Com S 611 Spring Semester 2007
Discrete Algorithms for Mobile and Wireless Networks

Lecture 7 Part B: Tuesday, 6th February 2007

Instructor: Soma Chaudhuri Scribe: Puviyarasan Pandian

1 Lecture topic

This lecture is about the presentation of the paper "Fine-Grained Network Time Synchronization using Reference Broadcasts" written by Jeremy Elson, Lewis Girod and Deborah Estrin. The presentation was done by Puviyarasan Pandian.

2 Introduction

Time synchronization is essential in distributed systems and the Network Time protocol is used for this purpose across systems in the Internet. In wireless sensor networks and actuators, the precision requirement for time synchronization is in the order of μ-seconds. Various applications of these networks like some domain specific applications such as measurement of time-of-sound, forming a low-power TDMA radio schedule, integration of a time-series of proximity detections into a velocity estimate, suppressing redundant messages by recognizing duplicate detections of the same event by different sensors, distributing a beamforming array or sensor network applications like coordinated future action, secure cryptographic schemes, ordering logged events during system debugging and so on, require such high precision.

The traditional method of time synchronization involves a server sending a beacon with its clock value in it to a set of clients and the clients synchronize their clocks based on the received clock value. Reference Broadcast Synchronization comes up with a fundamentally different design where the receivers synchronize each other based on the arrival time of a broadcasted message at the receiver. Thus the limitation of this technique is that it requires a network with a physical broadcast channel.

3 Related work

Lamport's paper on the importance of *virtual clocks* in systems where causality is more important than absolute time can be considered as the first landmark paper in computer clock synchronization. A lot of research has happened in clock synchronization algorithms in networks since then. Most of these algorithms follow the traditional synchronization technique where a simple connectionless messaging protocol is used for the exchange of clock information among clients and one or more servers, methods to mitigate the effects of nondeterminism in message delivery and processing and an algorithm in the client for updating local clocks.

Mill's Network Time Protocol(NTP) though has features like highly scalable, self-configurable in multi-hop networks and robust, provides not the sufficient precision needed for wireless adhoc network applications. Global positioning system(GPS) also provides synchronization but only in open space with a clear sky view. CesiumSpray system(by Verissimo and Rodrigues) and the 802.11-based broadcast synchronization scheme(by Mock et al) come close to RBS but are restricted to single-hop networks. The precision achieved by Liao et al is similar to RBS in the μ-second range but the technique depends on the network topology's latency and determinism guarantees. Further, Hill et all and Ganeriwal et al came up with another microsecond precision algorithm but by using a tight integration with the MAC layer.

4 Traditional Synchronization Methods

The existing synchronization mechanisms normally follow the traditional mechanisms described above. Extending it further is to use the total round trip time in synchronization by sending a client message following the server's clock message. A proper analysis of the sources of error in this mechanism will give us a good idea on how RBS works better.

4.1 Sources of Error

Nondeterminism in the synchronization mechanism is the main source of error. This is due to the random events happening while the message is sent, transmitted or received. Accordingly the latency in message transmission can be decomposed into

1. *Send Time*, the latency caused at the sender's side when the message is being constructed. This can be due to the load in the sender's system or some queueing technique employed in the network layers or due to context switches in the sender's operating system.

2. *Access Time*, the latency caused when the sender waits for access to the transmission medium to send the message. This can be due to the MAC layer protocols in use.

3. *Propagation Time*, the latency due to the message transmission from sender to receiver in the transmission channel. Normally this can be assumed to be zero in a network where the sender and receiver share the same media, since this delay is in the order of nanoseconds and we are working for a microsecond precision.

4. *Receive Time*, the latency at the message reception side to receive the message and notify the host of the message.

   Existing methods work on reducing these sources of error by increasing the number of message transmissions to better estimate the round trip delay and accordingly synchronize the clocks whereas RBS uses the property of broadcast messages where a broadcasted message that is received by two receivers have the same latencies for send and access times and thereby removing these errors when synchronization is done with respect to this broadcasted message as shown in the Figure 1.

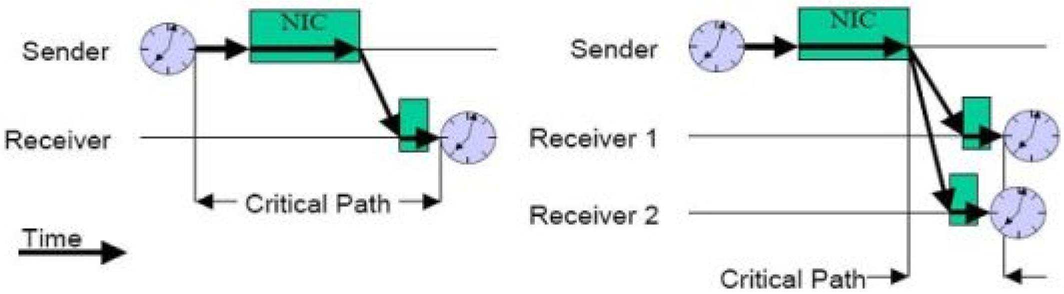

Figure 1: Critical paths of Traditional Synchronization mechanism(left) and RBS(right)

5 Reference Broadcast Synchronization

Let us consider a simple model where there is just one transmitter and two receivers and state the algorithm for the same.

5.1 Algorithm

1. The transmitter transmits a reference broadcast packet to both the receivers.
2. Each receiver records the time at which this packet was received in its local reference clock.
3. The receivers exchange their observations. Based on the exchanged observations the receivers synchronize their clocks.

For example, consider a node i at position (0,0) and it detects a target at time $t = 4$ seconds, and a node j at position (0,10) detects the same target at time $t = 5$ seconds, then based on these information we can conclude that the target is moving North at a speed of 10 units per second.

5.2 Performance dampeners

A relative local synchronization scheme like this is affected by the number of receivers in the system to be synchronized, the nondeterminism of the receivers and the clock skew between the receivers. The algorithm proposed above can be extended further to account for these dampeners as follows.

- The nondeterminism at the receivers can be taken care of by considering the receiver error to be Gaussian(Empirical studies on the receiver error have shown them to be Gaussian.) and thereby use more than one reference for a statistically precise synchronization.

  In this model, a transmitter broadcasts m packets and each node(r) of the n receivers notes the time of receival of these packets(b) based on the reference clock($T_{r,b}$). The offset of a receiver i to any other receiver j can be computed as $\frac{1}{m}\sum_{k=1}^{m}(T_{j,k} - T_{i,k})$.

- It is always possible that clocks at different nodes tick at different rates due to the oscillator characteristics such as lack of absolute accuracy and short-term or long-term frequency instability. Thus considering clock skew into the picture, we can calculate the offsets from multiple observations as a least-squares linear regression method, which will find the best fit line through the several phase observations over time.

RBS has been tested on Berkeley Motes and is found to have synchronization of clocks within 11μ-sec. In Compaq IPAQs using a 11 Mbit/sec 802.11 network, a synchronization of $6.29 \pm 6.45\mu$-sec, 8 times better than NTP has been achieved. With Kernel timestamps in the receiver end, a precision of 1.85 ± 1.28 μ-sec was obtained.

6 Multi-Hop Time Synchronization

The RBS algorithm so far discussed works well for single domain networks. This mechanism can be extended for multiple domains as well. Let us consider an example of clock conversion in a simple multi-domain topology as shown in Figure 2

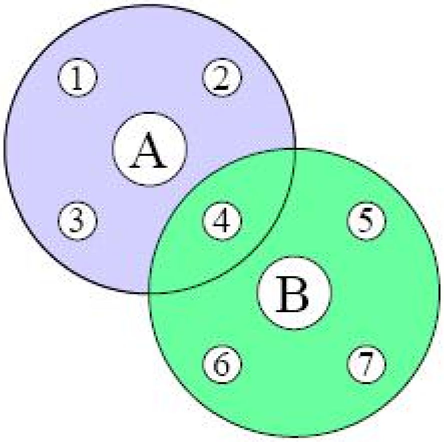

Figure 2: A simple topology.

In Figure 2, Nodes A and B send sync pulses at times as P_A and P_B, establishing two locally synchronized neighborhoods. Receiver 4 hears both A and B, and can relate nodes in either neighborhood to each other. Let us consider two events that are tracked at node 1 and node 7. Node 1 finds that event E_1 occured 2 seconds after P_A. Node 7 receives event E_2 4 time units after P_B. Now, to synchronize across these domains, these nodes contact node 4 to know that P_A has occurred 10 seconds after P_B. Hence,

$$\begin{aligned}
E_1 &= P_A + 2 \\
E_7 &= P_B - 4 \\
P_A &= P_B + 10 \Rightarrow \\
E_1 &= E_7 + 16
\end{aligned}$$

6.1 Algorithm

Writing this down as a formal algorithm, the notation we use is $E_i(R_j)$ to denote the time of event i occurred at receiver j.

1. Nodes R_1 and R_7 observe their corresponding events E_1 and E_2 at times $E_1(R_1)$ and $E_2(R_7)$
2. A best-fit line has been established at R_4 as described in Section 5.2. Using this method, $E_1(R_1)$ is converted to $E_1(R_4)$.
3. R_4 uses the similar line for B's broadcast to convert it to $E_1(R_7)$
4. Now the recorded times of the two events are as per the clock of R_7 and can be compared by $E_1(R_7)$ - $E_2(R_7)$.

6.2 Multihop Networks

Extending this multihop algorithm to more complex network topologies, let us consider the topology shown in Figure 3.

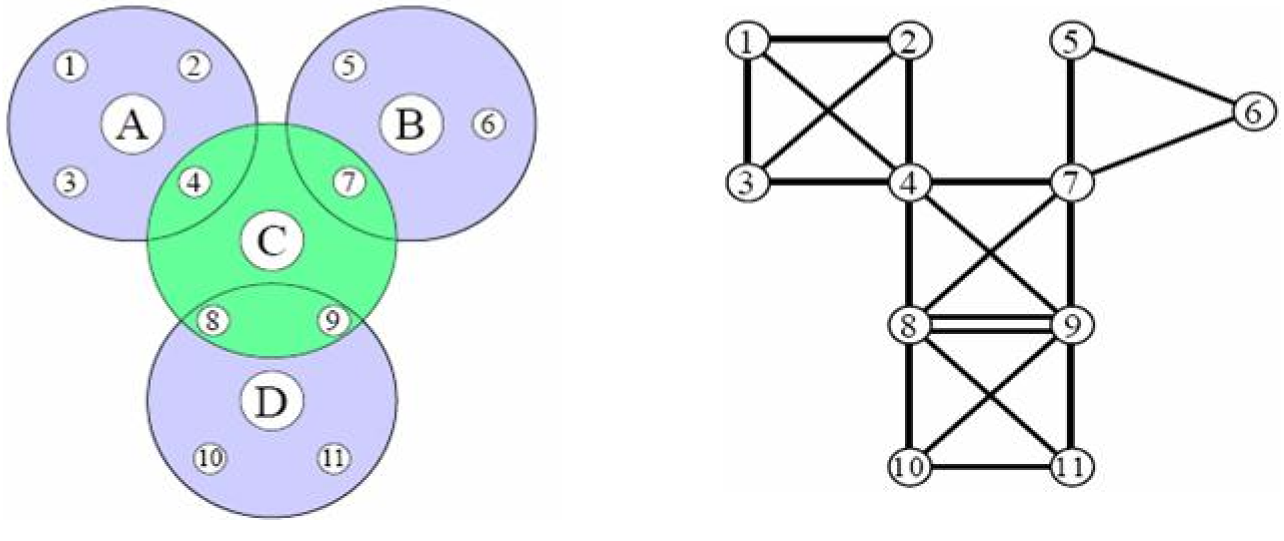

Figure 3: A multi-hop network topology. Physical topology(left) and its corresponding logical topology(right).

Each transmitter(lettered nodes) of a synchronization pulse establishes its own neighborhood of receivers(numbered nodes) as shown in Figure 3. This physical topology can be visualized logically as a graph with the receiver nodes as vertices of the graph. There exist an edge between any two vertices in the graph if the corresponding nodes receive sync pulse from the same transmitter which enable them to relate their clocks. Now a time conversion from one clock to another in the network is a simple path between the corresponding nodes in the graph.

For example, we can compare times of $E_1(R_1)$ and $E_{10}(R_{10})$ by $E_1(R_1) \rightarrow E_1(R_4) \rightarrow E_1(R_8) \rightarrow E_1(R_{10})$. A shortest-path search algorithm like Dijkstra's algorithm can be used here for the time conversions. The edges can be weighted also based on the quality of conversions obtained due to the RMS error of the best-fit line.

This approach may not be suitable for large networks since the need for global information of the whole network for constructing the logical topology may not be available always. Just like in the Internet's link-state routing algorithm, a localized approach can be employed where time conversion is built into the packet forwarded from node to node based on the time of the local clock at each hop. RBS can also be used to synchronize with external consistent timescales by connecting one of the nodes with this timescale and synchronizing from this node.

In this multihop technique, the precision is found to decay slowly with an average error for an n-hop network being proportional to $\sqrt{n}$. RBS has also been found to work well with several real time applications for applications such as accoustic time-of-flight ranging, autonomous position location and blind beamforming on accoustic signals.

7 Conclusion & Future work

RBS has been implemented and tested in different environments and hardware and has been found to achieve a μ-sec precision clock synchronization. It is also proven to be robust, reliable and resource-efficient for both performance measurement and for real-time applications than traditional algorithms. Further research can be done on automatic detection of beacon senders, multiple beacon-senders in a single neighborhood, addressing several performance issues and tradeoffs such as improving precision vs bandwidth conservation in the presence of redundant beacons, minimum time to acquire synchronization to within a given precision bound, precision vs frequency of beacons and so on.