Section 3: Week 6: Clock Synchronization

Nate Bachmeier

TIM-8120: Distributed Systems

October 27th, 2019

North Central University

# Clock Synchronization

A common challenge of presenting a single system image across multiple nodes is ensuring that the sequence of events processed, is correct. Many strategies to address this issue focus on time stamps, though this approach has challenges as most clock implementations are imperfect. Clock skew causes these imperfections and needs Clock Synchronization Protocols to mitigate.

# Physical Clocks

Most mechanical devices have timers, not clocks. These timers rely on the oscillation of quartz crystal as a means to decrement a counter, signifying the duration until the clock tick system interrupt occurs (Ubolkosold, Knedlik, & Loffeld, 2005). These crystals “suffer from large frequency shifts due the the high sensitivity to external or internal factors (Zhang, et al., 2019).” According to Zhang et al., atomic clocks rely on energy transitions to achive 100,000 times better precision than quartz after ten days. These more precise instruments are difficult to deploy more broadly due to both their physical size and power consumption. Researchers are continuing to explore methods to reduce these limitations.

As these timers raise clock ticks, the more extensive system needs to report that it is relative to some universal reference point. Universal Coordinated Time (UTC) is the de-facto solution as alternatives, such as regional time zones, are impacted by political cross cutting concerns. For example, Eastern Time in America changes at different points of the year for day light savings. Another solution is to rely on the local time zone of the device. This design introduces additional challeges for systems spanning multiple locations as the example value “2019-10-27 10:03:00” occurs numerous times.

# Clock Synchronization Algorithms

## Network Time Protocol (NTP)

Many distributed systems rely on an NTP service as their centralized source of truth. The service uses a hierarchial solution to replicate the observed time from more precise sources. For instance, a parent server might rely on Global Positioning Services (GPS) using expensive hardware. Once the GPS value is known, all subscribers can receive or forward it across the network in an economical manner.

## Berkeley Algorithm (BA)

The BA algorithm starts with a group of peers performing an election process to choose a master. The master periodically polls the slaves for their local time, then adjustes itself based on the average value after accounting for round trip time (RTT) (Gusella & Zatti, 1989).

## Reference Broadcast Synchronization (RBS)

RBS relies on a beacon to periodically broadcast reference packets to receivers, which compares the arrival time against the local time (Akbar, Ichsan, & Darmawan, 2019). The beacon message contains a sequence number, not a timestamp. The receiver then compares the reference message timestamp against its peers.

## Logical Clocks

Many scenarios need to know the relative offset of correlated messages, not necessarily the UTC of the operation. For these scenarios, a simple incrementer can be sufficient to denote the sequence. Alice sends a message (with counter=0) to Bob, then Bob forwards then message (with counter=1) to Charlie.