NS-3 Based IEEE 1588 Synchronization Simulator for Multi-hop Network

Tatsuya Maruyama*, Tsutomu Yamada*, Shouji Yoshida†, Mitsuyasu Kido†, Chikashi Komatsu†
*Hitachi, Ltd., Research & Development Group,
Center for Technology Innovation - Energy
1-1, Omika-cho 7-chome, Hitachi-shi
Ibaraki-ken, 319-1292 Japan
Email: {tatsuya.maruyama.db, tsutomu.yamada.bs}@hitachi.com
†Hitachi, Ltd., Infrastructure Systems Company
2-1, Omika-cho 5-chome, Hitachi-shi
Ibaraki-ken, 319-1293 Japan
Email: {shoji.yoshida.bw, mitsuyasu.kido.pq, chikashi.komatsu.tw}@hitachi.com

***Abstract*—Network time synchronization using IEEE 1588 has tended to be applied to a wide network(i.e. a multi-hop network) with the aid of IEEE 1588 version 2 enhancement. However, designing a system to fulfill the desired synchronization accuracy is difficult because there are many factors that have an impact on multi-hop networks. Therefore, we developed a time synchronization simulator based on the network simulator 3 (NS-3) by developing a clock conversion model, IEEE 1588 simulation model, and application model with activation coordinated with synchronized time.**

I. Introduction

Introduction of IT led by web technologies brings higher performance and lower cost operation to control systems. IEEE 1588, which is compatible with IEEE 802.3, has been standardized for time synchronization, which is one of the control system requirements. The IEEE 1588 specification recommends hardware-based timestamp implementation to achieve highly accurate synchronization while conventional network synchronizations (e.g. NTP) implement software-based timestamps. Recently, IEEE 1588, which originated from time synchronization for a local system, has tended to be applied to a wide area network(ver.2[1]) due to the fact that network synchronization applications for, for example, power utilities and telecommunication require operations on a wide area network.

Designing a system to fulfill the desired synchronization accuracy is difficult because there are many factors that have an impact on multi-hop networks. The following three multi-hop network parameters affect synchronization accuracy.

Network-related parameters
: The number of nodes, network topology, number of network switches, distance between nodes and network switches, etc.

Node-related parameters
: Communication traffic, processing time of hardware and software, oscillator accuracy, synchronization algorithm, etc.

Communication-method-related parameters
: Communication throughput, 1 or 2 step mode of IEEE 1588, execution period of synchronization protocol, communication scheduling method such as time division multiple access (TDMA), etc.

Therefore, we developed a network simulator that can assess synchronization accuracy by enhancing NS-3.

II. Requirements

We summarize the network simulator requirements for our purpose. By taking into account function and performance, the following characteristics are required for multi-hop network simulation enabling IT communications.

- Supported communication protocols
  - Existing protocols like IP family.
  - IEEE 1588.
- Supported evaluation items
  - Communication performance(bandwidth, delay).
  - Synchronization accuracy.
- Application model activation in relation to synchronized time
- Simulation speed

For the function, we considered the requirements in the utilization procedures of the simulator: design, execution, and evaluation. In the design, the IP family and IEEE 1588 were selected as the supported communication protocols in accordance with the recent trend of introducing IT to control systems. For example, IEC 61850, which is a communication protocol for power utilities, may use TCP/IP for control and IEEE 1588 for time synchronization[2]. In the execution, to simulate synchronization among nodes, the behaviors of each node were activated in relation to the synchronized time on each node. This also requires synchronization accuracy for an evaluation item of the simulator. For the performance, the simulation speed that is a typical requirement of a network simulator was selected.

As the authors knew, there were no existing network simulators to satisfy these requirements, so this was our

978-1-4673-7595-5/15/$31.00

motivation for developing a network simulator. Developing one from scratch was an option, but enhancing an existing network simulator was preferable because an existing network simulator already supports existing protocols (i.e. IP family) and evaluation items (i.e. bandwidth and delay).

III. NS-3 ENHANCEMENT

The selection of a network simulator is described, and we selected NS-3. The characteristics of NS-3 and our modification to enhance it to achieve time synchronization are also described.

A. Network simulator selection

We decided to enhance an existing network simulator and compare it with open source network simulators: NS-2 [3], NS-3 [4], OPNET, and OMNeT++[5]. The reason open source simulators were candidates was to enable the enhancement of the functions.

All candidate simulators supported existing protocols and evaluation items. On the other hand, no simulators supported IEEE 1588 protocol and synchronization evaluation. We selected NS-3 from the above-mentioned network simulators because of its advantages such as high development efficiency, high simulation speed, and low memory usage [6].

B. NS-3

NS-3[4] is a digital event simulator that executes registered events serially. The user should prepare the following two models for NS-3 simulations.

- Object models
- Scenario model

Object models refer to communication protocols like IP family, application models, and communication devices like nodes, network devices, and communication channels. The user can develop an object-oriented model using NS-3 and extend an existing object model to the new protocol model. The scenario model refers to the simulation configuration to specify the layout of these object models like network topology, the number of network devices attached to a node, and the execution timing of application models.

The user can customize the communication environment by changing the parameters of the number of nodes, delay on communication channels, and so on. Basic communication protocols and general application models are offered by NS-3.

C. Clock conversion model

Next, we describe NS-3 modification to simulate synchronization accuracy. Upon evaluation of synchronization accuracy, a common challenge to network simulators including NS-3 is that the time of each node (the local time) is not modeled. Only the common time for the whole simulation (the simulator time) is defined.

That is why we modeled the local time. In the simulator, because the simulator time was the only variation parameter that changed with time, we decided to model the local time based on the simulator time. In addition, time modeling at each node was not limited to the uniform model. A simple modeling was just a constant offset to the simulator time, but complex modeling needed to take into account clock drift including the effect of temperature. For example, temperature affects not only the clock frequency but also the frequency drift even though the same clock source is used[7]. Thus, the clock model has to be changed in accordance with the desired synchronization accuracy, the simulation conditions such as temperature, and the others. Considering these requirements and the advantages of NS-3 object-oriented modeling, we defined the clock conversion model between two time domains (Fig.1). The conversion model offers methods to convert a time to the other time domain.

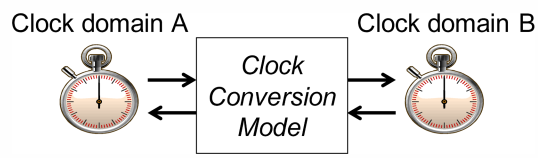

Fig. 1. Clock conversion model

The local time at each node can be modeled by extending the clock conversion model. The clock conversion model also converted not only between the simulator time and local time but also between the local time and the system time, which is synchronized time to a master by a slave.

D. IEEE 1588 Model

Next, the development of IEEE 1588 protocol model is described. The function structure of IEEE 1588 ordinary clock is shown in Fig.2.

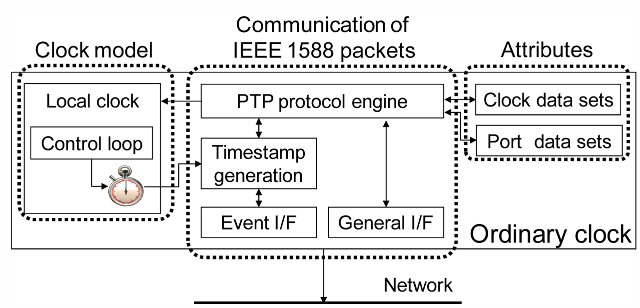

Fig. 2. Ordinary clock model

The ordinary clock comprised a clock model, IEEE 1588 communication function, and attributes.

The layer structure of the developed model is shown in Fig.3.

We placed two clock conversion models between the simulator time and local time and between the local time and system time for clock models. Then, we developed SyncAlgorithm class modeling, which is a synchronization algorithm to control the parameters of the clock conversion model between the local time and system time. The new synchronization algorithm can be modeled by extending the SyncAlgorithm class.

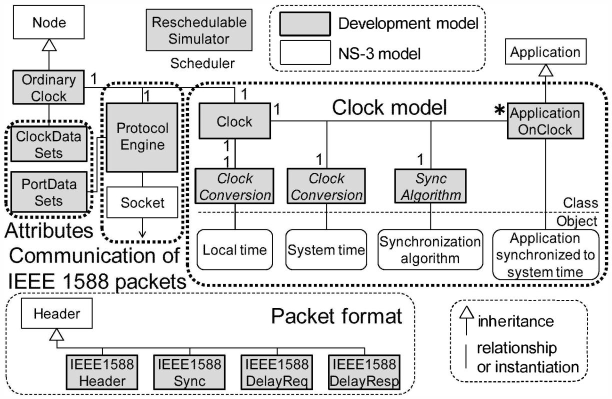

Fig. 3. IEEE 1588 Model

IEEE 1588 communication functions were developed as an IEEE 1588 master and slave model. These models communicated with IEEE 1588 packets and informed the SyncAlgorithm object of the obtained time information. At the moment, Announce message and best master clock algorithm to determine the master have not been implemented, and this is our future work.

IEEE 1588 specification recommends the timestamp at the network interface, and then we developed the IEEE1588 network device model by extending the Ethernet model of NS-3 (Fig.4).

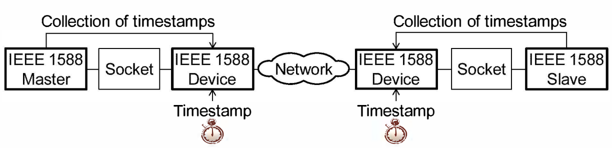

Fig. 4. IEEE 1588 network device model

E. Application model activation coordinated with synchronized time

Next, we describe the control of communication timing, which was coordinated with synchronized time. In NS-3, which is a digital event simulator, the combination of event processing and event activation time in simulator time is registered as an event in the event queue. To control communication timing coordinated with the time domain of each node, it is preferred that the event activation time is specified by the node's own time domain: local time (inherent time of the node) or system time (synchronized time to the master). We developed this function as the ApplicationOnClock class inherited from the normal application model, which is called the Application class. The Application class is a basic class to simulate a user application, so the ApplicationOnClock class is a user application in relation to the node's own time. When the ApplicationOnClock class is registered as an event, the specified time in the node's own time domain is converted to the simulator time (i.e. the common time for the whole simulator).

To achieve this is a challenge. When the simulator is able to express node-specific time with the clock conversion model, events on a node are registered with the conversion of execution start time (system time) to simulator time. At this time, update of the time by synchronization means a change in node-specific time. That is, the registered events on the node have to be rescheduled.

To enable this, we developed the ReschedulableSimulatorImpl class by enhancing the rescheduling function to the DefaultSimulatorImpl class offered by the NS-3. The rescheduling function takes the following steps when time synchronization occurs (Fig.5).

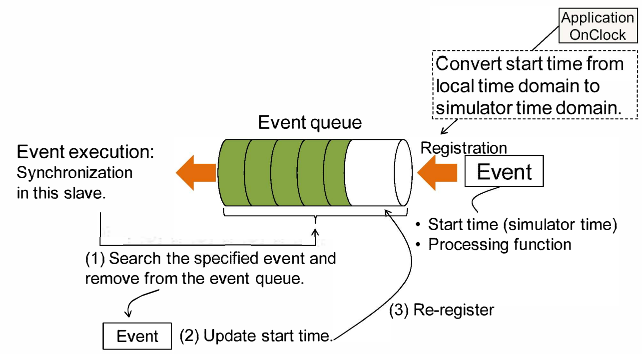

Fig. 5. Rescheduling steps

1) Remove the events tracked by the Clock object from the event queue.
2) Update the execution start time of the event.
3) Register the events to the event queue again.

These extensions (clock conversion model, IEEE 1588 protocol models, and application model activation coordinated with synchronized time) enabled us to simulate time synchronization.

IV. Evaluation

To evaluate the developed simulator, we compared a simulated result and an experimental result for a simple topology. We also simulated synchronization on a multi-hop network and measured simulation speed.

A. Comparison between simulation and experiment

First, we compared a simulation result and an experimental result for a simple topology where a master had connected two slaves via a switch. The oscillator accuracy was 50 ppm, execution period of synchronization was $2\,ms$ (i.e. a Sync message was transmitted every $2\,ms$), and a 1 Gbps IEEE 802.3 was used as the network. The difference was measured every $1\,ms$ in both the simulation and experiment. The switch was modeled as normal distribution with $\pm 500\,ns$ for the difference between the forward path and the backward path based on preliminary measurements.

Fig.6 and Fig.7 show the result.

Both of them plot the measure points within $\pm 500\,ns$, and so they could be considered relatively equivalent.

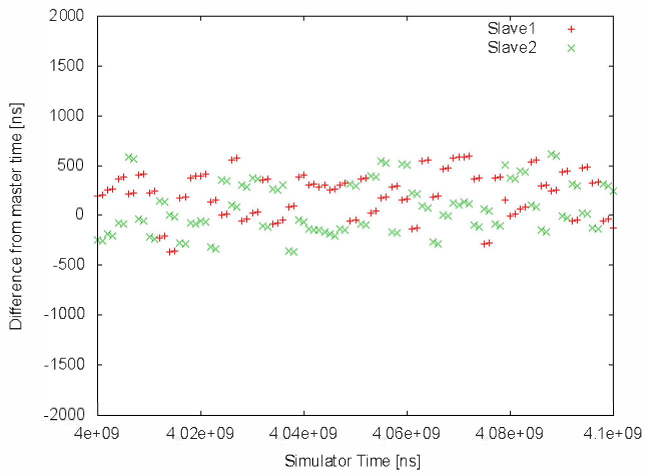

Fig. 6. Simulation result (simple topology)

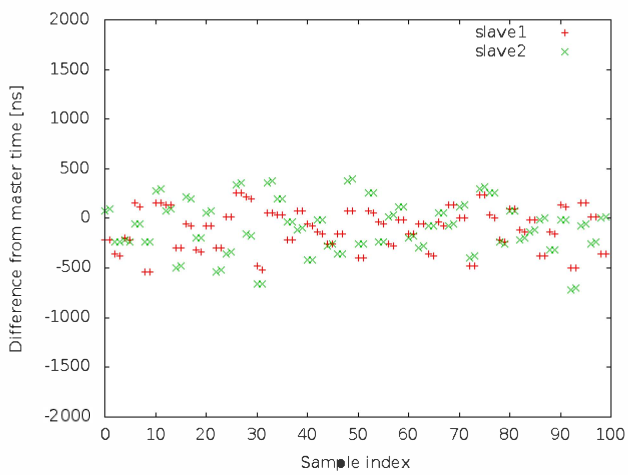

Fig. 7. Experimental result (simple topology)

B. Synchronization simulation

We evaluated a network topology including network switches using the developed simulation platform. Differences between the local time of a master node and system time of slave nodes were compared. All master nodes and slave nodes were modeled as ordinary clock models. If you are performing an experiment using actual equipment, cyclic signals (e.g. PPS signals) of each node have to be monitored at long distance. Simulators can easily evaluate synchronization accuracy using simulator time, which is common time to all nodes.

Fig.8 shows the network topology.

Table I shows the simulation conditions.

To evaluate conflicts among synchronization packets in switches, all packets except for synchronization packets were disabled, and the delay of the switch in the forward path and backward path is the same. Each slave replies to Sync messages randomly; thus, conflicts may occur at random. The distance between a node and a switch was set to 2 km, so the minimum distance and maximum distance between nodes were 4 km and 6 km, respectively. Jitter was set to the local time of each node, and the local time was calculated from the system time using (1).

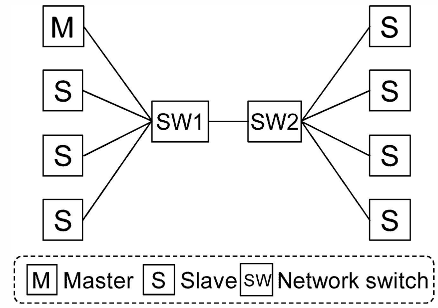

Fig. 8. Network topology for evaluation

TABLE I
SIMULATION CONDITIONS

Item	Condition
Network-related parameters	
Number of nodes	8
Topology	Fig.8
Number of switches	2
Distance between node and switches	2 km (10 μs delay)
Device-related parameters	
Application traffic	0
Processing time of HW/SW	Random below 5 μs
Oscillator accuracy	Random below 50 ppm
Synchronization algorithm	Offset model (P control)
Communication-related parameters	
Communication throughput	1 Gbps
IEEE 1588 transmission mode	1 step
Sync transmission period	Approximately $2\,ms(2^{-9}$ second)
Communication scheduling	No

$$T_L = k \times (T_s + Offset) \tag{1}$$

where k is the proportionality coefficient that indicates jitter (i.e. oscillator accuracy) and $Offset$ is the constant time offset between a master and a slave. The offset model (i.e. P control) was applied as a synchronization algorithm, and then the correct time was offset between master and slave during synchronization.

Fig.9 shows the evaluation result.

As shown in Fig.9, the synchronization error was $\pm 1\,\mu s$ (positive side was $916\,ns$, negative side was $758\,ns$). The variation in synchronization error seemed to be dependent on the probability of packet conflicts in a network switch. The size of the synchronization packet was 90 bytes, so a packet needed a processing time of $720\,ns$ on a 1 Gbps network. The worst delay occurred when the synchronization packets of 4 slaves conflicted simultaneously in SW2. The last packet in the case was delayed by $(720 + 96) \times 3 \simeq 2.45\,\mu s$ ($96ns$ is the inter-frame gap). In this case, an approximately $1.2\,\mu s$ synchronization error (half of $2.45\,\mu s$) may occur. The simulation results indicate a $1\,\mu s$ error as maximum, hence 3 packets may conflict ($916\,ns \times 2 = 1832\,ns$ and $(720 + 96) \times 2 = 1632\,ns < 1832\,ns$).

Besides, the delay between SW1 and SW2 was $10\,\mu s$, which was longer than the maximum processing time of $5\,\mu s$ in a node and the delay caused by conflicts in SW1; then, the

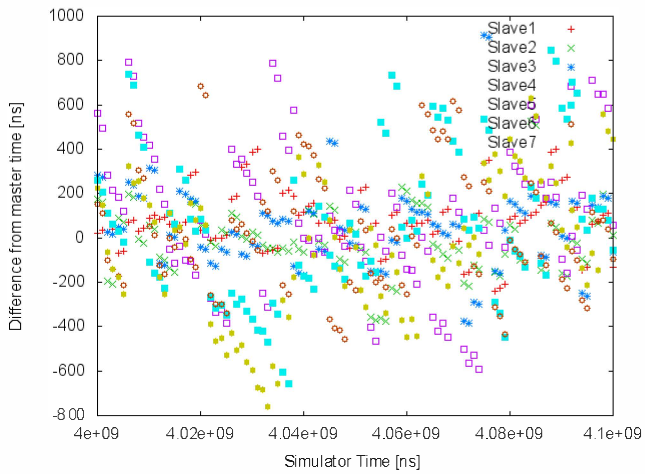

Fig. 9. Result of synchronization evaluation

synchronization packets of slaves connected to SW1 and SW2 were not considered to conflict.

In this way, the synchronization accuracy was simulated, and measures can be taken when designing a system, for example, reducing the number of nodes, reducing the number of network switches, applying a more accurate oscillator, and changing the synchronization algorithms.

For example, Fig.10 shows the case of reducing the number of nodes from 4 to 2 connected to each switch.

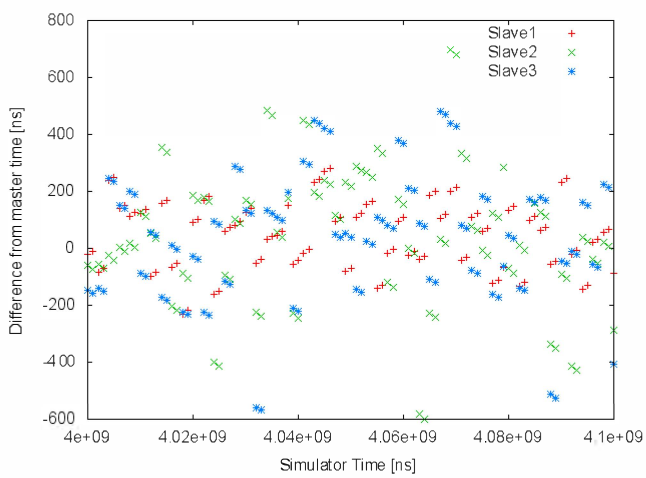

Fig. 10. Result of synchronization evaluation(4 nodes)

Reducing the number of nodes decreased the probability of conflicts between synchronization packets. As a result, the synchronization error is improved to $\pm 800\,ns$ (positive side is $801\,ns$, negative side is $600\,ns$). In Slave 1, variation of synchronization accuracy was not seen because other slaves that conflict do not exist.

Fig.11 shows a result of the case that execution period of synchronization changed to $15.6\,ms$ (2^{-6} second). This means the interval between Sync message transmissions is extended.

As Fig.11, synchronization error becomes larger. This could be caused by the accumulation of synchronization error

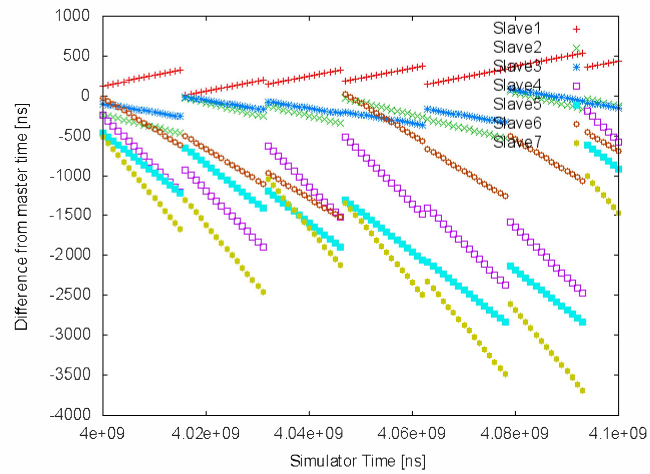

Fig. 11. Result of synchronization evaluation($15.6\,ms$ execution period of synchronization protocol)

because each slave is free running during execution period. In this way, using the developed simulation platform can evaluate synchronization accuracy by changing the parameters in accordance with system requirements.

C. Simulation speed

We measured the simulation speed to evaluate the overhead of our time synchronization enhancements with changing the number of nodes. The execution environment was a PC with Phenom II 2.8 GHz (6 cores) as the CPU and Linux 2.6.27 as the OS. To measure the speed, the *time* command was used.

Fig.12 shows the result.

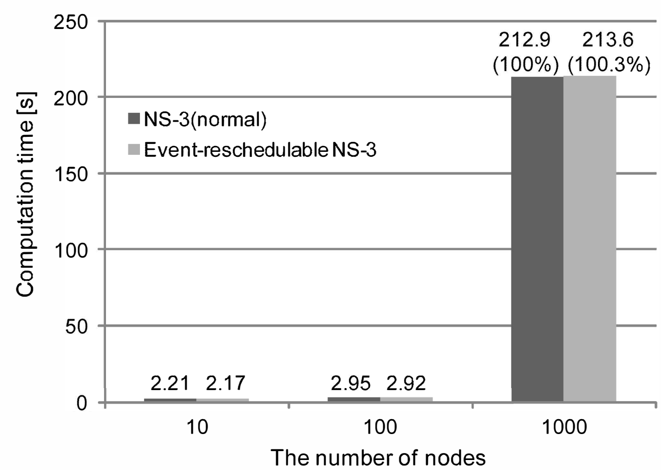

Fig. 12. Result of simulation speed

As Fig.12 shows, the overhead is 0.3% in the case of 1000 nodes. In addition, the computation time of the developed simulator is less than that of a normal NS-3 in the case of 10 and 100 nodes. The overhead was caused by event queue searching ($O(n)$) when removing registered events from the event queue. The result shows the overhead is negligible.

V. Related Work

In other network simulators, there are researches supporting IEEE 1588 simulation.

Lin et al. developed a simulation model of IEEE 1588 ordinary clock to OMNeT++[8]. However, the evaluation was still for a directly connected network, and a multi-hop network simulation was not conducted.

Chen et al. developed an IEEE 1588 model to OPNET[9]. The research targeted multi-hop networks and analyzed asymmetric delay.

Lv et al. developed an IEEE 1588 simulation function to OMNeT++ for wireless multi-hop networks[10].

However, none of the simulations report on event rescheduling coordinated with synchronized time. Whether events can be rescheduled affects the communication performance (bandwidth, delay) and synchronization accuracy. For example, the application bandwidth in TDMA that is described by Lv et al. depends on the accuracy of the time slots. Likewise, the transmission timing of synchronization packets affects synchronization accuracy (synchronization execution period and possibility of conflicts between synchronization packets and other packets). Therefore, to simulate more accurately, event rescheduling coordinated with synchronized time is required.

In addition, we enhanced NS-3, which has a higher simulation speed compared to others, to simulate time synchronization. This means a larger-scaled simulation is possible.

VI. Conclusion

Based on the trend that IEEE 802.3 and IEEE 1588 are promising ways to control systems in multi-hop networks, we developed a simulation platform to assess the synchronization accuracy for various parameters. By using this simulator, the synchronization accuracy and communication performance can be evaluated before designing a system. In future work, the accuracy of the simulation will be improved through actual system evaluations.

References

[1] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pp. c1–269, July 2008.

[2] IEC, *IEC/TR 61850-90-4:2013 Communication networks and systems for power utility automation - Part 90-4: Network engineering guidelines.* International Electrotechnical Commission, 2013.

[3] "The network simulator ns-2," http://www.isi.edu/nsnam/ns/.

[4] T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, "ns-3 project goals," in *In WNS2 f06: Proceeding from the 2006 workshop on ns-2: the IP network simulator*, 2006.

[5] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *In Proceedings of the First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools 2008)*, March 2008.

[6] E. Weingartner, H. vom Lehn, and K. Wehrle, "A Performance Comparison of Recent Network Simulators," in *Communications, 2009. ICC '09. IEEE International Conference on*, June 2009, pp. 1–5.

[7] S. Schriegel and J. Jasperneite, "Investigation of industrial environmental influences on clock sources and their effect on the synchronization accuracy of ieee 1588," in *Precision Clock Synchronization for Measurement, Control and Communication, 2007. ISPCS 2007. IEEE International Symposium on*, Oct 2007, pp. 50–55.

[8] Y. Liu and C. Yang, "OMNeT++ based modeling and simulation of the IEEE 1588 PTP clock," in *Electrical and Control Engineering (ICECE), 2011 International Conference on*, Sept 2011, pp. 4602–4605.

[9] J. Chen, H. Wang, C. Hu, K. Ma, and Z. Cai, "Modeling of IEEE1588 on OPNET and Analysis of Asymmetric Synchronizing Error in Smart Substation," *Energy and Power Engineering*, vol. 5, no. 4B, pp. 540–545, 2013.

[10] B. Lv, Y. Huang, T. Li, X. Dai, M. He, W. Zhang, and Y. Yang, "Simulation and Performance Analysis of the IEEE1588 PTP with Kalman Filtering in Multi-hop Wireless Sensor Networks," *Journal of Networks*, vol. 9, no. 12, pp. 3445–3453, 2014.