CS268 Project Proposal - Synchronization of Clocks in Wireless Networks

Vasuki Narasimha Swamy, Jessica Ko

The “Tactile Internet” promises to introduce several new emerging technology market opportunities as well as new kinds of public services. Immersive and interactive applications such as robotics, gaming, smart-healthcare, and smart grid all could involve precise H2M and M2M interaction. The Internet of Things (IoT) envisions a heterogeneous network where devices are connected seamlessly via the technology best suited for their needs. To enable these Tactile Internet and high-performance IoT applications, ultra-reliable communication with latencies of about 1ms is crucial. This domain is largely unexplored and the techniques used by existing standards are fundamentally ill suited for low-latency and high-reliability.

An existing domain which parallels these requirements is high performance industrial control systems. As no existing wireless technology can support these requirements, wired communication protocols (Fieldbus) are used. They support short messages (10s of bytes) to/from closeby sensor/actuators (10s - 100s) delivered regularly (100s - 1000s of times per second). The protocols are ultra-reliable (probability of a single packet delivery failure of 10^{-8}) with very low latency (a couple of ms). Synchronized cooperative communication based wireless protocols proposed in [1], [2] achieves QoS (high-reliability and low latency) similar to wired fieldbus systems by exploiting multi-user diversity and distributed space-time coding to achieve. One of the main requirements for protocols aiming at high-performance IoT applications is tightly synchronized clocks to avoid collision and thus avoiding any decoding errors which lead to violation of latency requirements. We aim to address this problem in our project.

The goal of this project is development and analysis of protocols to achieve clock synchronization in a wireless network while tolerating a small (predefined) amount of error. We model the time at each node n (T_n) to have two components: the time from the oscillator ($T_{n,o}$) and the correction time ($T_{n,c}$) such that $T_n = T_{n,o} + T_{n,c}$. We call T_n to be the virtual clock time of node n. We want the virtual clock time differences between any two nodes m and n to be within a tolerable limit say T_ϵ i.e., $|T_m - T_n| \leqslant T_\epsilon$. The target T_ϵ is in the order of 100ns.

There is no dearth of literature in this field. [3] takes a theoretic look at this problem under almost ideal conditions where the clocks start at a random time but are ticking at the same frequency. They introduce uncertainty in message delivery time which results in a fundamental achievable lower limit in how well synchronized the clocks can be. The seminal work “Network Time Protocol”(NTP) [4] deals with synchronizing over a very large network (basically the internet) in a hierarchical fashion. The offsets achieved by this protocol is in the order of hundreds of milliseconds and is not fully suitable for control applications. [5], [6] look at a reference broadcast based protocol which is a significatn improvement over NTP as the offsets are in the order of microseconds and also deal well with skews (difference of clock frequencies). Works like [7]–[10] prove fundamental limits in clock synchronization along with stating that estimating all unknowns like clock offsets and link delays is impossible.

We envision three ways of approaching this problem: a centralized (beacon oriented) solution, a distributed solution, and a hierarchical solution. Each solution is well suited for different scenarios. For example, a centralized (and hierarchical) solution works better when most clocks are not synchronized. But when few nodes are not synchronized, a distributed or hierarchical synchronization would be more suitable. We will consider various jitter and drift models for clocks and address both the transient as well as steady state dynamics of the clocks. We will analyze these methods both theoretically as well as through simulations. We expect to have a few fundamental theorems under suitable assumptions and model based simulation results, for example number of rounds of communication it took for the network to reach equilibrium and how the network responded to different clock events (jitter vs drift). **If time permits**, we will try to build a protocol that piggybacks over WiFi.

References

[1] V. Narasimha Swamy *et al.*, "Cooperative communication for high-reliability low-latency wireless control," in *IEEE International Conference on Communications (ICC)*, 2015.

[2] ——, "Network coding for High-Reliability Low-Latency wireless control," in *IEEE Wireless Communications and Networking Conference: Workshop on 5G & Vertical Industry*, Doha, Qatar, Apr. 2016.

[3] J. Lundelius and N. Lynch, "An upper and lower bound for clock synchronization," *Information and control*, vol. 62, no. 2, 1984.

[4] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, 1991.

[5] J. Elson *et al.*, "Global synchronization in sensornets," in *LATIN 2004: Theoretical Informatics*, 2004.

[6] ——, "Fine-grained network time synchronization using reference broadcasts," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, 2002.

[7] N. M. Freris *et al.*, "Fundamentals of large sensor networks: Connectivity, capacity, clocks, and computation," *Proceedings of the IEEE*, vol. 98, no. 11, 2010.

[8] N. M. Freris and P. Kumar, "Fundamental limits on synchronization of affine clocks in networks," in *46th IEEE Conference on Decision and Control*, 2007.

[9] A. Giridhar and P. Kumar, "Distributed clock synchronization over wireless networks: Algorithms and analysis," in *2006 45th IEEE Conference on Decision and Control*, 2006.

[10] N. M. Freris *et al.*, "Fundamental limits on synchronizing clocks over networks," *Automatic Control, IEEE Transactions on*, vol. 56, no. 6, 2011.