Understanding about time synchronisation

Contents

I. Why ? 1

II. How ? 1

A. How to communicate the time 1

1. Advantage 1

2. Disadvantage 2

B. How to take the time 2

1. Hardware Clock model 2

2. Software clocks 2

III. Synchronization technique 2

I. Why ?

For example, the times of occurrence of physical events are often crucial for the observer to associate event reports with the originating physical events. Physical time is also crucial for determining properties such as speed or acceleration.

In sensor networks, many sensor nodes may observe a single physical phenomenon. One of the key functions of a sensor network is hence the assembly of those distributed observations into a coherent estimate of the original phenomenon. This process is known as data fusion.

Time is also a valuable tool for intra-network coordination among different sensor nodes.

II. How ?

A. How to communicate the time

NTP was designed for large-scale networks with a rather static topology (such as the Internet). Nodes are externally synchronized to a global reference time that is injected into the network at many places via a set of master nodes.

1. Advantage

The operation of NTP is independent of the underlying physical network topology. In the NTP overlay hierarchy, a master and a client can be separated by in the physical network.

NTP would have to operate with a single master node, which uses its local time as the reference time.

2. Disadvantage

With NTP all nodes would be continuously synchronized with maximum precision. High consumption in energy.

B. How to take the time.

How to keep the time in synchronization with the master ?

1. Hardware Clock model

Digital clocks measure time intervals. They typically consist of a counter h (which we will also refer to as “the (local) clock”) that counts time steps of an ideally fixed length.

The rate is assumed to be constant. This is reasonable if the required precision is small compared to the rate fluctuation. We have a little derivation that we can’t predict.

2. Software clocks

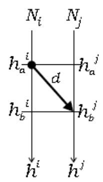
A software clock is a function taking a local clock value h(t) as input and transforming it to the time c(h(t)). This time is the final result of synchronization, and we therefore call it the synchronized time. For example, c(h(t)) = t0+h(t)−h(t0)

III. Synchronization technique

When data is sent you have a delay between the message sent and the message received.

How the sensors can find the current timestamp with the timestamp send? It depends of the architecture and of the method of communication.

Example of unidirectional communication:



If a-priori bounds on the message delay are known, that is dmin< d<dmax, then the estimation, you can say that h_b=h_a+0.5(dmax-dmin). You add the mean of the delay.