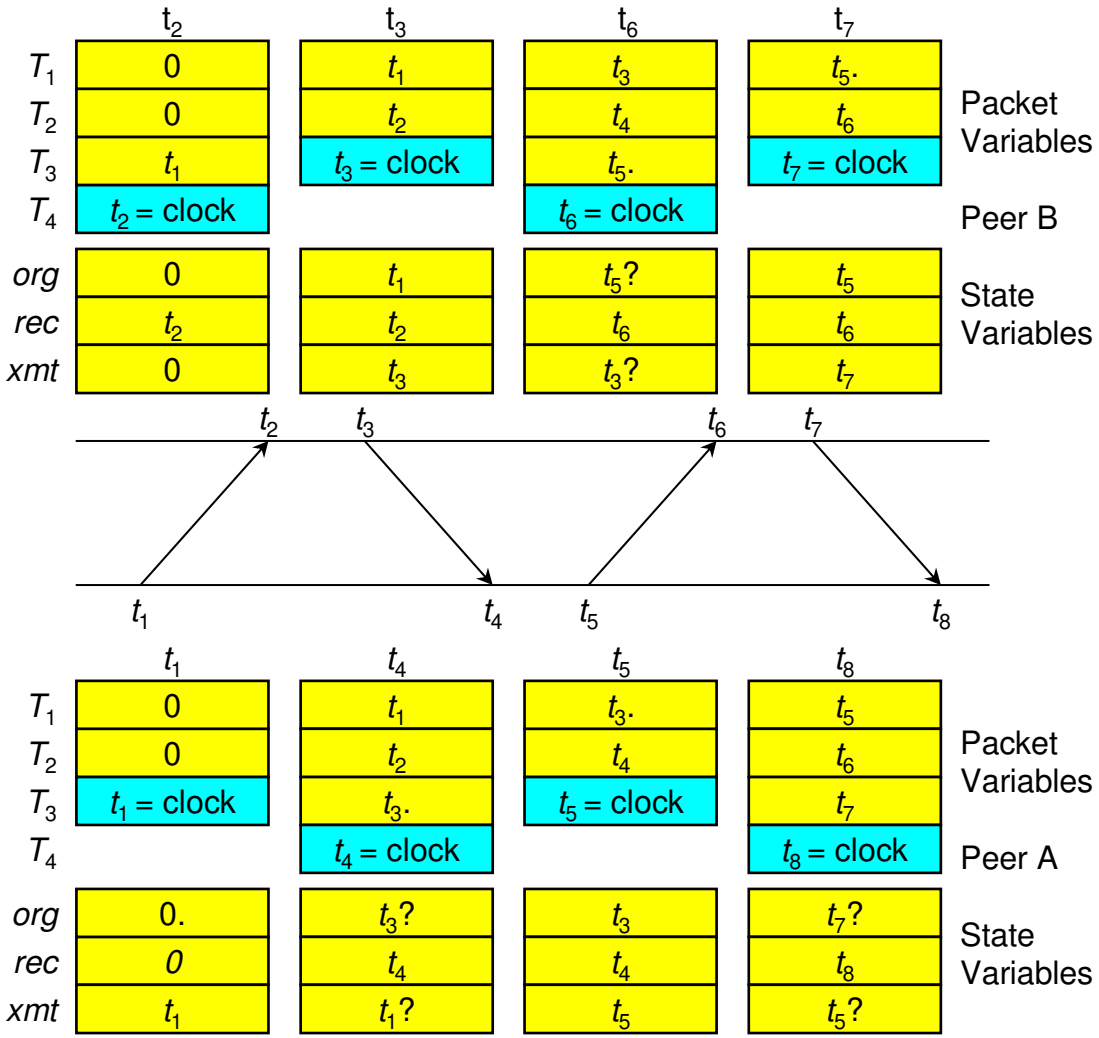


- Peer process runs when a packet is received.
- Poll process sends packets at intervals determined by the clock discipline process and remote server.
- System process runs when a new peer process update is received.
- Clock discipline process runs at intervals determined by the measured network jitter and clock oscillator (VFO) frequency wander.
- Clock adjust process runs at intervals of one second.

NTP peer protocol

- Packet header includes T_1, T_2 and T_3 timestamps.
- Peer state variables *org, rec* and *xmt* record the transmit and receive times of the most recent packet received.
- When a packet is transmitted
  - Copy *org* to T_1 and *rec* to T_2.
  - Copy the current time to *xmt* and to T_3.
- When a packet received
  - If T_3 is the same as *xmt,* this is a duplicate packet.
  - If T_1 is not the same as *org*, this is a bogus packet.
  - Otherwise, copy T_3 to *pkt* and copy the current time to T_4 and *rec.*
- Note that the protocol is symmetric and allows time values to flow both ways simultaneously and is resistant to replays and drops.
- Note the special conditions when either or both peers first start up.

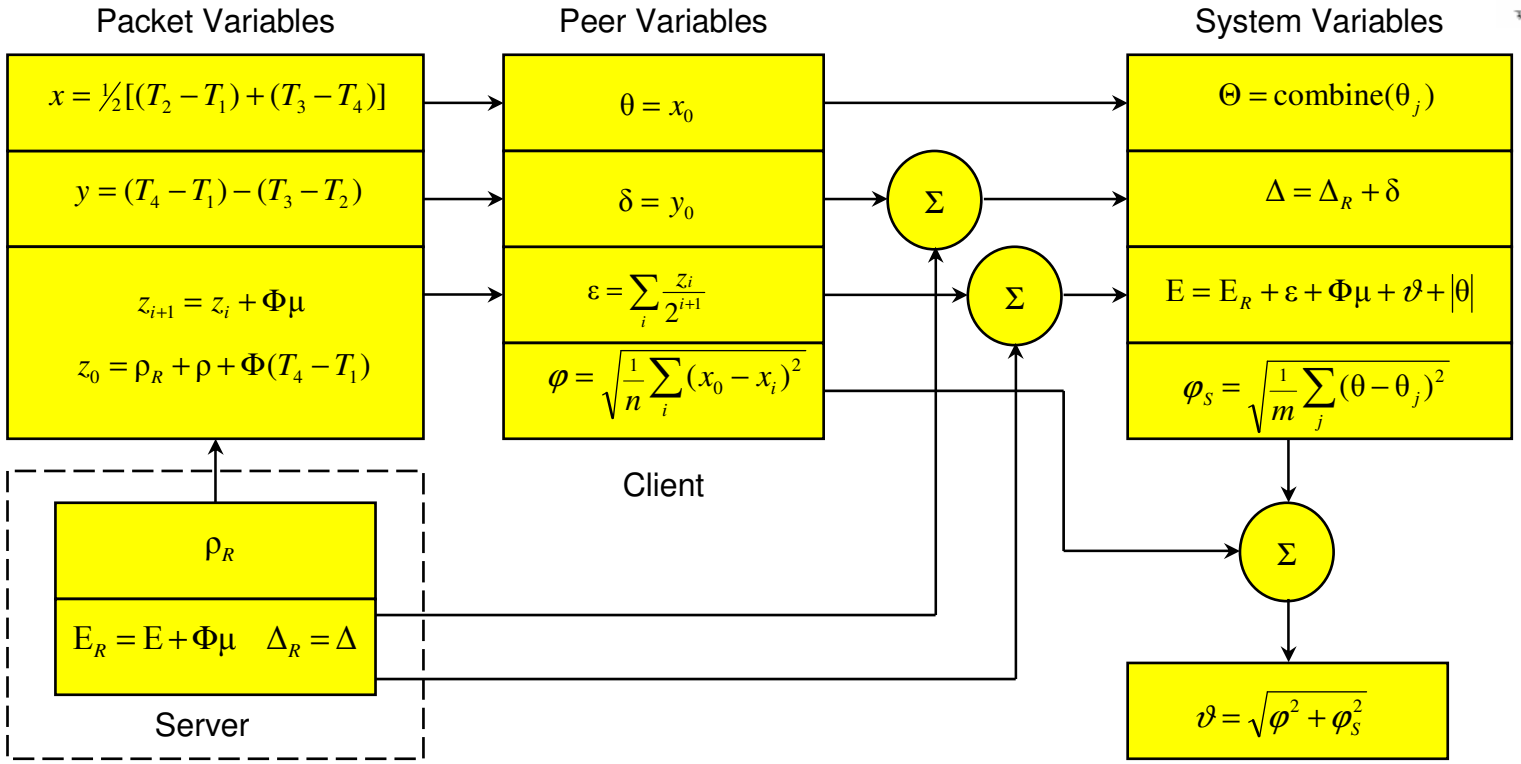
NTP peer protocol example



State Variables	
Name	Description
org	origin timestamp
rec	receive timestamp
xmt	transmit timestamp

Packet Variables	
Name	Description
t_n	origin timestamp
t_{n+1}	receive timestamp
t_{n+2}	transmit timestamp
t_{n+3}	destination timestamp

Computing time values and error estimates



- Packet variables are computed directly from the packet header.
- Peer variables are groomed by the clock filter.
- System variables are groomed from the available peers.

Variable, Parameter and Procedure Libraries

1. NTP Packet header format
2. Process variables
3. Parameters
4. Procedure cross index

Sir John Tenniel; *Alice's Adventures in Wonderland,*Lewis Carroll

Naming conventions

- The names on the following pages are distinguished by the process to which the belong. When necessary to disambiguate the name, the prefix tags above will prepend he name.
  - r. receive packet
  - x. transmit packet
  - p. peer/poll process
  - s. system process
  - c. local clock process
  - p.f. clock filter
  - s.m chime list
  - s.v. survivor list
- The variable names and formula names are used interchangeably to improve readability and reduce flow chart size.

NTP packet header format

Name	Formula	Description
leap	*leap*	leap indicator (LI)
version	*version*	version number (VN)
mode	*mode*	mode
stratum	*stratum*	stratum
poll	τ	poll interval ($\log_2$ s)
precision	ρ	precision ($\log_2$ s)
rootdelay	Δ	root delay
rootdisp	E	root dispersion
refid	*refid*	reference ID
reftime	*reftime*	reference timestamp
org	T_1	origin timestamp
rec	T_2	receive timestamp
xmt	T_3	transmit timestamp
dst*	T_4	destination timestamp*

Packet Variables

* Strictly speaking, *dst* is not a packet variable; it is the value of the system clock upon arrival.

Name	Formula	Description
keyid		key ID
mac		message digest

Message Authentication Code (MAC)

LI	VN	Mode	Strat	Poll	Prec
Root Delay					
Root Dispersion					
Reference ID					
Reference Timestamp (64)					
Origin Timestamp (64)					
Receive Timestamp (64)					
Transmit Timestamp (64)					
MAC (optional 160)					

Process variables I

Name	Formula	Description
Configuration Variables		
srcaddr		source address
dstaddr		destination address
version	*version*	version
hmode	*hmode*	host mode
keyid	*keyid*	key ID
flags		option flags
Packet Variables		
leap	*leap*	leap indicator
pmode	*pmode*	packet mode
stratum	*stratum*	stratum
ppoll	τ	poll interval
rootdelay	Δ	root delay
rootdisp	E	root dispersion
refid	*refid*	reference ID
reftime	*reftime*	reference timestamp
org	T_1	origin timestamp
rec	T_2	receive timestamp
xmt	T_3	transmit timestamp
Working Variables		
t	*t*	update time
filter		clock filter
offset	θ	clock offset
delay	δ	roundtrip delay
disp	ε	dispersion
jitter	φ	jitter

Peer Process Variables (p.)

Name	Formula	Description
t	*t*	update time
offset	θ	clock offset
delay	δ	roundtrip delay
disp	ε	dispersion

Clock Filter Variables (p.f.)

Name	Formula	Description
t	*t*	update time
leap	*leap*	leap indicator
stratum	*stratum*	stratum
poll	τ	poll interval
precision	ρ	precision
rootdelay	Δ	root delay
rootdisp	E	root dispersion
refid	*refid*	reference ID
reftime	*reftime*	reference time
chime		chime list
survivor		survivor list
p	*p*	system peer
offset	Θ	combined offset
jitter	ϑ	combined jitter
flags		option flags

System Process Variables (s.)

Process variables II

Name	Formula	Description
t	*t*	update time
state	*state*	current state
offset	θ	current offset
base	θ_B	base offset
last	θ_B	previous offset
count	*count*	jiggle counter
freq	*freq*	frequency
jitter	φ	RMS jitter
wander	η	RMS wander

Local Clock Process Variables (c.)

Name	Formula	Description
hpoll	*hpoll*	host poll interval
burst	*count*	burst counter
reach	*reach*	reach register
unreach	*unreach*	unreach counter
outdate		last poll time
nextdate		next poll time

Poll Process Variables (p.)

Name	Formula	Description
p	*p*	association ID
type	*t*	edge type
edge	*edge*	edge offset

Chime List Variables (s.m.)

Name	Formula	Description
p	*p*	association ID
metric	λ	survivor metric

Survivor List Variables (s.v.)

Name	Formula	Description
keyid	*keyid*	key ID
mac	*mac*	message digest

Message Authentication Code (MAC)

Parameters I

Name	Value	Description
VERSION	4	version number
PRECISION	-18	precision (log2 s)
MINDISP	.01	minimum dispersion (s)
MAXDISP	16	maximum dispersion (s)
MAXDIST	1	distance threshold (s)
MAXSTRAT	16	maximum stratum (infinity metric)
MINPOLL	4	minimum poll interval (16 s)
MAXPOLL	17	maximum poll interval (36.4 h)
PHI	15e-6	frequency tolerance (15 PPM)
NSTAGE	8	clock register stages
NMAX	50	maximum number of peers
NSANE	1	minimum intersection survivors
NMIN	3	minimum cluster survivors
SGATE	3	spike gate threshold
BDELAY	.004	broadcast delay (s)

Peer Process Parameters

M_SACT	1	symmetric active
M_PASV	2	symmetric passive
M_CLNT	3	client
M_SERV	4	server
M_BCST	5	broadcast
M_BCLN	6	broadcast client (pseudo mode)

Mode Assignments

Parameters II

Name	Value	Description
STEPT	0.128	step threshold (s)
WATCH	900	stepout threshold (s)
PANICT	1000	panic threshold (s)
PLL	65536	PLL loop gain
FLL	18	FLL loop gain
AVG	4	parameter averaging constant
ALLAN	1500	compromise Allan intercept (s)
LIMIT	30	poll-adjust threshold
MAXFREQ	500e-6	frequency tolerance (500 PPM)
PGATE	4	poll-adjust gate

Local Clock Process Parameters

Name	Value	Description
UNREACH	12	unreach counter threshold
BCOUNT	8	packets in a burst
BTIME	2	burst interval (s)

Poll Process Parameters

Name	Value	Description
A_NONE	0	not authenticated
A_OK	1	authentiction OK
A_ERROR	2	authentication error
A.CRYPTO	3	crypto_NAK received
A.NKEY	4	yhntrusted key

Authentication Code Assignments

Procedure cross index I

Name	Description	Related Routines
receive	receive packet	*main, md5, mobilize, packet, find_assoc,access, fast_xmit
packet	process packet	*receive, clock_filter
clock_filter	clock filter	*packet, *poll

Peer Process Routines

Name	Description	Related Routines
main	main program	*system, mobilize, recv_packet, receive
clock_select	clock select	*clock_filter, fit, clock_update
clock_update	clock update	*clock_select, clock_combine, local_clock
clock_combine	clock combine	*clock_update, root_distance
root_dist	root distance	*fit, *clock_select, *clock_combine
fit	fit to synchronize	*clock_select, *poll, root_dist

System Process Routines

Name	Description	Related Routines
local_clock	clock discipline	*clock_update, rstclock, step_time, adjust_time
rstclock	state transition	*local_clock

Local Clock Process Routines

Name	Description	Related Routines
clock_adjust	clock_adjust	*kernel, poll

Clock Adjust Process Routines

Procedure cross index II

name	description	related routines
poll	poll	*clock_adjust, clock_filtert, peer_xmit, poll_update
poll_update	poll update	*packet, *poll
peer_xmit	peer transmit	*poll, md5
fast_xmit	fast transmit	*receive, md5

Poll Process Routines

Name	Description	Related Routines
md5	message digest	*receive, *peer_xmit, *fast_xmit
find_assoc	find association	*receive
mobilize	mobilize association	*main, *receive
clear	clear association	*receive, *clock_update, *poll, *peer_xmit
access	access mask	*receive

Utility Routines

name	description	related routines
recv_packet	receive packet	*main
xmit_packet	send packet	peer_xmit, fast_xmit
get_time	get time	*main, peer_xmit, fast_xmit
step_time	step time	*local_clock
adjust_time	adjust time	*local_clock

Kernel Interface Routines

Packet sanity tests (reference implementation only)

Test	Comment	Code	Condition	Routine
	Packet Flashers			
drop	implementation error	none	$T_3 = 0$ or ($T_1 = 0$ and $T_2 \neq 0$) or ($T_1 \neq 0$ and $T_2 = 0$)	receive
1	duplicate packet	pkt_dupe	$T_3 = xmt$	receive
2	bogus packet	pkt_bogus	$T_1 \neq org$	receive
3	invalid timestamp	pkt_proto	*mode* ≠ BCST and $T_1 = 0$ and $T_2 = 0$	receive
4	access denied	pkt_denied	access restricted, untrusted key, etc.	receive
5	authentication error	pkt_auth	MD5 message hash fails to match message digest	receive
6	peer not synchronized	pkt_unsync	*leap* = 11 or *stratum* >= MAXSTRAT or $T_3 <$ *reftime*	packet
7	invalid distance	pkt_dist	$\Delta_R < 0$ or $E_R < 0$ or $\Delta_R / 2 + E_R >$ MAXDISP	packet
8	autokey keystream error	pkt_autokey	MD5 autokey hash fails to match previous key ID	receive
9	autokey protocol error	pkt_crypto	key mismatch, certificate expired, etc.	receive
	Peer Flashers			
10	peer stratum exceeded	peer_stratum	stratum > sys_stratum in non-symmetric mode	accept
11	peer distance exceeded	peer_dist	distance greater than MAXDIST	accept
12	peer synchronization loop	peer_loop	peer is synchronized to this host	accept
13	peer unfit for synchronization	peer_unfit	unreachable, unsynchronized, noselect	accept

Flow Diagrams

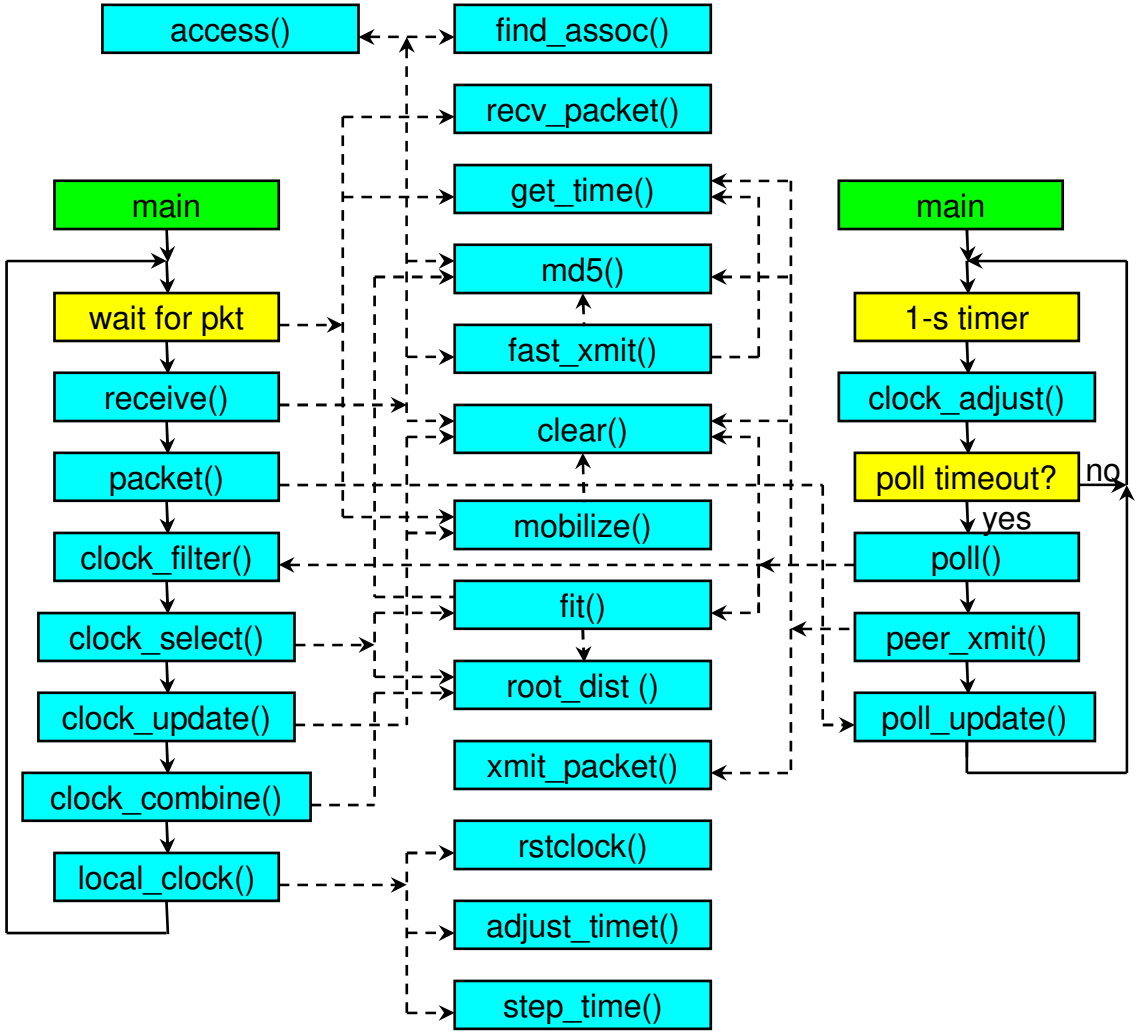
1. Main Program
2. Peer Process
3. System Process
4. Clock Discipline Process
5. Clock Adjust Process
6. Poll Process

Sir John Tenniel; *Alice's Adventures in Wonderland,*Lewis Carroll

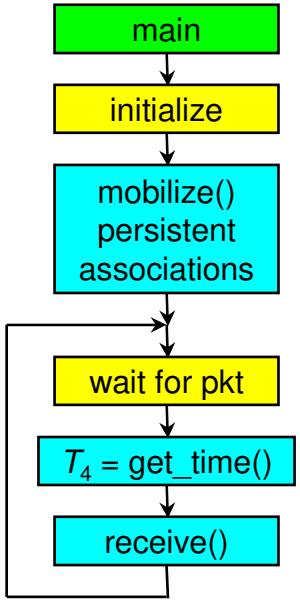
Control flow

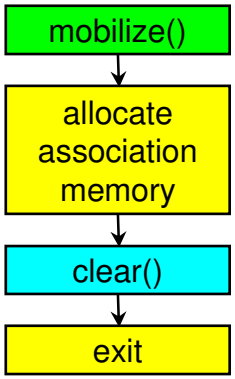
- The main program waits for a packet arrival, then control flows by each of the procedures connected by solid arrows.
  - A client request requires no persistent association; the server response is handled directly by fast_xmit.
  - The packet procedure calls poll_update since it updates the packet poll variable.
- The main program waits for one second, then calls clock_adjust.
- At the poll timeout, control flows by each of the procedures connected by solid arrows.
  - The peer_xmit procedure calls clock_filter when the server has not been heard for three poll intervals. It calls clear on timeout for ephemeral associaitons.
- The dotted arrows show which procdures are called by each procedure with control returning to the calling procedure.

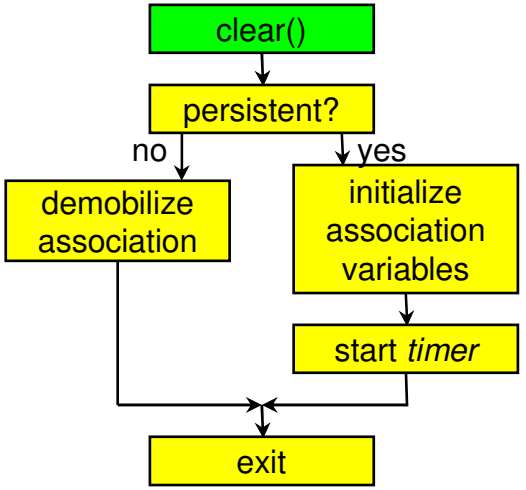
Procedure flow



Main program



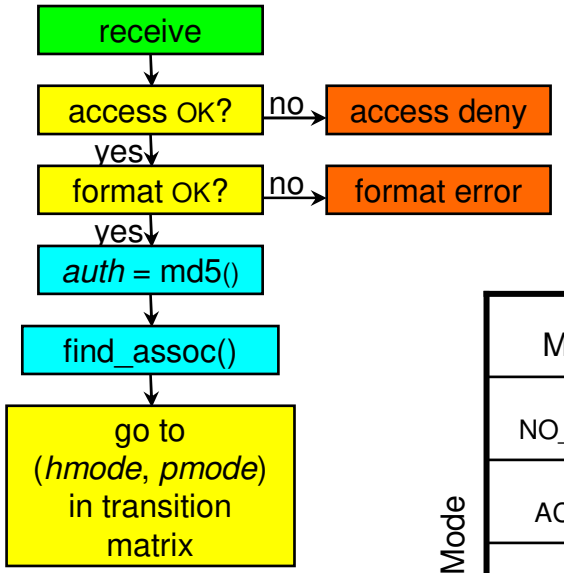


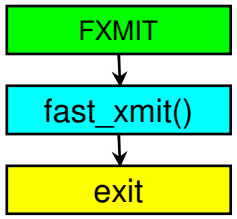


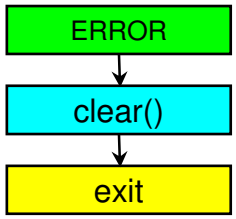
Peer process

- receive procedure
  - Verify integrity, authenticity and consistency of packet data.
  - Match packet to persistent association (client) or reply directly (server).
- packet procedure
  - Compute clock offset, roundtrip delay and dispersion.
  - Copy packet header data to peer state variables
- clock_filter procedure
  - Select the best from among the past eight samples.
  - Calculate filter dispersion, jitter and related values.
  - Implement popcorn spike suppressor.

Peer process: receive() procedure I







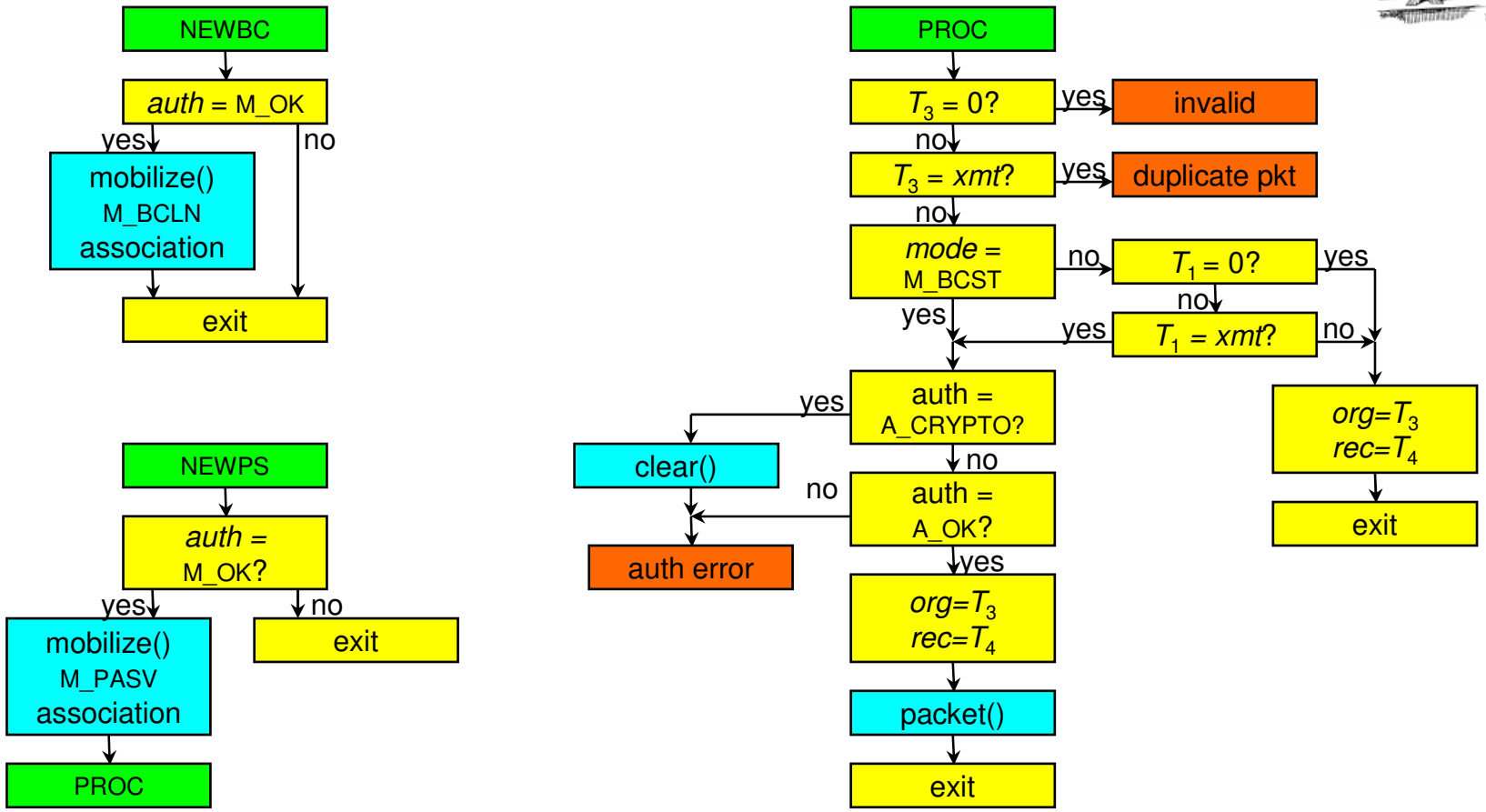
Packet Mode

Association Mode

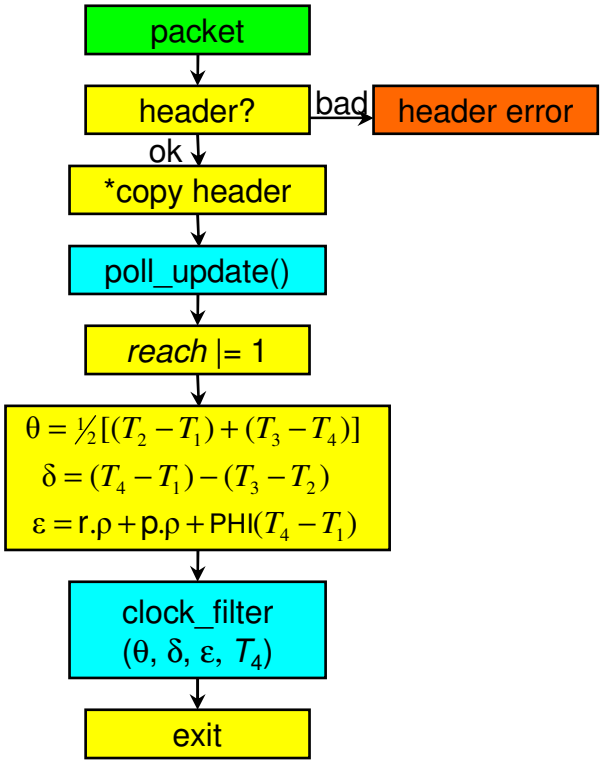
Mode	ACTIVE	PASSIVE	CLIENT	SERVER	BCAST
NO_PEER	NEWPS		FXMIT		NEWBC
ACTIVE	PROC	PROC			
PASSIVE	PROC	ERROR			
CLIENT				PROC	
SERVER					
BCAST					
BCLIENT					PROC

The default (empty box) behavior is to discard the packet.

Peer process: receive() procedure II



Peer process: packet() procedure



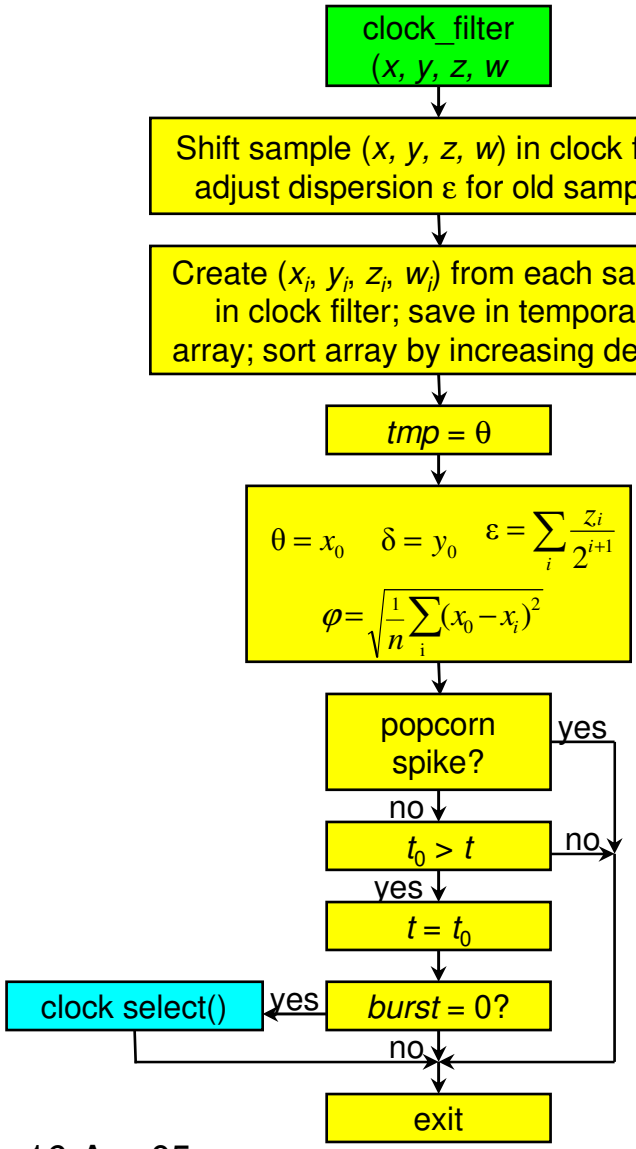
Peer Variables		Packet Variables
leap	←	*leap*
mode	←	*mode*
stratum	←	*stratum*
poll	←	*ppoll*
Δ	←	Δ
E	←	E
refid	←	*refid*
reftime	←	*reftime*

*Copy Header

Variable	Process	Description
T_1	packet	origin timestamp
T_2	packet	receive timestamp
T_3	packet	transmit timestamp
T_4	packet	destination timestamp
θ	peer	offset
δ	peer	delay
ε	poll	dispersion
r.ρ	packet	peer poll interval
p.ρ	peer	host poll interval
reach	poll	reach register
PHI	parameter	frequency tolerance

packet() Procedure

peer process: clock_filter() procedure



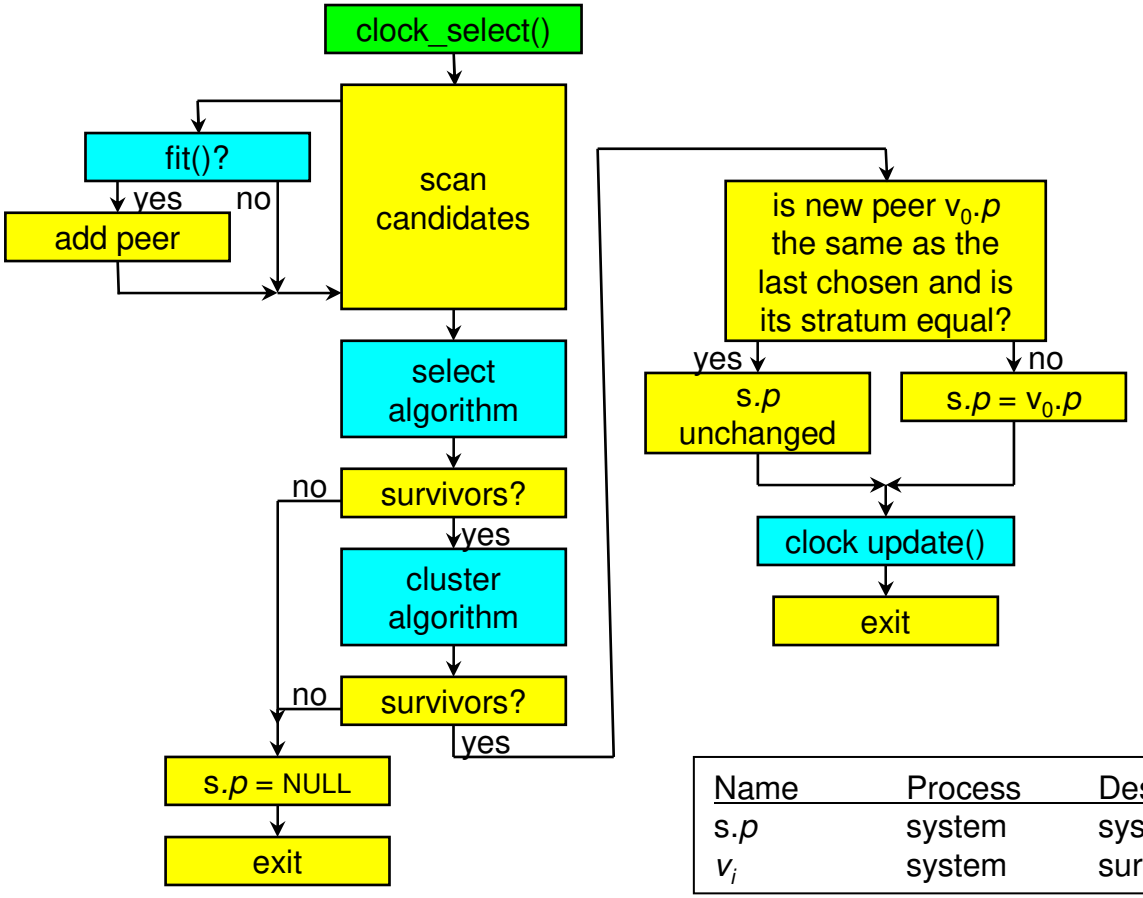
Variable	Process	Description
θ	peer	clock offset
δ	peer	roundtrip delay
ε	peer	filter dispersion
φ	peer	filter jitter
t	peer	last update time
n	peer	number of filter samples
(*x*, *y*, *z*, *w*)		from packet procedure
tmp		temporary
burst	poll	burst counter
τ	local clock	poll interval

clock_filter() Procedure

System process

- clock_select procedure
  - select algorithm: classify available servers as truechimers or falsetickers.
  - cluster algorithm: find and discard outlyers until no more than three survivors remain.
- clock_update procedure
  - call clock_combine procedure to combine weighted server offsets.
  - Call local_clock procedure to discipline the system clock.
  - Update system variables
- rootdist function
  - Return synchronization distance to the primary reference source.
- fit function
  - Return TRUE if selected server is acceptable and root distance less than 1s

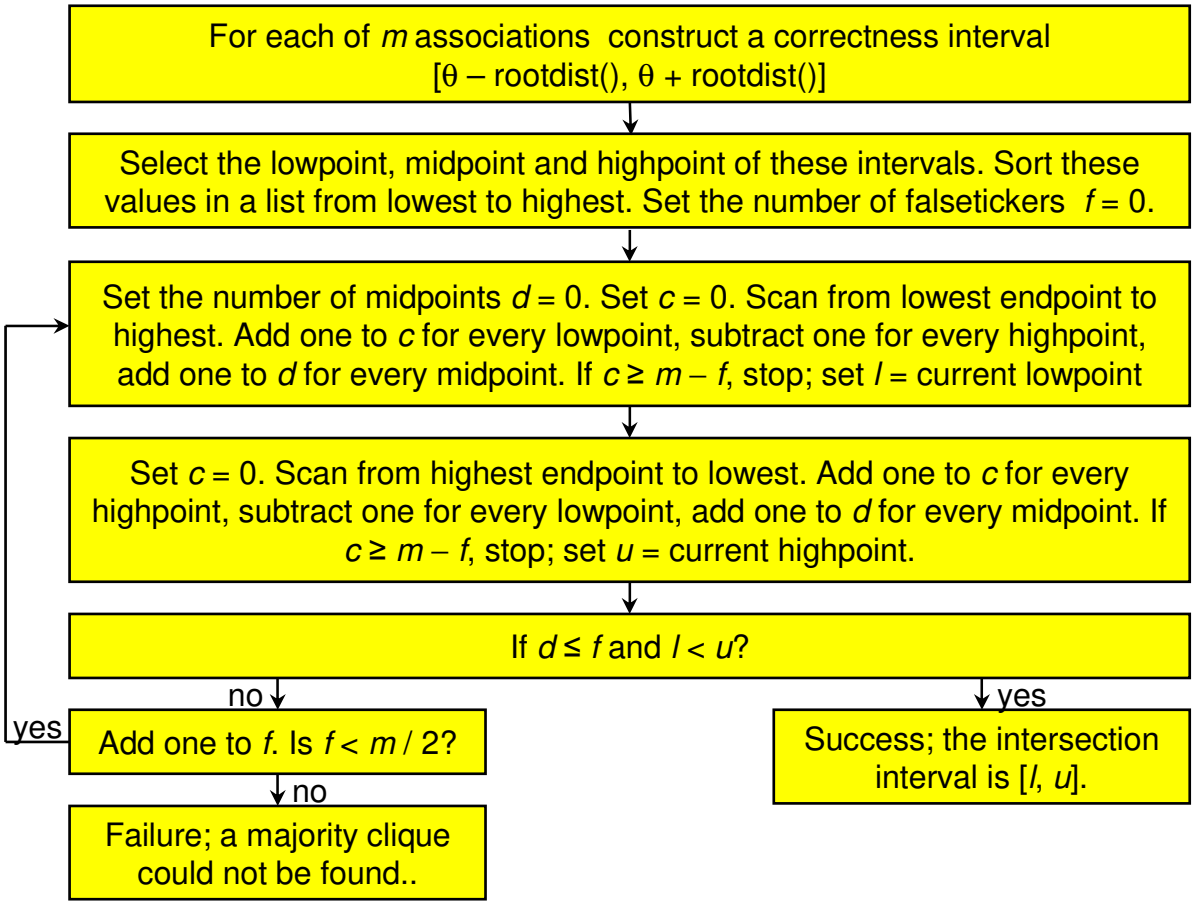
System process: clock_select() procedure



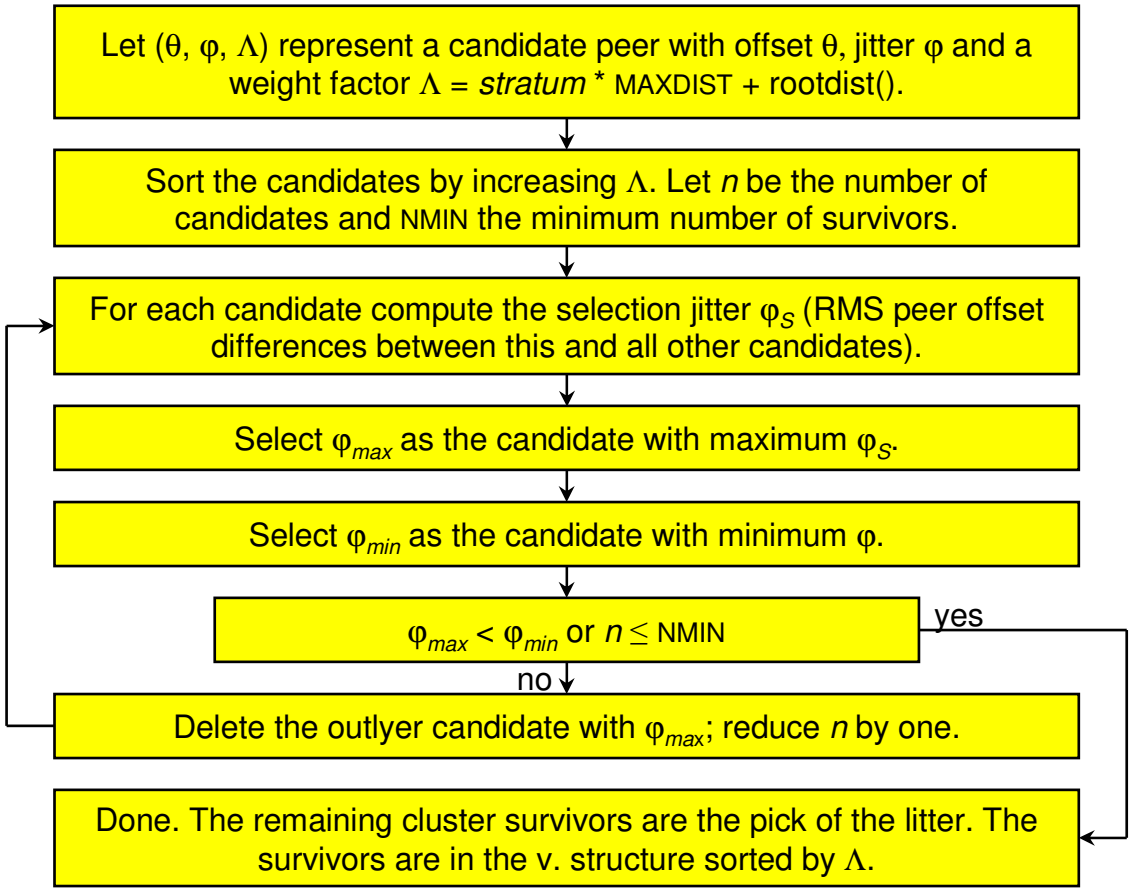
Name	Process	Description
s.p	system	system peer
v_i	system	survivor list sample

clock_select() Procedure

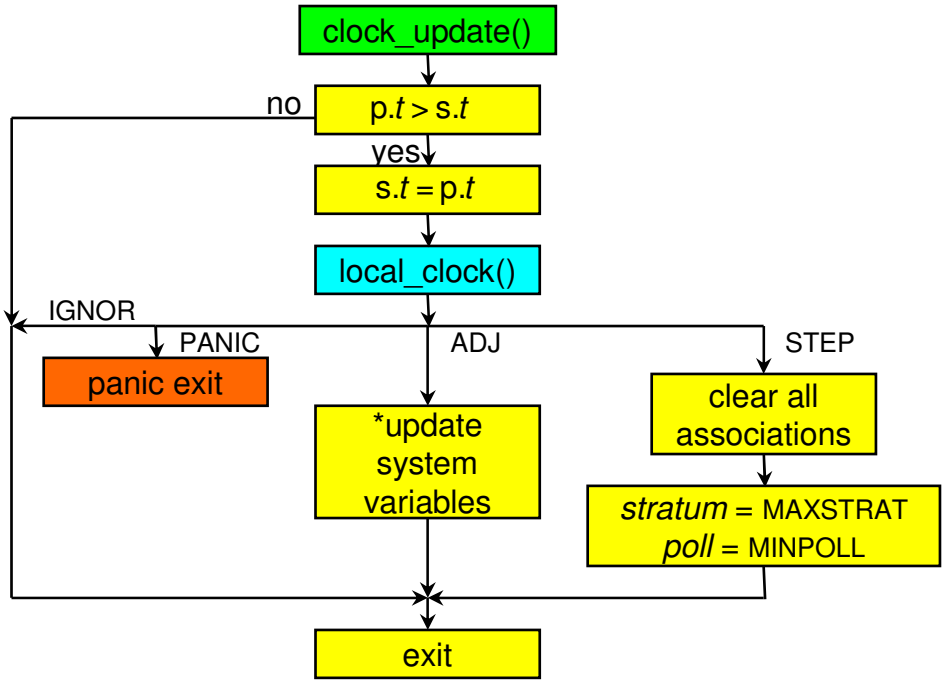
System process: intersection algorithm



system process: cluster algorithm



System process: clock_update() procedure



Name	Process	Description
p.*t*	peer	time
s.*t*	system	offset
s.*stratum*	system	stratum
s.*poll*	system	poll interval
MAXSTRAT	parameter	max stratum
MINPOLL	parameter	min poll interval

clock_update Procedure

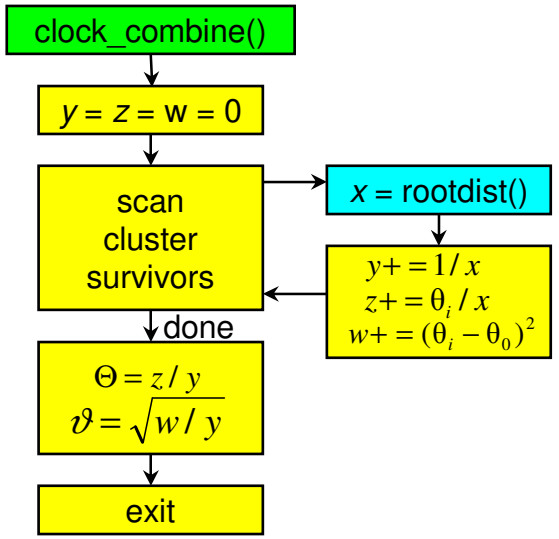
System Variables		Peer Variables
leap	←	*leap*
stratum	←	*stratum*
refid	←	*refid*
reftime	←	*reftime*
Δ	←	$\Delta + \delta$
E	←	$E + \varepsilon +$ PHI $\mu + \varphi + \lvert\theta\rvert$

*Update System Variables

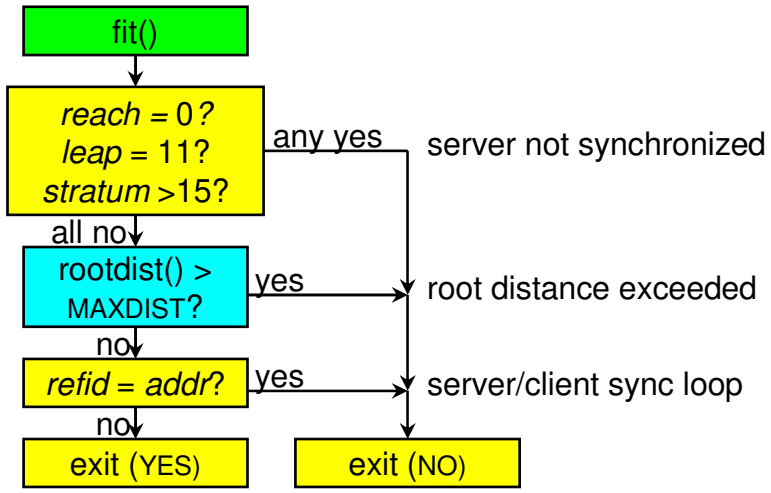
system process: clock_combine() procedure

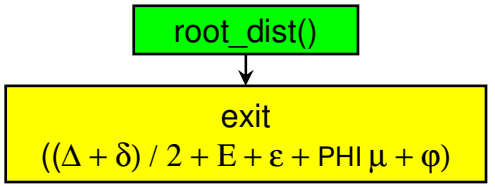
Variable	Process	Description
Θ	system	combined clock offset
ϑ	system	combined jitter
θ_0	survivor list	first survivor offset
θ_i	survivor list	*i*th survivor offset
x, y, z, w		temporaries

combine() Procedure



System process: fit() and root_dist() functions





Variable	Process	Description
leap	peer	leap indicator
stratum	peer	stratum
refid	peer	reference ID
addr	system	hashed local IP addr
reach	poll	reach shift register

fit() function

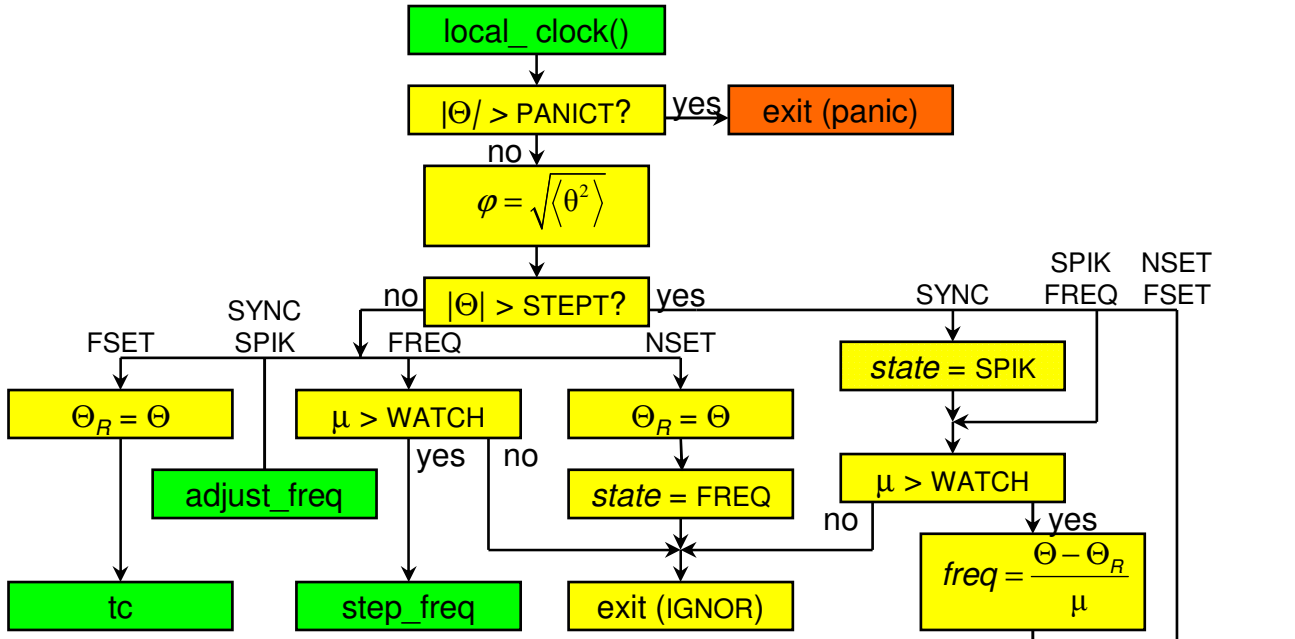
Variable	Process	Description
Δ	peer	root delay
δ	peer	delay
E	peer	root dispersion
ε	peer	dispersion
μ	peer	time since last update
φ	peer	jitter
PHI	parameter	frequency tolerance

rootdist() Function

Clock discipline process

- local_clock() function
  - Discipline system clock using adaptiver-parameter, phase/frequency-lock loop.
- rstclock procedure
  - Transition to new state and initialize variables.
- adjust_freq segment
  - Adjust oscillator frequency using PLL/FLL feedback loop.
- step_freq segment
  - Step oscillator frequency when first starting and no previous information.
- tc segment
  - Adjust time constant as a function of prevailing jitter and oscillator stability.

Clock discipline process: local clock() function I



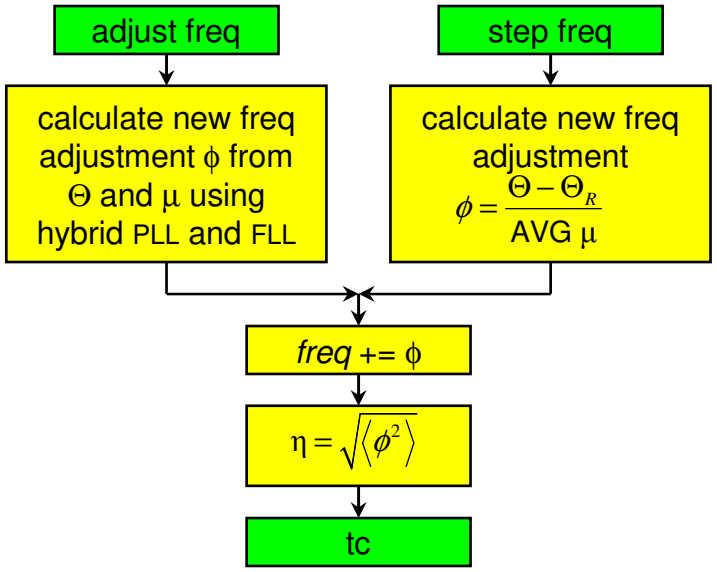
Variable	Process	Description
Θ	local clock	offset
Θ_R	local clock	residual offset
φ	local clock	jitter
μ	local clock	time since last update
state	local clock	state
count	local clock	counter
STEPT	parameter	step threshold (.125 s)
WATCH	parameter	stepout thresh. (900 s)
PANICT	parameter	panic thresh. (1000 s)

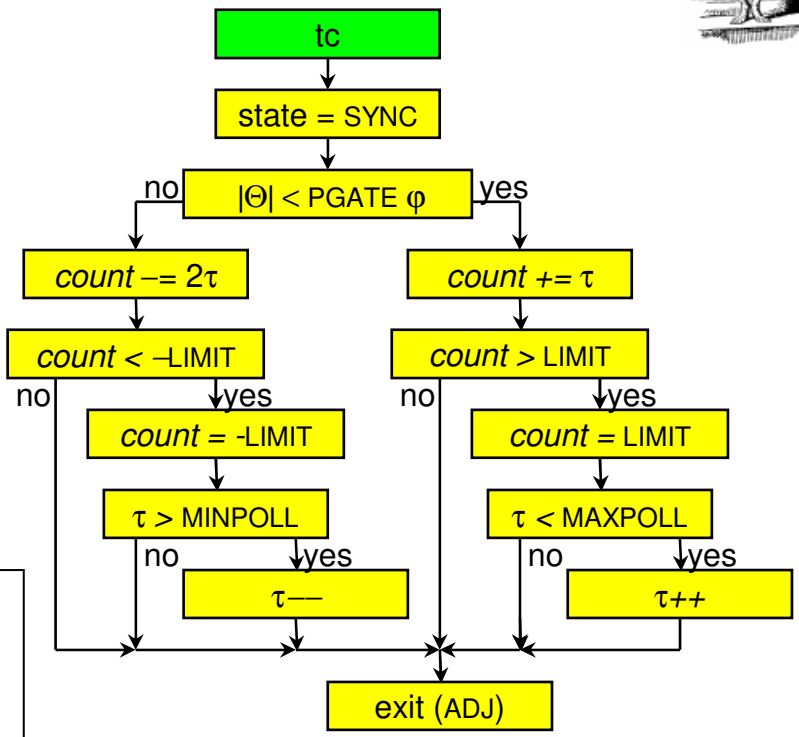
local_clock() Function

Clock discipline process: state transition matrix

State	|Θ| < STEP	|Θ| > STEP	Comments
NSET	>FREQ; adjust time	>FREQ; step time	no frequency file
FSET	>SYNC; adjust time	>SYNC; step time	frequency file
SPIK	>SYNC; adjust freq; adjust time	if (< 900 s) >SPIK else SYNC; step freq; step time	outlyer detected
FREQ	if (< 900 s) >FREQ else >SYNC; step freq; adjust time	if (< 900 s) >FREQ else >SYNC; step freq; step time	Initial frequency
SYNC	>SYNC; adjust freq; adjust time	If (< 900 s) >SPIK else >SYNC; step freq; step time	normal operation

Clock discipline process: local_clock() function II



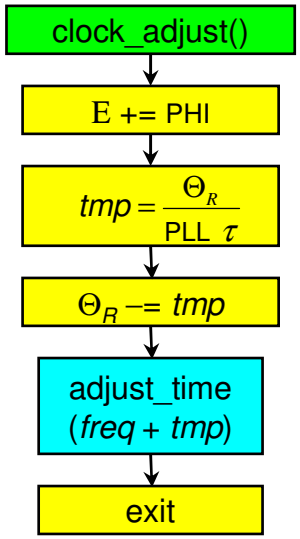


Variable	Process	Description
Θ	local clock	offset
Θ_R	local clock	residual offset
φ	local clock	jitter
μ	local clock	time since last update
ϕ	local clock	frequency adjustment
η	local clock	frequency wander
τ	local clock	poll interval
freq	local clock	frequency
count	local clock	counter
MAXPOLL	parameter	max poll interval
MINPOLL	parameter	min poll interval
LIMIT	parameter	hysteresis limit
AVG	parameter	averaging constant

Clock adjust process: clock_adjust() procedure

- clock_adjust() procedure
  - Called by kernel timer routines once each second.
  - Adjusts system clock frequency as computed by PLL/FLL.
  - system process computes initial system clock offset.
  - Reduce residual clock offset as exponential decay.
- This procedure can also be implemented in the kernel for reduced sawtooth error.

Clock adjust process: clock_adjust() procedure



Name	Process	Description
τ	local clock	poll interval
Θ_R	local clock	current offset
E	system	root dispersion
freq	local clock	frequency
tmp	local	temporary
PHI	parameter	tolerance (15 PPM)
PLL	parameter	PLL loop gain

clock_adjust() Procedure

Poll process

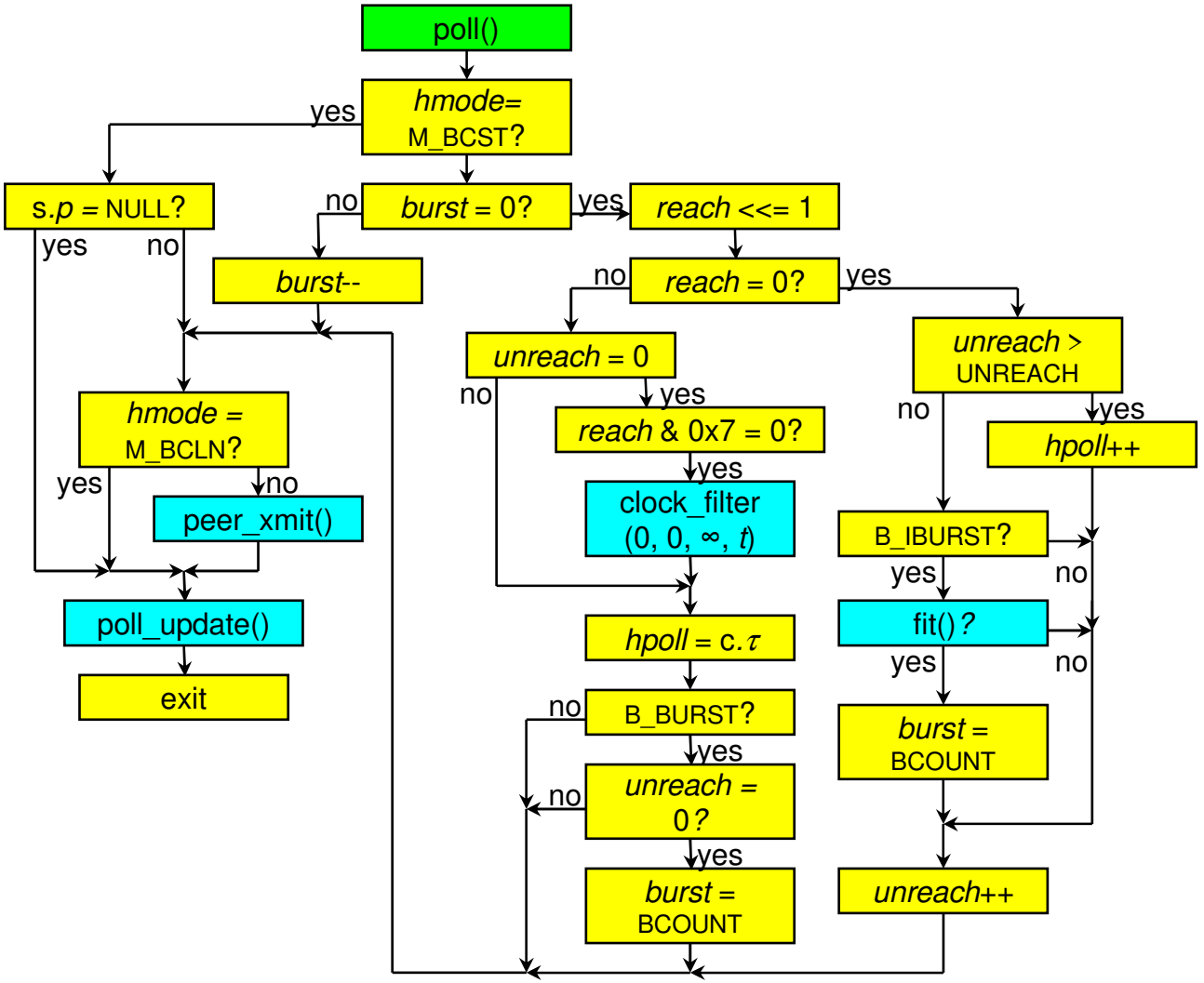
- poll() procedure
  - Determine when to transmit a packet according to poll and burst schedules.
- peer_xmit() and fast_xmit() procedures
  - Format and transmit an NTP packet.
- poll update() procedure
  - Mitigate the poll interval as a function of the host and peer poll intervals and defined lower and upper limits.

Poll process: poll() procedure variables

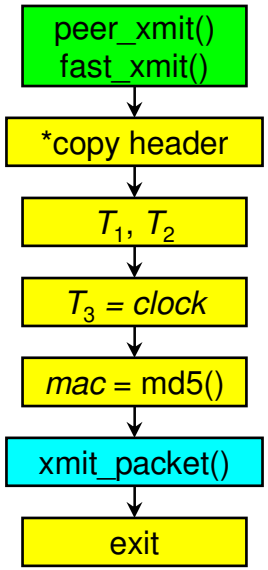
Name	Process	Description
hpoll	poll	host poll interval
hmode	poll	host mode
count	poll	burst counter
reach	poll	reach register
unreach	poll	unreach counter
t	local clock	current time
τ	local clock	poll interval
p	system	system peer
M_BCST	parameter	broadcast server
M-BCLN	parameter	broadcast client
B_BURST	peer flag	burst enable
B_IBURST	peer flag	initial burst enable
B_COUNT	parameter	pkts in a burst

poll() Procedure

Poll process: poll() procedure



Poll process: peer_xmit() and fast_xmit() procedures



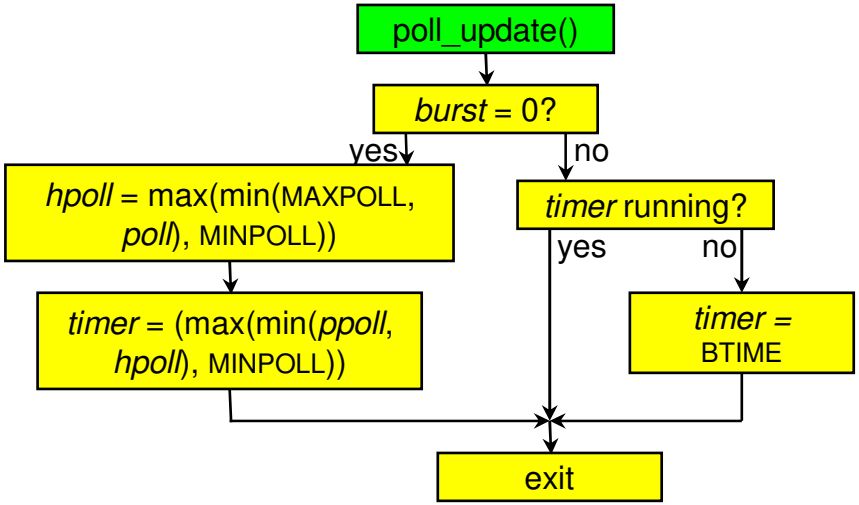
Packet Variable		Desig. Variable
leap	←	s.*leap*
version	←	p.*version*
mode	←	p.*hmode*
stratum	←	*s.stratum*
poll	←	p.*hpoll*
ρ	←	s.ρ
Δ	←	s.Δ
E	←	s.E
refid	←	s.*refid*
reftime	←	s.*reftime*
T_1	←	p.*org*
T_2	←	p.*rec*
T_3	←	*clock*
keyid	←	p.*keyid*
mac	←	md5

*peer_xmit() Procedure

Packet Packet		Desig. Variable
leap	←	s.*leap*
version	←	r.*version*
mode	←	M_SERV
stratum	←	s.*stratum*
poll	←	r.*poll*
ρ	←	s.ρ
Δ	←	s.Δ
E	←	s.E
refid	←	s.*refid*
reftime	←	s.*reftime*
T_1	←	r.T_3
T_2	←	r.T_4
T_3	←	*clock*
keyid	←	r.*keyid*
mac	←	md5

*fast_xmit() Procedure

Poll process: poll_update() procedure



Variable	Process	Description
ppoll	peer	peer poll interval
hpoll	poll	host poll interval
burst	poll	burst counter
timer	kernel	system timer
BTIME	parameter	burst time
MINPOLL	parameter	minimum poll interval
MAXPOLL	parameter	maximum poll interval

poll_update() Procedure

n

Further information

- Network Time Protocol (NTP): http://www.ntp.org/
  - Current NTP Version 3 and 4 software and documentation
  - FAQ and links to other sources and interesting places
- David L. Mills: http://www.eecis.udel.edu/~mills
  - Papers, reports and memoranda in PostScript and PDF formats
  - Briefings in HTML, PostScript, PowerPoint and PDF formats
  - Collaboration resources hardware, software and documentation
  - Songs, photo galleries and after-dinner speech scripts
- FTP server ftp.udel.edu (`pub/ntp` directory)
  - Current NTP Version 3 and 4 software and documentation repository
  - Collaboration resources repository
- Related project descriptions and briefings
  - See “Current Research Project Descriptions and Briefings” at http://www.eecis.udel.edu/~mills/status.htm