Trust Domain Virtual Clock:

Exploiting ICN properties
ICN RG @ IETF 123

Kathleen Nichols
nichols@pollere.net

Problem: Clock Synchronization over a Limited Domain

Defined-trust is an information-centric type framework for Limited Domains (RFC8799) using a Defined-trust Transport (**DeftT**) protocol over IP focused on Operational Technology / Critical Infrastructure networking

- DeftT provides a publish/subscribe API to **collections** of signed, hierarchically named **publications,** *timestamped* to prevent replay attacks and permit expiration
- Each collection is managed by a synchronization module that wraps publications in protocol data units (PDUs) which are exchanged with other member entities via system transports, e.g., UDP, TCP, IPv6, LoRA
- PDUs are hierarchically named, prefixed with the Domain identifier and their collection name (e.g., **msgs**, **cert**, **keys**)

» Scott Gray of Operant Networks found many devices in the field do **not** use an NTP and many deployments do **not** use external (internet) communications
  - DeftT timestamps allow for skew and transit delay (~0.5sec) but some devices had clocks off by hours

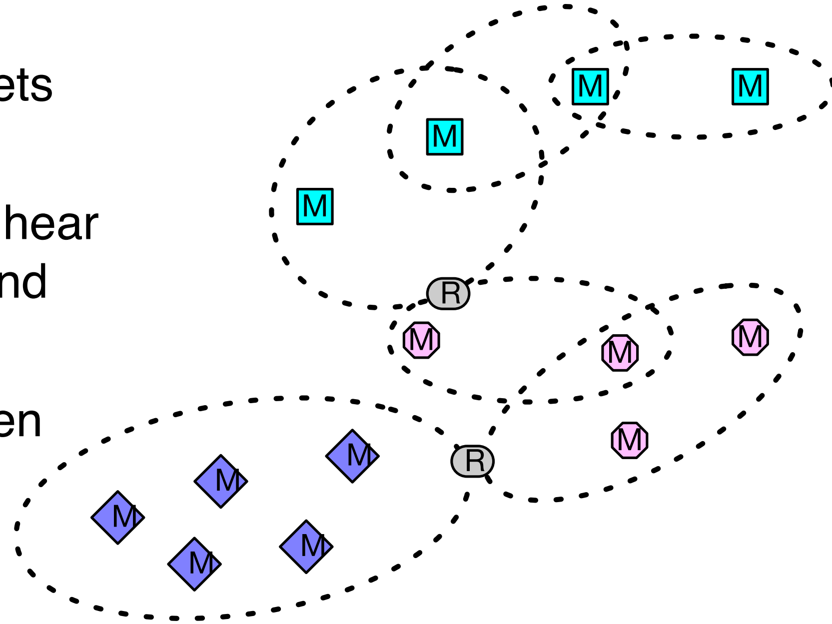
Solution: Give DeftT an integrated, self-contained virtual clock calibrated to its

Context: Defined-trust Communications in a Nutshell

- In Defined-trust, membership and legal communications are pre-defined in rules expressed in a declarative language and compiled to binary
- A certificate with this binary as content and signed by a trust anchor (a *schema*) specifies a *Defined-trust Limited Domain* (*Trust Domain* or TD)
- The schema is used by the integrated trust managment engine of a Defined-trust Transport (DeftT) protocol based on many-to-many synchronization of *publications* governed by the schema. Note certs are particular publications
- Member identities are a chain-of-trust, public certs each signed by the signing key associated with the next cert in the chain: at the root, the trust anchor of the TD, at the leaf, a unique identity cert
- DeftT runs over UDP(IPv6 link local multicast), UDP(unicast), and TCP and presents a publish/subscribe API
- Currently focused on critical infrastructure/operation technology networks

Code at https://github.com/pollere/DCT. Documentation at: https://pollere.net/publications.html, https://pollere.net/Txtdocs/defTrust.html, https://www.tdcommons.org/dpubs_series/7887

- A Trust Domain may span subnets (color) interconnected by relays
- Not all devices of a subnet may hear each other directly: sync can send publications of others in a cAdd
- Communication model is between members and collections, not between endpoints



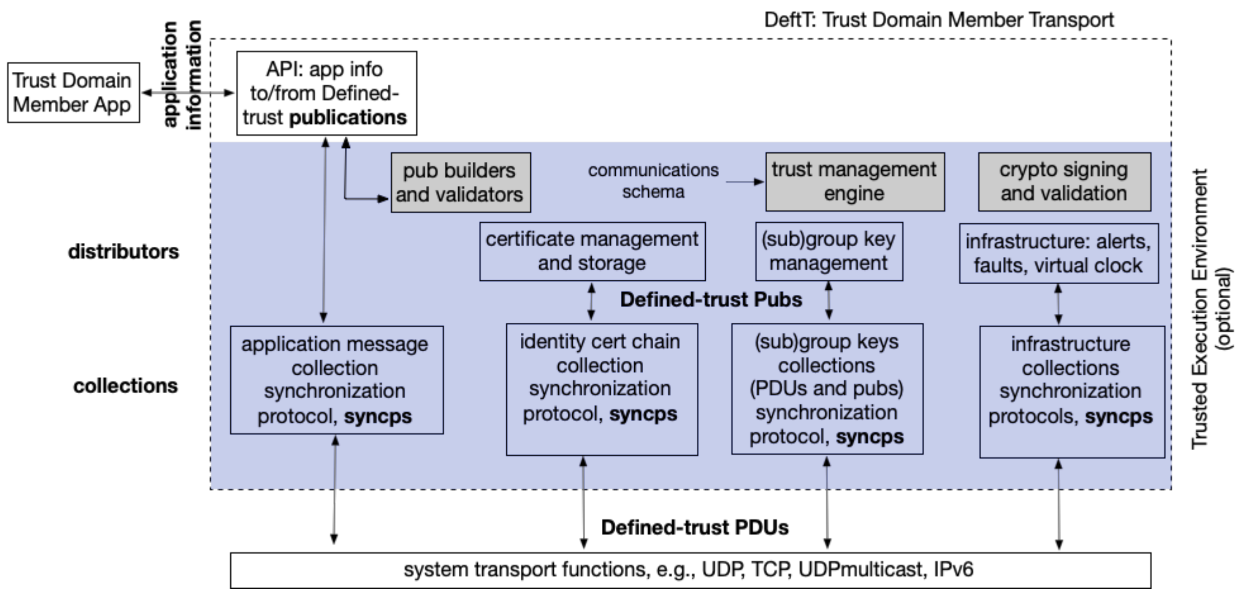
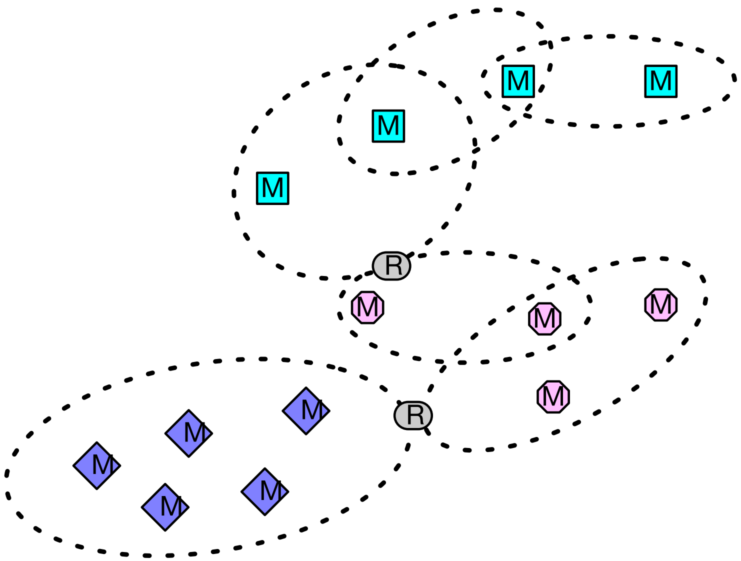


Diagram of DeftT

There are a lot of ways to calibrate clocks in a network

- Internet-based approaches (e.g., NTP) access globally accurate time servers
- Sensor network approaches designate or elect primary sources and build trees to distribute the reference time. They estimate skew and offset parameters relative to the reference.
- For DeftT, neither approach is attractive; looking for something self-contained, robust to communication topology changes and failures and using collections, not endpoints. We can approach calibration somewhat differently:
  - VC is *only* used for publications and is implemented as an **offset** to the local sysclk. Members use their own system clocks for timers, etc.
  - VC doesn't need to track "actual" time nor be monotonic over its history
  - Relative precision can be as large as ~0.5 seconds (though aiming for better)
  - *Ignore* skew estimation and just synchronize the VCs of members when they drift out of tolerance. Publish the local VC often enough (~10 minutes) to catch drifting out of tolerance (preset to a value ~ 20ms - 0.5 sec, depending on network properties. Actual clock drift is expected to be ~40us/sec or 12 ms in 5 min)
  - Can borrow from other approaches while meeting our needs



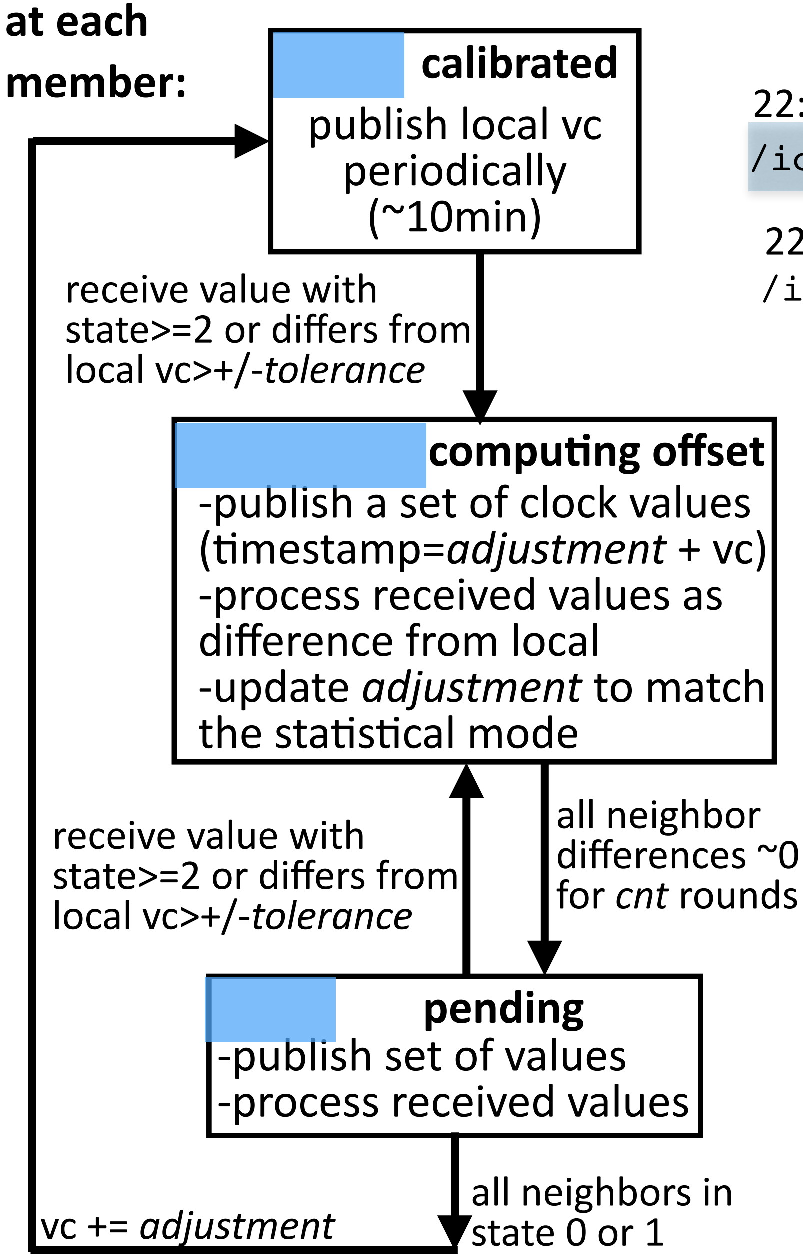
- TD may have no internet connection
- TD may span interconnected subnets (color)
- Relays have DeftTs in multiple subnets but will calibrate across their DeftTs
- Not all nodes of a subnet may hear each other directly
- Deployments may have intermittent connectivity and nodes that sleep

Approach:

- **completely distributed** between members
- **no** use of external servers
- TDVC calibration is within single-hop reachable *neighborhoods* and propagates gossip- or diffusion-style (non-default sync parameters)
  - a neighborhood **may** be smaller than a subnet
  - TDVC's syncps does **not** send pubs of others
  - TDVC's clock pubs have short lifetimes so only sent once
- members with precise clocks can have a *capability* in their identity chain

DeftT's Collection-based Communications

- Pubs are organized into separately managed collections:
  - **msgs** for pubs that carry application data (required)
  - **cert** for certificate pubs: identity chains, schema (required)
  - **keys** for group key management where encryption is used (optional)
  - **tdvc** for coordinating TD virtual clock (optional)
  - **logs** for log information (optional)
- DeftT's PDUs are collection-specific TLV-formatted containers used by its sync module to synchronize a collection across all members of a TD
  - descendant (with additional parentage) of CCN/NDN's Interest/Data
  - **cStates** announce the state of a member's collection
  - **cAdds** respond to a specific cState and carry (as content) pubs that are missing from that cState
- Pubs and cAdds use the same basic TLV format (included at end of slides)
- cStates/cAdds are specfic to a collection
- The TD virtual clock is a *distributor* that uses the **tdvc** collection



22:08:39.670183 cAdd contains:
`/iot2/clk/0/6/gate/25-07-16@20:08:39.672720`

22:09:45.332386 cAdd contains:
`/iot2/clk/0/6/sidedoor/25-07-16@20:09:45.302682`

timestamp diff of 35.728ms > 30ms at gate causes:

22:09:45.342823 cAdd contains publication:
`/iot2/clk/2/6/gate/25-07-16@20:09:45.338410`

sets off a calibration cycle at all members

name components:

- prefix for domain
- subtopic for vc offset calibration
- publisher’s calibration state
- # in nbrhood in agreement with this vc
- publisher’s name tag (for debugging)
- virtual clock value

pub lifetimes for this subtopic are set too small to be resent

TDVC calibration method - calibration cycle

- Receiving members compute clock difference values (amount local virtual clock is ahead of sender, positive or negative)
  - save the value in a record for the sender if the first clock publication from a neighbor or a neighbor whose state has changed
  - else compare the difference to the stored value for sender, replacing the old value if smaller
  - update the record with sender's current **nhSz** and state
  - minimum received value is used to alleviate the non-determinism of processing and transport delays
- Members perform an offset computation a short delay after sending their set of clock publications
- Rounds of sending clock publication sets and computing offsets continue until all neighbors are within tolerance.
  - a member in-tolerance with all its neighbors moves to state 1 and must remain in tolerance through several rounds and have all neighbors in state 1 or 0 before moving to *calibrated* state 0
  - a member not in calibration increments its state (rolling over to 2)

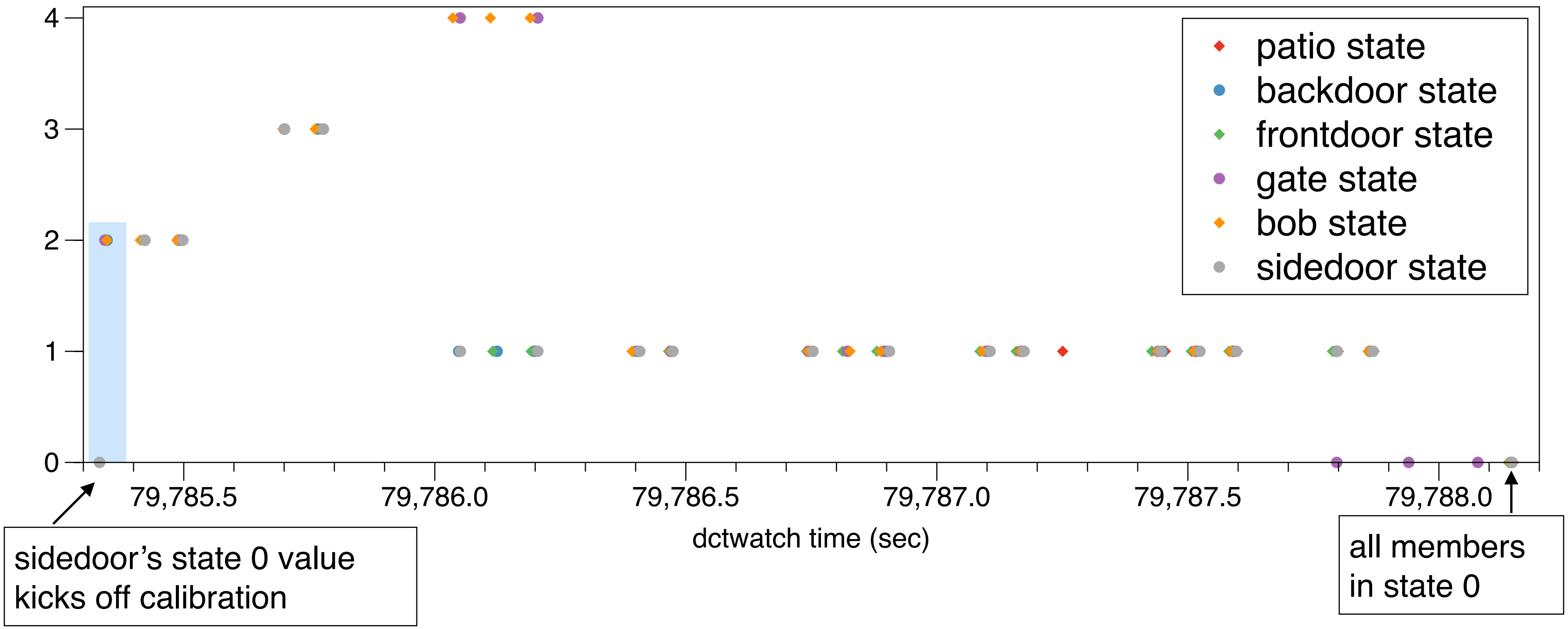
```
22:09:43.918099  St  frontdoor 212 /^573c751de902b8f4/tdvc/^f0018f668bea8d94..
22:09:45.332386  Ad  sidedoor  295 /^573c751de902b8f4/tdvc/^c2fec679
 | /iot2/clk/0/6/sidedoor/25-07-16@20:09:45.302682
22:09:45.342823  Ad  gate 292 /^573c751de902b8f4/tdvc/^c2fec679
 | /iot2/clk/2/6/gate/25-07-16@20:09:45.338410
22:09:45.345069  St  bob  232 /^573c751de902b8f4/tdvc/^f0018f668bea8d94..
22:09:45.347125  Ad  bob  291 /^573c751de902b8f4/tdvc/^82ce2750
 | /iot2/clk/2/6/bob/25-07-16@20:09:45.334044
22:09:45.347163  Ad  frontdoor  297 /^573c751de902b8f4/tdvc/^82ce2750
 | /iot2/clk/2/6/frontdoor/25-07-16@20:09:45.309936
22:09:45.347193  Ad  patio  293 /^573c751de902b8f4/tdvc/^c2fec679
 | /iot2/clk/2/6/patio/25-07-16@20:09:45.325470
22:09:45.347217  Ad  sidedoor  296 /^573c751de902b8f4/tdvc/^82ce2750
 | /iot2/clk/2/6/sidedoor/25-07-16@20:09:45.316750
22:09:45.347242  Ad  backdoor  296 /^573c751de902b8f4/tdvc/^82ce2750
 | /iot2/clk/2/6/backdoor/25-07-16@20:09:45.341739
[showing only publications with dctwatch time of their cAdds]
22:09:45.413460  /iot2/clk/2/6/bob/25-07-16@20:09:45.401995
22:09:45.422358 /iot2/clk/2/6/backdoor/25-07-16@20:09:45.410975
[clk publications continue, e.g.]
22:09:45.842839 /iot2/clk/3/4/frontdoor/25-07-16@20:09:45.840072
22:09:45.846561 /iot2/clk/3/4/patio/25-07-16@20:09:45.843065
22:09:45.846590 /iot2/clk/3/2/gate/25-07-16@20:09:45.849128
[at the finish]
22:09:48.077476 /iot2/clk/0/6/gate/25-07-16@20:09:48.072083
22:09:48.138361 /iot2/clk/0/6/frontdoor/25-07-16@20:09:48.129340
22:09:48.140845 /iot2/clk/0/6/bob/25-07-16@20:09:48.134228
22:09:48.143715 /iot2/clk/0/6/backdoor/25-07-16@20:09:48.136646
22:09:48.145901 /iot2/clk/0/6/sidedoor/25-07-16@20:09:48.145300
22:09:48.148510 /iot2/clk/0/6/patio/25-07-16@20:09:48.147917
```

>30ms diff starts calibration

3 **clk**s in each "round"

mid-calibration

all 6 members in state 0



Processing **logs** publications (logs when adjustment changes, backdoor was unchanged):

```
79787.795945 -7200.000493 #gate
79788.132631 -7200.005494 #frontdoor
79788.138413 -7200.005263 #bob
79788.145320 -7200.000575 #sidedoor
79788.147928 -7200.000595 #patio
```

-first column is the dctwatch time

-second column is dctwatch time minus the logs pub's timestamp to estimate the vc offset used by the publisher

-adjustments are within 5ms (calibration is to +/-quantization)

Practical issue: processing and transit delay add noise to clk values

- Clock differences are quantized, rounding down. Quantization should be slightly greater than the expected transit/processing delay for domain's subnets
- TDVC distributor includes a round-trip delay (RTD) estimator for its neighborhood that will measure processing and transit delays
  - for broadcast subnets, this estimate can be used as the quantization
  - subnets with more significant delay can subtract from clock differences
- A domain may use quantization alone to deal with delays on broadcast subnets while subtracting RTD/2 from other subnets (e.g., long TCP delay)
- TDVC's neighborhood-based RTD approach works for both unicast and broadcast
- Samples are taken continuously with older samples discarded and minimum values used for delay estimates
- If delay estimates change significantly, quantization can be updated

send ping
periodically (~10min):
-randomly select a member **tp**
-save **tp** and systime (*now*)
-publish ping

publishes:
`iot2/png/^becafe49e7c8b63f../gate/25-07-17@00:21:01.958478`

id cert thumbprint of

"passive" ping RTDs will be smaller, but so far this is negligible

receive ping (not my tp)
save **tp** and *now*

receive ping (my tp)
send echo

publishes:
`/iot2/eco/frontdoor/25-07-16@23:21:01.958957`

receive echo
-use signing chain to get publisher's tp
-retrieve ping time, if any
-subtract from *now*
-save on RTD list (per nbrhood or per sender)

name components:
–prefix for domain
–subtopic for vc offset calibration
–thumbprint of identity cert of destination
–publisher's name tag (for debugging)
–virtual clock value

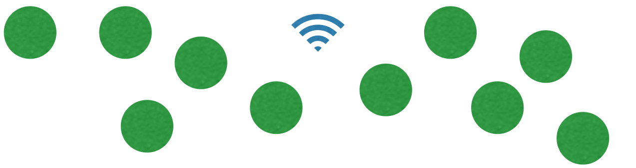
Early Testing Results

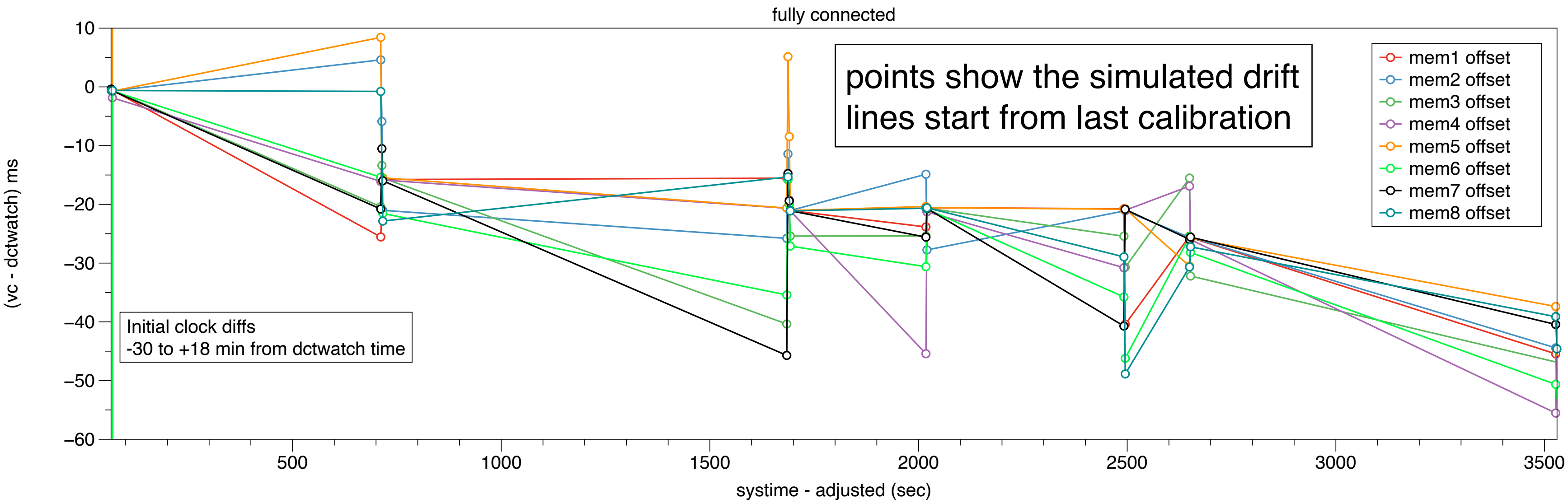
- 8 members emulated on a single machine; RTDs are < ~2msec, using 5ms quantization and 30ms difference tolerance
- each gets a random offset at initialization random(-5,5)*360sec and a randomized drift emulation that adds (-2,2)*5ms at 2.5sec intervals
- DCT can emulate three types of connectivity in a single subnet: full, two overlapping groups, and a linear arrangement (1 or 2 neighbors each)
- Initial calibration took 2.8 sec for fully connected and 2groups, and 8 sec for linear arrangement (includes built-in wait times)
- Plotted values:
  - members publish log messages on first entry to state 2 (calibration start) and first entry to state 0 (finished)

```
17:37:51.303358  2.59m  Ad  308  /^573c751de902b8f4/logs/^b7df1186
 | /iot2/tdvc/cal/device/frontdoor/1/25-07-20@17:25:51.303050
```

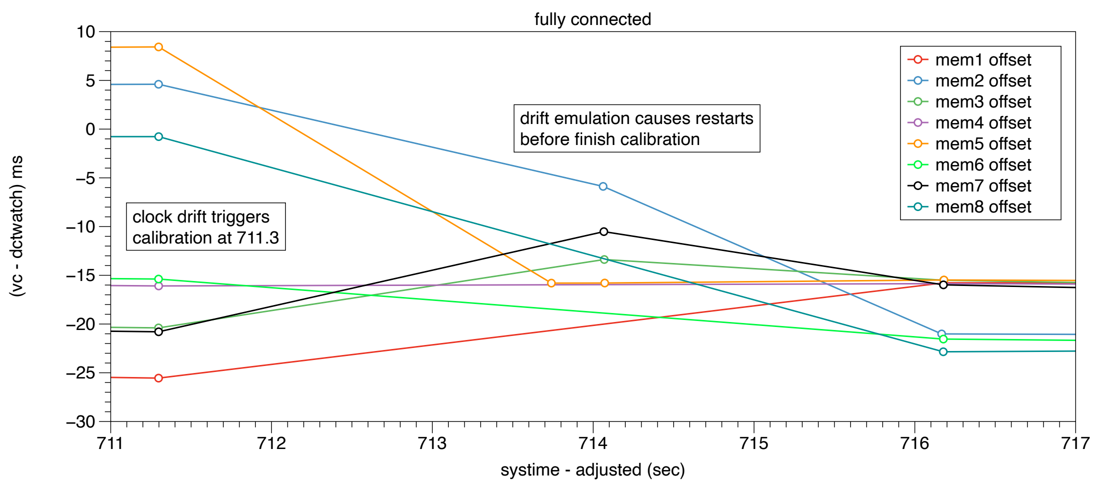
  - difference of log's timestamp from the system clock (the dctwatch time) is an *estimate* of the offset value used at a member
  - offsets should be within +/- 5ms of each other
  - extract state from **clk** publications to illustrate calibration cycle behavior

Eight members fully connected via broadcast

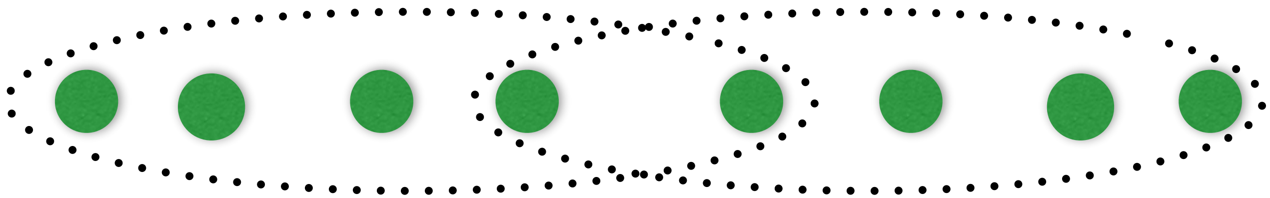


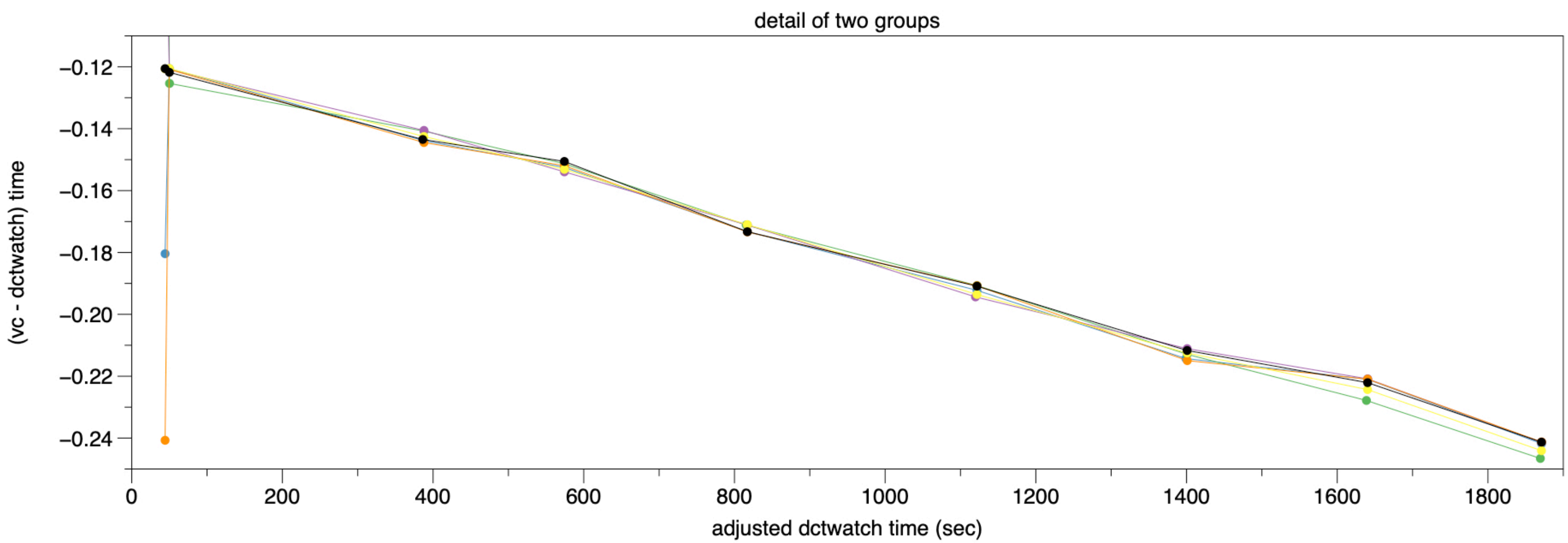


command line prints showed all 8 had the *same* offset after first calibration. Estimate differences due to variations in processing and capture times

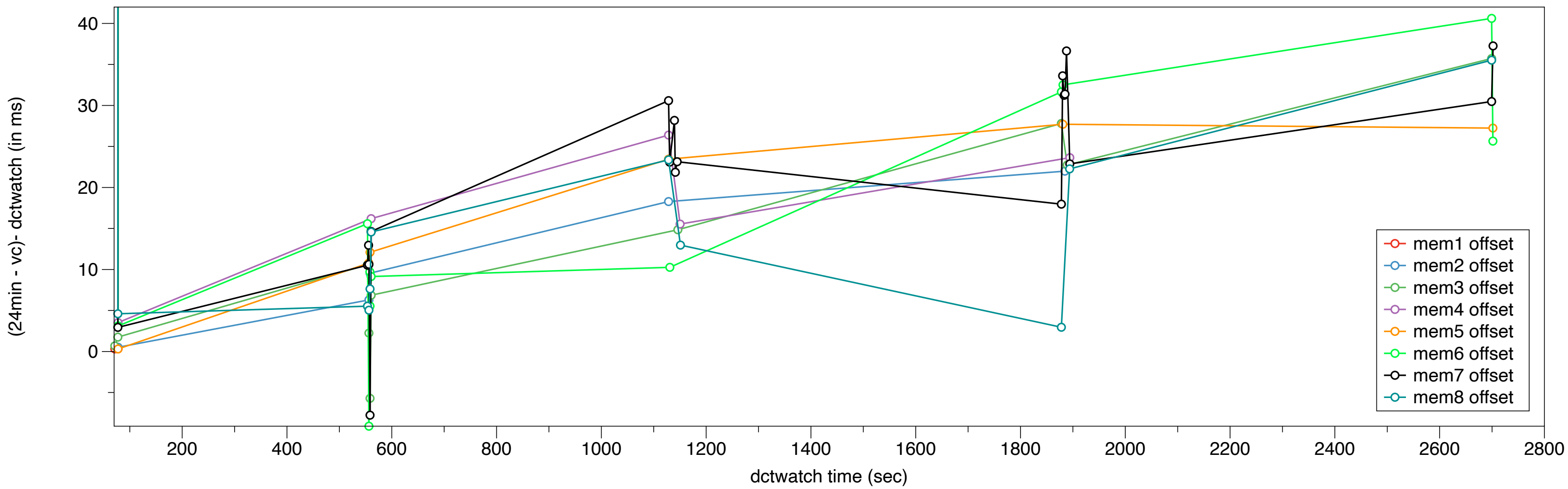


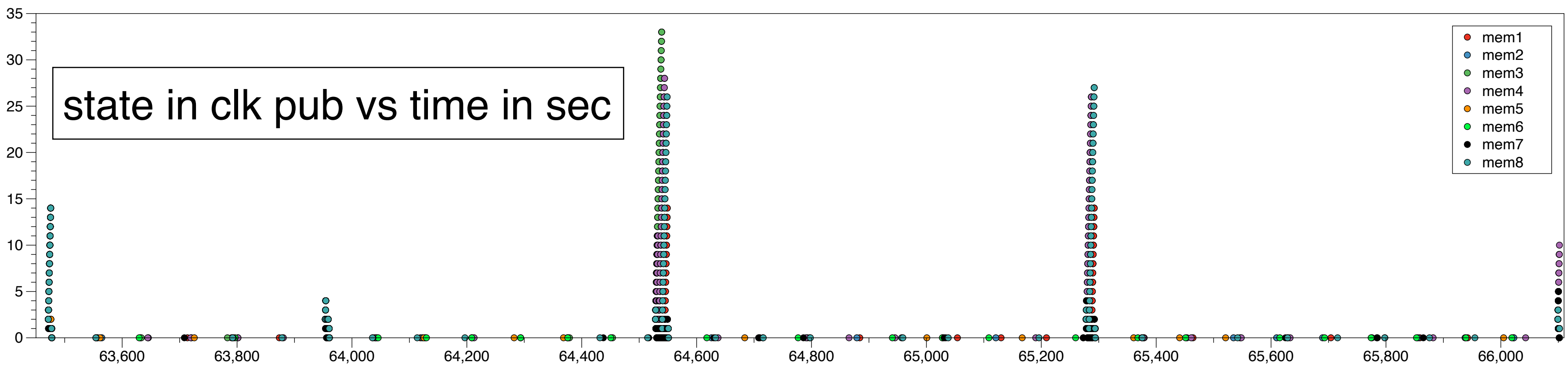
Split the members into two overlapping groups
(a hack that can be accessed by compiling with the MESHTEST=2 flag)





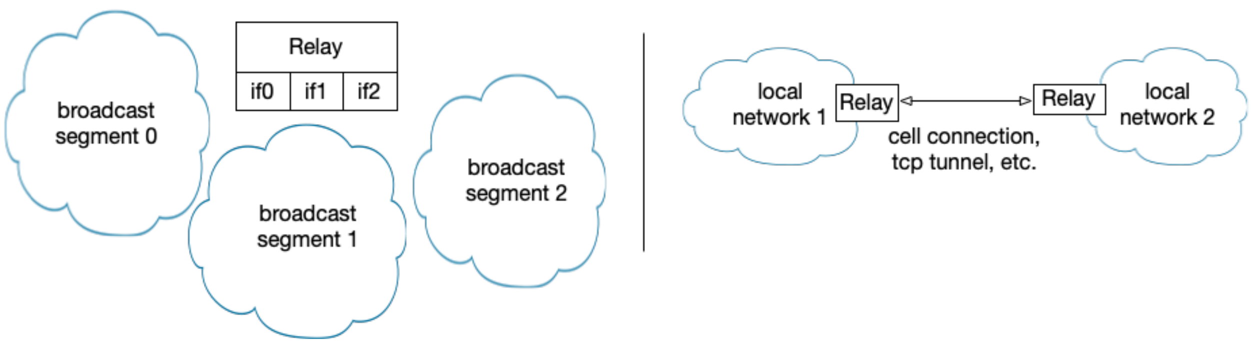
Next, order members so each talks to adjacent members only





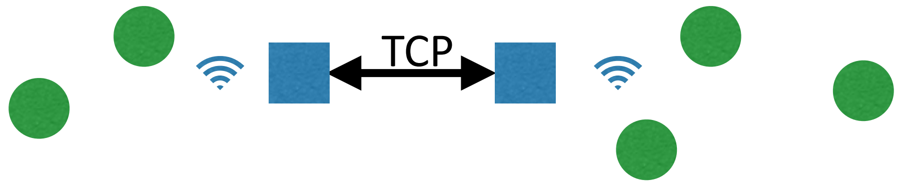
Relayed Trust Domains

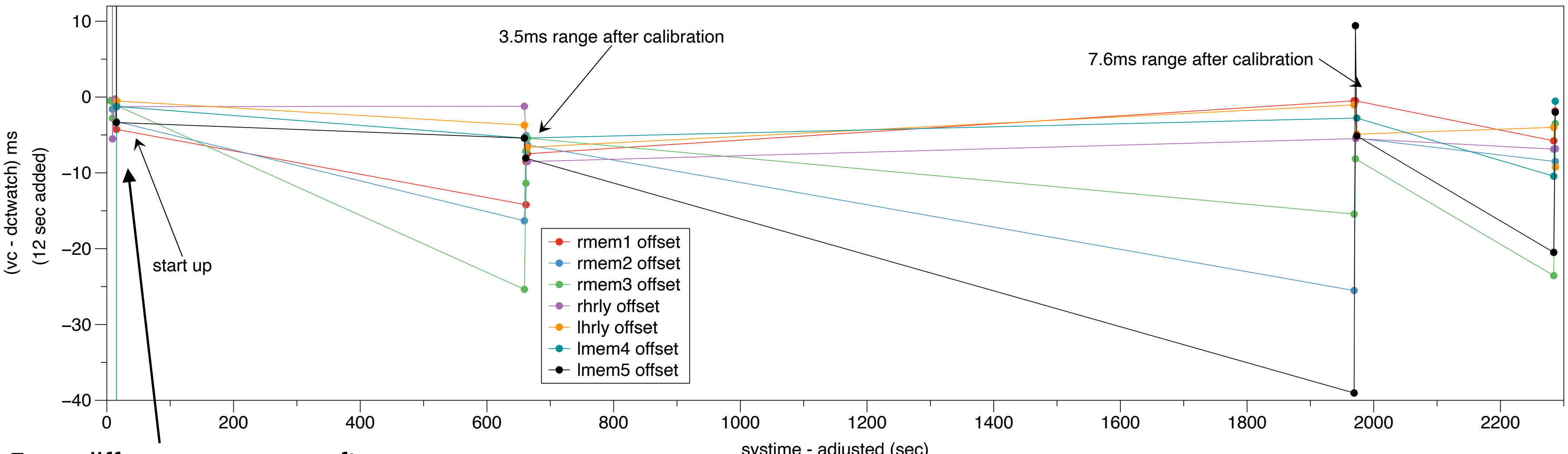
Relays are used to connect different subnets of a single Trust Domain. They have DeftTs in multiple subnets and move validated publications between them (PDUs are confined to a single subnet)



Relays act as a *single* node in the virtual clock distribution

Connect through relays starting right-side members and relay first

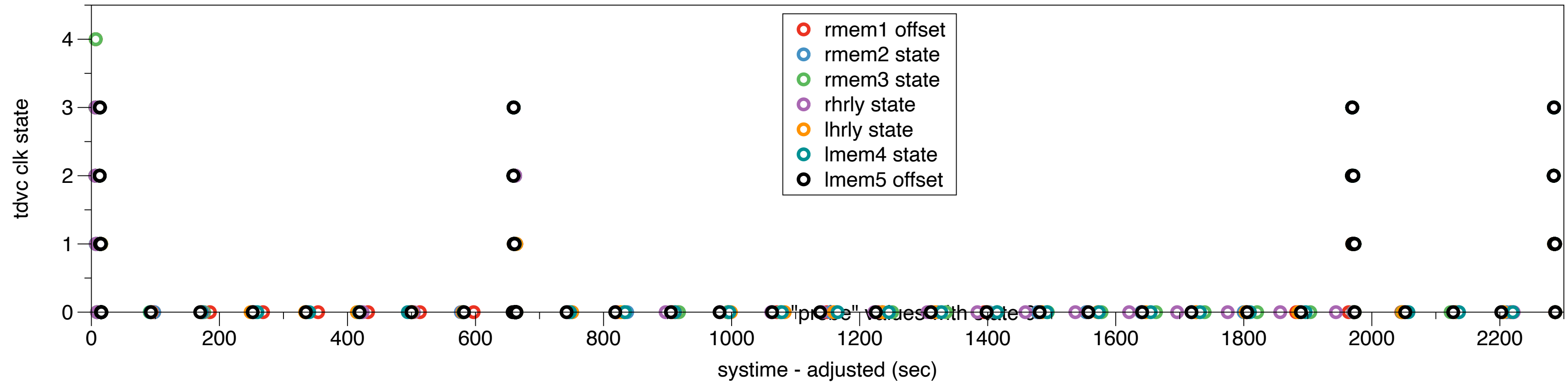




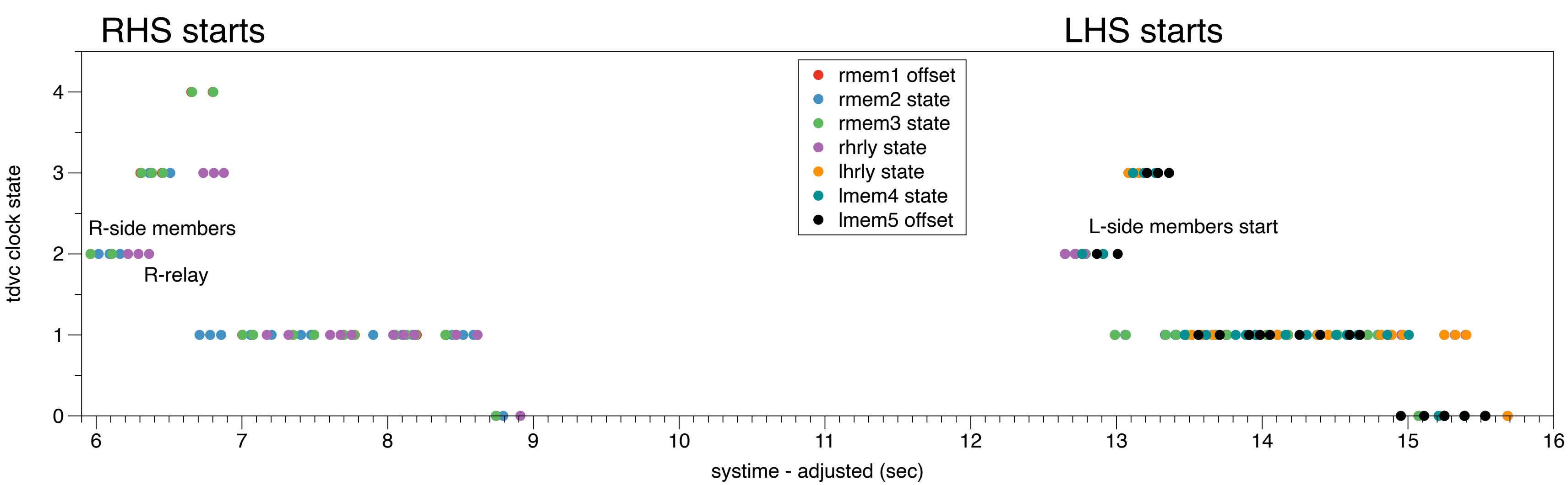
5ms difference range after calibration

- Calibration starts when a clk publication is received with a timestamp difference > 30ms
- The connecting lines help show where vc ended up after last calibrate

Corresponding values of calibration state in clk publications



Detail at start up



Next steps for TDVC distributor

- Larger scale testing will be coordinated by Operant Networks
  - testbed at a national lab
  - investigate convergence properties
  - investigate robustness to connectivity changes
  - investigate parameters
  - test the round-trip delay computation under range of conditions
- Integrate use of clock **capability** in identities for devices with access to accurate clocks
- Iterate core algorithm if needed.

What’s Going On?

(I am **always** trying to answer that question when I work on this stuff)

- Information-centric networking changes (and challenges) our perspective
  - Instead of monitoring connection state between two parties, ICNs communications are distributed, with multiple parties
  - In Defined-trust, communication state is the evolution of collections at individual members over time
- Diagnostic tools could use some work/thought
- **dctwatch** is a sort of tcpdump-for-DeftT to observe the on-the-wire(less) behavior (in tools subdirectory of github’s DCT)
- more recently, using a **logs** distributor, taking advantage of pub/sub:
  - modules log events and **logs** publishes them, e.g.:

```
logsCb_(crName("tdvc")/"cal"/m_certs[m_tp].name()[1]/m_certs[m_tp].name()[2]/m_nbrs, {});
```

results in:

```
22:01:35.311783  165.u  Ad  51975>56362  307 /^573c751de902b8f4/logs/^b7df1186
 | /iot2/tdvc/cal/device/sidedoor/6/25-07-16@20:01:35.311168
```

  - logs can be extracted from dctwatch or a subscriber can process

- Tools from distributed systems, like TSViz (https://bestchai.bitbucket.io/tsviz/), with ability to group/collapse members?

What Good is It?

What if I can use NTP and don’t care about having a domain-specific clock?

Making use of the features ICN approaches provide can let us solve old problems in new ways

The information the neighborhood RTD estimator collects can have other uses, e.g.:

- keep a list of (recent) neighbors and their delay-distance
- log this as /tdvc/nbrhd/<optional name tag>/timestamp with content of this list
- **logs** subscriber can track connectivity of the domain and its evolution with time

Other performance/measurement data may also be useful to collect on a “neighborhood” basis

Extras

Offset computation

1. Replicate (in a vector) each neighbor's quantized minimum clock difference values by its **nhSz** less the minimum nhSz value plus 1. Local clock difference is represented by 0s.
2. If the absolute value of all clock differences are less than tolerance, set **state**=1 and **nhSz** is equal to the number of distinct neighbors heard from plus 1 (for self). Go to step 8.
3. Increment **state**, rolling over to 2 at 255.
4. Set **adjust** to the statistical *mode* of the clock differences vector. If more than one value at the mode, **adjust** = smallest non-zero absolute value difference. In case of a negative and positive at that value, use the negative, i.e., move in the forward clock direction. (other tie-breakers are possible)
5. If **adjust** is zero, check for an *impasse* where **adjust** has been zero and not all neighbors are in tolerance for a several rounds (e.g., ten). If an *impasse* and there are negative difference values, **adjust** = smallest absolute value
6. Set **nhSz** to the number of neighbors whose clock difference values are within tolerance of **adjust** plus one (for self since **adjust** will put the local virtual clock in tolerance) If **nhSz** is equal to the total size of the neighborhood (all members heard from), **state** = 1 (in tolerance and counting neighbors)
7. If **adjust** != 0, add **adjust** to the running virtual clock **calibrate** for this calibration cycle
8. The next set of clock publications use the updated **state** and **nhSz** and subtract **calibrate** from the local virtual clock to compute the publication timestamp.
9. When all neighbors have been in state 1 for a set number of rounds (~3) *and* all neighbors are in state 0 or 1, calibration is complete, **state**= 0, local virtual clock -= **calibrate** and **calibrate**= 0

- Calibration restarts if an out-of-tolerance neighbor is detected (via periodic clock publications)
- Step 4 above can be implemented with approaches other than using the mode; e.g., a version using the median was also used but doesn't converge as well for non-fully connected networks

Collections and DeftT formats

cState

tdID	collection name	collection IBLT	nonce	lifetime

tdID: the first 8 bytes of the SHA-256 thumbprint of TD's schema cert
collection name: cert, msgs, keys (keys/pdus, keys/pubs), tdvc, logs
IBLT: run-length compressed IBLT of items (pubs) in the publisher's instance of this collection
nonce: four random bytes chosen by publisher
lifetime: time receiving member can hold this cState (in ms after arrival)

cAdd

tdID	collection name	csID	type	content	SigInfo	SigValue

csID: 32 bit hash of the cState responding to
type: cAdd (42)
content: contains one or more complete, valid pubs
SigInfo: contains SigType (same across a collection) and KeyLocator (if required)
SigValue: result of signing the cAdd using the indicated signing algorithm

Publication Names and Format

Publication name

app- and collection- specific components	timestamp

Publication

name	type	content	SigInfo	SigValue

timestamp: GMT style time point in microseconds (a number)
SigInfo: while cAdd can use non-origin specific algorithm like AEAD, pubs must be signed by originator

Collection names are in the cAdd and the richer descriptive names are in the Pubs. This could be “thought of” as a hierarchical name:

tdID	collection name	app- and coll-specific	timestamp

Publication names for the tdvc collection will be shown later. Although the need for and details of a Trust Domain Virtual Clock are specific to Defined-trust Communications, the exploitation of collection-based communication semantics to enable robust and efficient implementations may also apply to NDN/CCN