Fine-Grained Network Time Synchronization using Reference Broadcasts

Jeremy Elson, Lewis Girod, Deborah Estrin

Summary by Jonathan Ledlie on January 27, 2003

1 How RBS works

See Figure 1 for an illustration of these steps:

1. A transmitter broadcasts a reference packet to two receivers (i and j).
2. Each receiver records the time that the reference was received, according to its local clock.
3. The receivers exchange their observations.

2 Key Idea

Reference-Broadcast Synchronization (RBS) uses broadcast communication to allow the *receivers* of a synchronization message to synchronize clocks with one another. By removing several components of non-determinism (jitter) from traditional time synchronization (where receivers try to synchronize with the sender), Elson *et al.*achieve better accuracy than previous methods, *e.g.*, NTP.

RBS *requires* a physical broadcast channel and cannot be used in networks that employ point-to-point.

Nodes can synchronize time (1) relative to each other or (2) relative to an external timescale, *e.g.*, UTC. Many sensor applications require only relative time.

3 Precision Achieved

Using off-the-shelf 802.11 wireless Ethernet with 30 reference messages (more messages give better accuracy with diminishing returns):

- Within broadcast radius of sender: $1.85 \pm 1.28 \mu sec$,
- Within 4 hops of sender (*i.e.*, in multihop): $3.68 \pm 2.57 \mu sec$.

4 NTP

NTP works by clients synchronizing with a reference clock on a time server. The clients and server cooperate to determine the time it takes for a message to get from the server to a client using several exchanges of messages. Subsequently, clients deduct the delay from timestamped messages sent out by the server to set their local time. NTP has all four sources of nondeterminism and can be adversely affected by a busy network, as experiments in the paper show.

5 Sources of Time Sync Error

These times are all nondeterministic in that they may differ from one message to the next. RBS eliminates the nondeterminism caused by Send and Access. It can minimize Receive Time by having the kernel or NIC timestamp the message at a low level helps RBS be more accurate.

A comparison of the sources of error in a typical protocol (*e.g.*, NTP) and RBS is given in the paper and is duplicated here in Figure 2.

Send Time The time spent at the sender to construct the message; the time to send the message from the host to the network interface.

Access Time The delay incurred waiting for access to the transmit channel (*e.g.*, contention, control packets)

Propagation Time The time in transit from the sender to the receiver, once it has left the sender. This time is very small on a LAN, but can dominate the total delay in a WAN.

Receive Time The time needed to process a message once it has arrived at the receiver's network interface. This typically includes an interrupt.

6 Key Advantages of RBS

- The largest sources of nondeterministic latency can be removed from the critical path.
- Multiple broadcasts allow tighter synchronization because residual errors can be eliminated through statistical means.
- Outliers and lost packets as part of these multiple broadcasts can similarly be handled statistically.
- It can perform post-facto synchronization, where nodes determine the time an event has occurred after the event occurs, allowing them to save energy by letting their clocks drift.

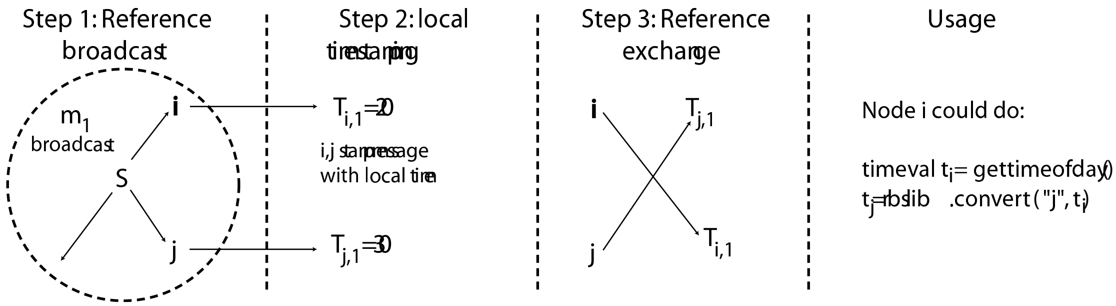
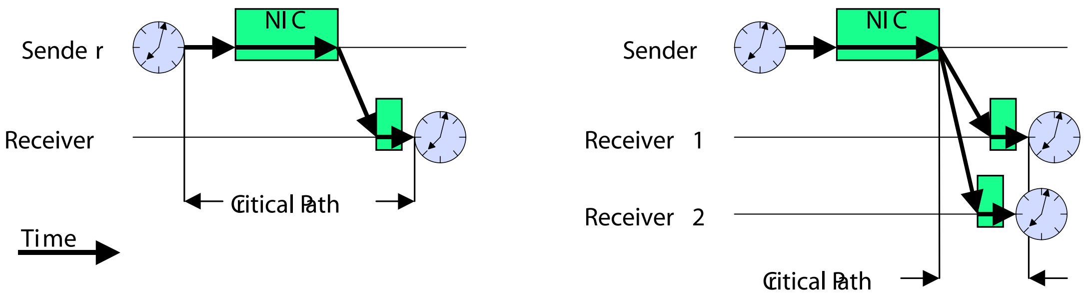


Figure 1: RBS Time Synchronization Example.



: A critical path analysis for traditional time synchronization protocols (left) and RBS (right). For traditional protocols working on a LAN, the largest contributions to nondeterministic latency are the Send Time (from the sender's clock read to delivery of the packet to its NIC, including protocol processing) and Access Time (the delay in the NIC until the channel becomes free). The Receive Time tends to be much smaller than the Send Time because the clock can be read at interrupt time, before protocol processing. In RBS, the critical path length is shortened to include only the time from the injection of the packet into the channel to the last clock read.

Figure 2: Critical Path for typical protocol on left and for RBS on right. By shrinking this amount of time, RBS reduces the amount of nondeterminism.

- Absolute and relative timescales can be achieved.

7 Details

Clock Skew Immediately after protocol, clocks are synchronized, but clock skew causes this synchronization to diverge. Using multiple reference messages allows RBS to both correct for nondeterminism in Propagation and Receive Time and for clock skew. Using least squares to get the best-fit line to the multiple references allows an estimate of a neighboring node's time that takes clock skew into account.

Post-facto Sync Instead of keeping clocks in sync when nothing interesting is happening, let them get out of sync. When an event occurs, extrapolate backwards (again using least squares to generate a line) to determine actual time of occurrence (relative to some node).

Absolute timescale Give some node the capability to keep accurate track of absolute time (*e.g.*, with GPS). Make nodes give time relative to that node.

Multihop RBS To compare two events on nodes i, j not within one multicast radius, include time conversion in the forwarding of a packet from i to j.

8 Related Work

- Lamport's relative clocks. Especially useful for relative ordering of events.
- Mills' NTP.
- GPS. Achieves 200 nsec relative to UTC, but requires sky view.
- CesiumSpray. Similar idea in same broadcast area, but uses different techniques for UTC and multiple broadcast.