sustainability

MDPI

Article

Sustainable Load-Balancing Scheme for Inter-Sensor Convergence Processing of Routing Cooperation Topology

Hyun-Woo Kim [1], Jong Hyuk Park [2,*] and Young-Sik Jeong [1,*]

1 Department of Multimedia Engineering, Dongguk University, Seoul 04620, Korea; hwkim@dongguk.edu
2 Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea
* Correspondence: ysjeong@dongguk.edu (Y.-S.J.); jhpark1@seoultech.ac.kr (J.H.P.); Tel.: +82-2-2260-3374 (Y.-S.J.); +82-2-970-6702 (J.H.P.)

Academic Editor: Marc A. Rosen
Received: 15 February 2016; Accepted: 27 April 2016; Published: 3 May 2016

Abstract: Recent advancements in Information Technology (IT) have sparked the creation of numerous and diverse types of devices and services. Manual data collection measurement methods have been automated through the use of various wireless or wired sensors. Single sensor devices are included in smart devices such as smartphones. Data transmission is critical for big data collected from sensor nodes, such as Mobile Sensor Nodes (MSNs), where sensors move dynamically according to sensor mobility, or Fixed Sensor Nodes (FSNs), where sensor locations are decided by the users. False data transfer processing of big data results in topology lifespan reduction and data transfer delays. Hence, a variety of simulators and diverse load-balancing algorithms have been developed as protocol verification tools for topology lifespan maximization and effective data transfer processing. However, those previously developed simulators have limited functions, such as an event function for a specific sensor or a battery consumption rate test for sensor deployment. Moreover, since the previous load-balancing algorithms consider only the general traffic distribution and the number of connected nodes without considering the current topology condition, the sustainable load-balancing technique that takes into account the battery consumption rate of the dispersed sensor nodes is required. Therefore, this paper proposes the Sustainable Load-balancing Scheme (SLS), which maximizes the overall topology lifespan through effective and sustainable load-balancing of data transfer among the sensors. SLS is capable of maintaining an effective topology as it considers both the battery consumption rate of the sensors and the data transfer delay.

Keywords: sustainable load-balancing; cloud computing; convergence processing; green communication

1. Introduction

Recent advancements in Information Technology (IT) have opened the door to the creation of numerous and diverse devices and services. Miniaturized wireless or wired sensor devices allow not only the automation of manual work but also data collection from regions that humans are not able to access directly. Such sensors possess routing functions for autonomous topology configuration and basic sensing functions for data collection. The number of sensors required for a particular topology ranges from dozens to hundreds and even thousands. Research on the effective transfer and application service of big data that are collected in these topologies is currently being actively conducted in many areas, including government and organizations. In the case of big data, three large topologies are considered [1–10].

The first is Fixed Topology (FT), which is structured in Fixed Sensor Nodes (FSNs) or Static Sensor Nodes (SSNs) where the sensor nodes are in a fixed location according to the user settings. FT is useful for a targeted area that requires fixed observation and cyclical data collection from the same location [6].

The second topology considered is Mobile Topology (MT) which is structured in Mobile Sensor Nodes (MSNs) with autonomous mobility. MT is suitable for a targeted area where data collection is difficult due to low accessibility by humans or other devices. It can also be used when flexible monitoring is required [6].

The third consideration is Hybrid Topology (HT), which is a mixed composition of FSNs and MSNs. HT comes handy for targeted areas that require fixed observation as well as active sensing following human movement or in other exceptional situations.

The big data collected through sensor nodes in FTs, MTs, and HTs are transferred to a sink node, this sink mode transfers big data to middleware or to a direct server. Big data are employed by diverse users, companies, and public organizations depending on their goals and objective. Application services include national safety, telecommunication systems, energy savings, process control, traffic control, healthcare, and distributed robots.

Data transfer processing is critical because drastic battery consumption of a single sensor node due to false data processing of big data in FTs, MTs, and HTs can have a negative influence in the whole topology. This has led to multiple studies on the development of various simulators and topology reconfigurations, which have created tools for determining the appropriate sensor nodes, sensor deployment, and load-balancing techniques for an observation area [4,6,7,11].

Despite the research and development surrounding such simulators, topology configuration for effective collection and transfer of big data has not been plausible due to its limited function and the topology reconfiguration and control of the load-balancing that do not reflect the constructed topology condition.

Therefore, in this paper, we propose the Sustainable Load-balancing Scheme (SLS) that considers the battery consumption rate of each sensor to achieve effective data transfer. SLS can maximize the topology lifespan by considering the battery remains and consumption rate of sensors in FTs, MTs, and HTs, and the routing condition of big data.

This paper is organized as follows.

Section 2 reviews the literature on simulators that were previously developed as a topology design and test tool for big data of sensors. This section also reviews the literature on load-balancing techniques for topology reconfiguration and control. Section 3 describes the proposed load-balancing, which considers the composition, visualization, and battery consumption rate of the SLS sensor nodes. Sections 4 and 5 describe the design and implementation of SLS, respectively. Section 6 examines performance improvements by comparing previous simulators and measures the topology lifetime in the SLS application. Finally, Section 7 contains the overall summary, conclusion, and future research tasks.

2. Related Works

2.1. Existing Topology Design and Verification Tools

Existing topology design and verification tools are explained in Table 1.

Table 1. Existing topology design and verification tools.

Tools	Description
ATEMU [11]	ATEMU, which was the first command-based simulator based on C, does not provide a Graphical User Interface (GUI). Moreover, its performance rate is extremely slow due to the implementation of sequential sensor nodes despite text-based simulation. It guarantees cycle accuracy; however, only the parameters provided on the different systems can be determined.
AVRORA [11]	AVRORA is Java-based and it is capable of the simultaneous simulation of multiple sensor nodes. Each sensor node is implemented by a single thread. Prompt understanding is difficult as a GUI is not provided.
GloMoSim [11]	GloMoSim uses Parsec language for parallel simulation of the topology of large-scale sensor nodes. However, as the number of sensor nodes that are to be built in the actual environment provide simple deployment, routing prediction of big sensing data and load-balancing simulation are not possible.
NS2 [11]	NS2 is a discrete event simulator that has a module-type approach. Although NS2 has a number of sensor setting functions, the application program model is not sufficient when interaction between the application program and the network is required. Although NS2 has a Network Animator for GUI support, simultaneity is excluded as the event command is called from the file where it was previously stored.
NS3 [11]	NS3 provides better performance in the aspect of memory management compared to NS2. However, it is a new simulation tool rather than an extended version of NS2 and it does not support all NS2 models. Consequently, simulation is not possible with incompatible models. Moreover, an analysis function for load-balancing is not supported.
QualNet [11]	QualNet is the next version of GloMoSim. It is capable of simulating module scenarios and object models developed by different designers. Although it supports the function of a sensor network, analysis for load-balancing is still required as it only provides limited RF analysis.
SENSE [11]	SENSE is a simulator based on C++ with a complex FT, MT, and HT configuration. Although we can use G-Sense for substituting the autonomous visualization tool, it is not appropriate for real-time processing of MSN movement. Therefore, with this simulator, it is difficult for users to detect topology configuration problems for big sensing data.
SWANS [11]	SWANS is a Java-based simulator capable of user model definition. Moreover, it can present network communication flow on topology on an interface. However, condition setting for load-balancing in FTs, MTs, and HTs is not possible and it also has the limitation of showing only event-based communication flow.
TOSSIM [11]	TOSSIM is a simulator developed at Berkeley University in the United States that simulates TinyOS. Although it is capable of actual movement inference and hardware analysis, simulation of OS and the functions of other sensor nodes is not possible due to TinyOS dependency. TOSSIM has a Java-based TinyViz for a GUI, but it is not sufficient for the movement of dynamic sensor nodes.

2.2. Topology Reconfiguration and Control and Load-Balancing Mechanism

A Connected Dominating Set (CDS) [12] guarantees connectivity to MSNs and FSNs. Moreover, it accelerates the routing process by simplifying the connected links [12].

The Construction Algorithm for Reliable CDS (CARCODS) [13] is a method of improving the performance of the composed ad-hoc topology. It suggests a CDS composition method using the neighboring composition report message broadcasting delay time that considers the residual battery, mobility, and the number of neighboring MSNs and FSNs [13].

Partial Reconstruction of CDS (PRCDS) [14] proposes a CDS partial reconfiguration algorithm that can effectively respond to the problem of critical node occurrence in a CDS-based routing protocol. In the CDS node setting for load-balancing, connectivity is searched under the condition where the topology reconfiguration due to the critical node is expanded to a 2-hop node. Moreover, it shows efficiency according to reconfiguration time in the case of within-range. However, reconfiguration frequently occurs due to critical node creation, which negatively affects the overall topology. Therefore, in this paper, we provide an improved topology by considering not only the critical node, but also the overall topology and each sensor's battery consumption rate.

The Simple Distributed Approximation Algorithm (SDAA) [15] removes unnecessary CDS composition nodes as the number comes close to the minimum number of gateway nodes that construct a CDS. Even though it was originally proposed to improve the overall topology lifespan by decreasing the energy consumption in MSNs and FSNs, it is not ideal for frequent data collection environments, such as those encountered with big sensing data. This paper provides for a stable overall topology lifespan by suggesting routing based on the battery consumption rate and considering the overall connectivity instead of simply making the number of connected nodes consistent.

3. SLS Scheme

An FT, MT, and HT with the SLS application function are shown in Figure 1. This operation not only considers the number of sensor nodes connected within the constructed topology for big sensing data transfer, but is also capable of bypassing it by considering the battery consumption rate, which eventually contributes to the maximization of the overall topology lifespan.

The sensor nodes with SLS implementation are largely composed of sensor id, observation, base information, and current information; the XML scheme is presented in Figure 2. This type of XML scheme is used for load-balancing in FTs, MTs, and HTs, and is aimed at the maximization of the overall topology lifespan for big data of sensors.

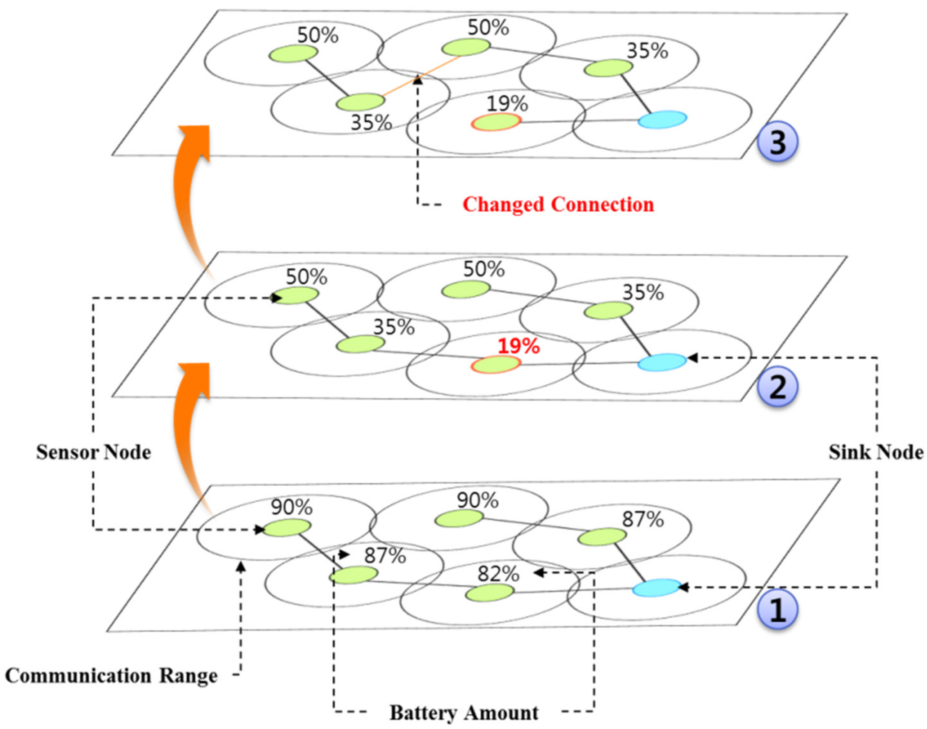

Figure 1. Load-balancing of SLS.

```
- <sensor>
    <id />
    <!--  sensor id   -->
  - <observation>
      <time />
      <!--  sensing time   -->
    - <data>
        <unit />
        <!--  sensor data type   -->
        <value />
        <!--  sensing value   -->
      </data>
    </observation>
  - <base_information>
      <location />
      <!--  location of sensor node(MSN, FSN)   -->
      <total_battery />
      <!--  total battery size of sensor node(MSN, FSN)   -->
      <active_power />
      <active_time />
      <sleep_power />
      <sleep_time />
      <wakeup_time />
    </base_information>
  - <current_information>
      <start_time />
      <!--  start activation time of sensor node(MSN, FSN)   -->
      <current_battery />
      <!--  current battery amount of sensor node(MSN, FSN)   -->
      <battery_consumption_rate />
      <!--  hourly consumption rate of sensor node(MSN, FSN)   -->
      <connected_node_count />
      <!--  number of connected other node   -->
    </current_information>
  </sensor>
```

Figure 2. XML scheme of sensor nodes for SLS.

Each SLS-applied sensor operates in two ways according to the threshold setting of the battery consumption rate and battery remains. The first case, where the threshold is not determined, operates as follows:

① The battery consumption rate per unit time is computed in the initially composed topology. The battery consumption rate is calculated by using the battery remains after the implementation to active and sleep modes with the time that the sensor first operated as a standard, which can be expressed as in Equation (1).

$$Battery_{consumption} = \frac{Battery_{total} - Battery_{current}}{CurrentTime - StartTime} \tag{1}$$

② The battery consumption rate is compared to that of other connectable neighboring sensor nodes. The basic method of comparison is to sending its own battery consumption rate during communication for connectivity maintenance. Here, the comparison is conducted when transferring sensing data or routing sensing data from other sensor nodes.

③ Referring to the battery consumption rate comparison results in ②, sensing data are transferred to the sensor nodes that have lower battery consumption rates.

④ The received sensor nodes iteratively perform ①, ②, and ③ until the transfer to the sink node is complete.

If the threshold of the battery consumption rate or the battery remains is defined, the operation occurs as follows:

① The battery consumption rate per unit time is computed in the initially composed topology. The battery consumption rate is calculated by using the battery remains after the implementation to active and sleep modes with the time that the sensor first operated as a standard, which can be expressed as in Equation (1).

② If $Battery_{consumption}$ is equal to or larger than the threshold value, proceed to ③. Otherwise, sensing data are transferred to the connected sensor nodes. If it is not larger than the threshold value, ① and ② are iteratively performed until the received sensor node transfers to the sink node, routing the sensing data.

③ The battery consumption rate is compared to that of other connectable neighboring sensor nodes. The basic method of comparison is to send its own battery consumption rate during communication for connectivity maintenance. Here, the comparison is conducted when transferring sensing data or routing sensing data from other sensor nodes.

④ Referring to the battery consumption rate comparison results in ②, sensing data are transferred to the sensor nodes that have lower battery consumption rates.

4. Design of the SLS

The Sustainable Load-balancing Scheme (SLS) can be divided by function into User Interface, Node Manager, Interaction Broker, Map Manager, Load Balancer, Map Controller, Coordinate Converter, and Viewer. Figure 3 presents a structure map of the overall function.

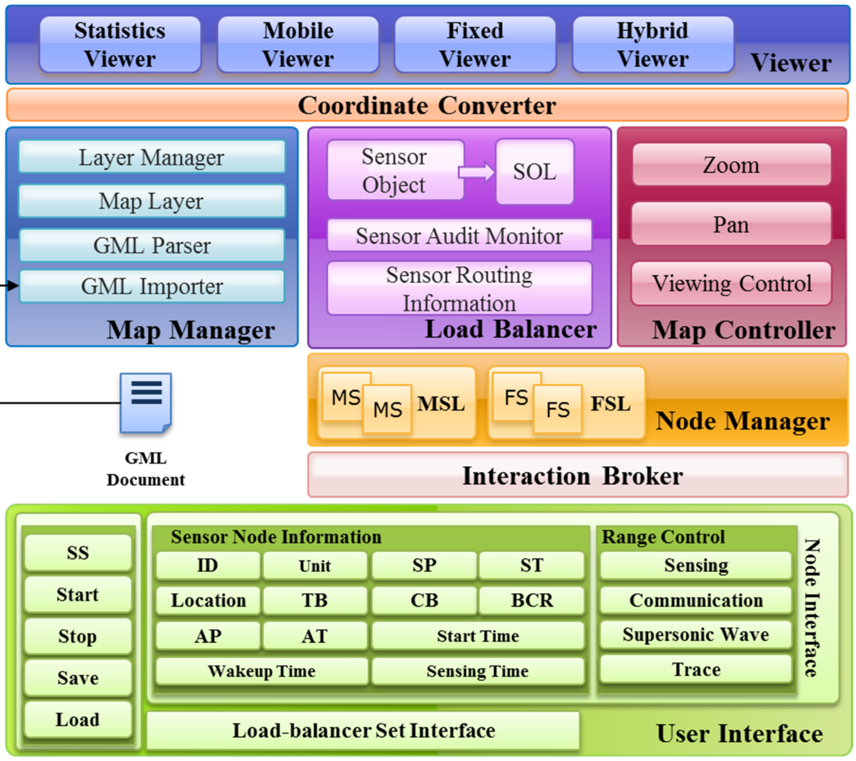

Figure 3. SLS architecture.

Specifically, the User Interface, which is composed of the Node Interface, Load-balancer Set Interface, and Set-Simulation (SS), sets new sensor deployment for the current topology and simulates load-balancing; Start and Stop, which begin and end the SS operation, respectively; Save, which saves the simulation setting condition; and Load, which brings back the previously saved simulation condition. The Sensor Node Information of the Node Interface receives user setting inputs regarding Unit and Location that represent sensor ID and sensor data type, Total Battery (TB), Active Power (AP), Active Time (AT), Sleep Power (SP), Sleep Time (ST), Current Battery (CB), Battery Consumption Rate (BCR), Start Time, Sensing Time, and Wakeup Time. Moreover, for basic sensor control, Sensing Range and Communication Range, Supersonic Wave Range, and Trace Range are determined.

Node Manager manages either by Mobile Sensor List (MSL) or Fixed Sensor List (FSL) according to the mobility of the deployed sensor nodes. Each sensor has routing data updated through the Load Balancer.

Interaction Broker plays the role of a broker that transfers the operation mode entered from the User Interface and the setting change of the sensor to the Map Controller, Node Manager, and Load Balancer.

Map Manager applies an actual topography and GML document that can be mapped to the SLS and managed. Specifically, it consists of a GML Importer that adds the GML document that was selected by the user to the SLS, the GML Parser that analyzes the added GML document, the Map Layer that forms the Map object by determining the existence of obstacles according to the object of the analyzed GML topography data and transfers them to the Layer Manager, and the Layer Manage that manages the topography information received from Map Layer.

Load Balancer manages the Sensor Objects in the Sensor Object List (SOL). The sensor objects in the SOL check the survival of the sensor and the arrival of the threshold that was determined by the user through the Sensor Audit Monitor. Once the threshold is reached, the Sensor Routing Information is changed to improve the overall topology lifespan. The changed information is transferred to the Node Manager.

Map Controller is in charge of zoom-in, zoom-out, range expansion, selective movement, *etc.* of the map that is visualized in the Viewer. These functional performance results are transferred to the Viewer through the Coordinate Converter to be visually shown to the user.

Coordinate Converter processes and transfers the operation condition data through the topography and the log information of sensor nodes such that they can be expressed in the Viewer.

Viewer visualizes the processed data transferred from the Coordinate Converter to the user based on their types. The Viewer is composed of the Mobile Viewer that visualizes dynamically moving sensor nodes, the Fixed Viewer that visualizes fixed sensor nodes, the Hybrid Viewer that shows both the Mobile Viewer and Fixed Viewer sensor nodes, and the Stats Viewer that visually expresses diverse analysis results regarding MSNs or FSNs. Through this Viewer, the user can check the load-balancing condition and deduct problems for the lifespan maximization of the overall topology according to the sensor deployment condition and actively respond to them.

5. Implementation of SLS

Figure 4 shows the operation screen for the SLS. Figure 4① illustrates the default settings for the MSN and the FSN and the Control View where the user selects the Sensing Range, Communication Range, Supersonic Wave Range, and Trace Range. Figure 4② represents the Viewer, which is in charge of user visualization according to the View Mode selected in ③. In Figure 4③, the Mobile Viewer, Fixed Viewer, Hybrid Viewer, and Stats Viewer are selected by the user. Figure 4④ shows the ability to set the threshold of the sensor nodes that operate in the topology for load-balancing. Figure 4⑤ shows the first deployment of the Mobile Viewer, while ⑥ and ⑦ show the change over time. Figure 4⑧ shows the first deployment of the Fixed Viewer, while Figure 4⑨,⑩ show the change over time. The evenness of the battery consumption of the overall topology can be checked by referring to these changes. Figure 4⑪ displays the Hybrid Viewer that shows the Mobile Viewer and Fixed Viewer either in single or in parallel view. Because of this Hybrid Viewer, the load-balancing condition among the sensors within the topology can be confirmed, as it shows changes, such as ⑫ and ⑬, according to the time change.

As shown Figure 5, SLS can recognize the energy consumption ratio of executing MSN and FSN through Statistic Module. ① in Figure 5 shows the status of the sensor node in the table including FSN status, MSN status and both. As shown in Figure 5, SLS exhibits the expected life time of the topology of MSN, FSN and Hybrid in the same configured environment as through ②. Further, ③ in Figure 5 explains the location information of the selected sensor node within MSN and FSN by the user. Finally, ④ visualizes the selected sensor nodes status that synchronized to ① in Figure 5. Therefore, we can find the location of sensor nodes that have to be replaced on the topology through execution of SLS.

Figure 4. Operation for each view mode in SLS.

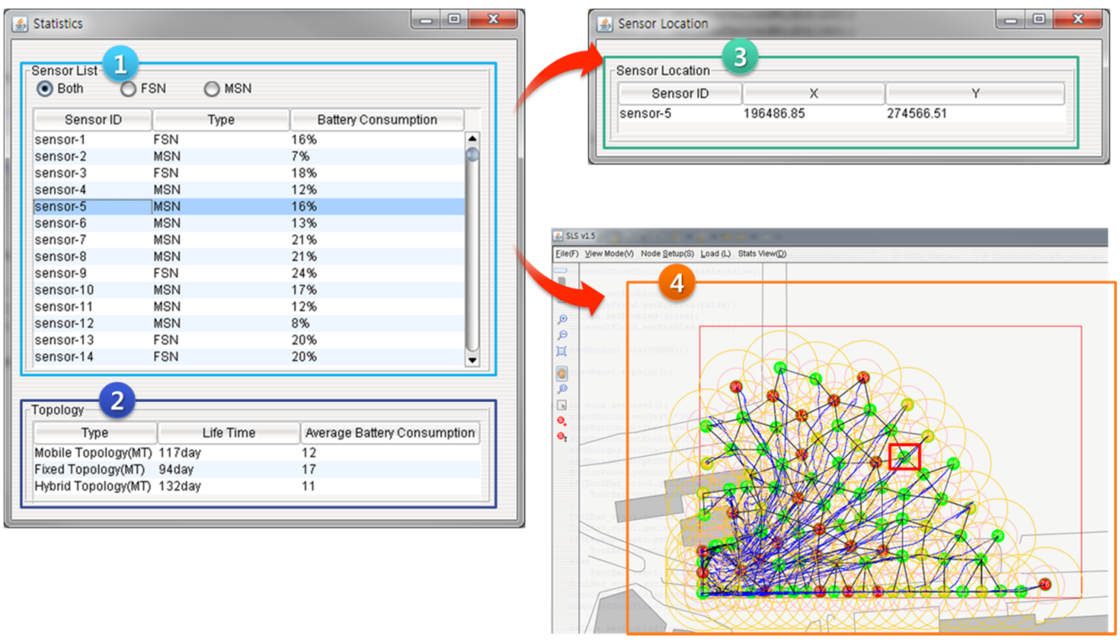

Figure 5. Life time prediction of sensor node and topology.

6. Performance Evaluation

For performance evaluation of the SLS, the coverage rate change was investigated according to the SLS application under identical environmental conditions and the same number of sensor nodes. Figure 6 shows the observation results for the coverage rate over time, where the identical number of sensor nodes is deployed under four different environments. In the case of 100 sensor nodes, the topology where the SLS was displayed showed relatively higher coverage from the tenth day. Similarly, the coverage showed a large differences between SLS and non-SLS over time in the case of 200 sensor nodes, 300 sensor nodes, and 400 sensor nodes, even though no large difference was observed at the beginning. Despite the use of identical protocols, the topology coverage rate for big data turned out to be higher with the application of SLS.

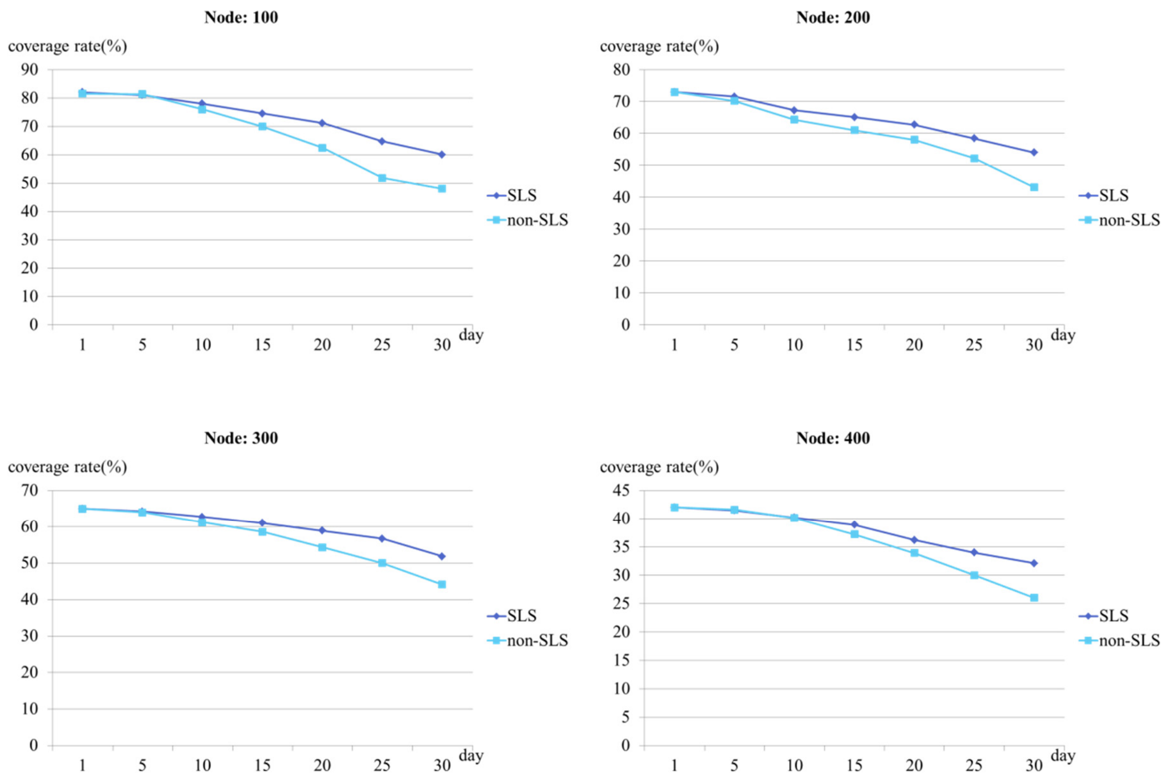

Figure 6. Comparison of coverage rate changes according to the SLS application for each topology.

7. Conclusions

The current topology design and verification tools for big sensing data collection have inherent problems. They have a slow processing speed and limited functions when simulating a number of FSNs and MSNs. Moreover, big sensing data generated in FTs, MTs, and HTs are composed of dozens to hundreds and even thousands of FSNs and MSNs; hence, drastic battery consumption can occur when they are routed with identical sensor nodes. This eventually affects the balance of the overall topology. Consequently, collecting required data can become implausible, causing secondary problems. Therefore, in this paper, we proposed the Sustainable Load-balancing Scheme (SLS), which bypasses sensor nodes with frequent routing, considering not only the number of sensor nodes that are connected to each FSN and MSN, but also the battery consumption rate. SLS provided effective big sensing data collection and transfer as the total topology lifespan is maximized through even battery consumption among the FT, MT, and HT.

Future studies will involve data transfer and storage centered at middleware, in addition to load-balancing for effective processing of big sensing data. Automatized deployment of numerous

MSNs and FSNs will also be examined for effective big sensing data collection and transfer in FTs, MTs, and HTs, taking into account the topography conditions.

Acknowledgments: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2014R1A1A2053564).

Author Contributions: All the authors contributed equally to this work. All authors read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gupta, G.P.; Misra, M.; Garg, K. An Energy Efficient Distributed Approach-Based Agent Migration Scheme for Data Aggregation in Wireless Sensor Networks. *J. Inf. Process. Syst.* **2015**, *11*, 148–164.
2. Jeong, Y.-S.; Kim, H.-W.; Jang, H.J. Adaptive resource management scheme for monitoring of CPS. *J. Supercomput.* **2013**, *66*, 57–69. [CrossRef]
3. He, B.; Li, Y. Big Data Reduction and Optimization in Sensor Monitoring Network. *J. Appl. Math.* **2014**, *2014*. [CrossRef]
4. Kim, H.-W.; Jeong, Y.-S. Integrated Validation System for the Simulation of Diverse Sensor in WSNs. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*. [CrossRef]
5. Pughat, A.; Sharma, V. A review on stochastic approach for dynamic power management in wireless sensor networks. *Hum.-Centr. Comput. Inf. Sci.* **2015**, *5*. [CrossRef]
6. Jeong, Y.-S.; Kim, H.-W.; Park, J.H. Visual Scheme for the Detection of Mobile Attack on WSN Simulator. *Int. J. Distrib. Sens. Netw.* **2013**, *2013*. [CrossRef]
7. Jeong, Y.-S.; Kim, H.-W.; Park, J.H. Visual Scheme Monitoring of Sensors for Fault Tolerance on Wireless Body Area Networks with Cloud Service Infrastructure. *Int. J. Distrib. Sens. Netw.* **2014**, *2014*. [CrossRef]
8. Ahn, H.; Jung, B.K.; Park, J.R. Effect of reagents on optical properties of asbestos and remote spectral sensing. *J. Converg.* **2014**, *5*, 15–18.
9. Jeong, Y.-S.; Song, E.-H.; Chae, G.-B.; Hong, M.; Park, D.-S. Large-Scale Middleware for Ubiquitous Sensor Networks. *IEEE Intell. Syst.* **2010**, *25*, 48–59. [CrossRef]
10. Ren, Y.; Wang, B.; Zhang, S.; Zhang, H. A Distributed Energy-Efficient Topology Control Routing for Mobile Wireless Sensor Networks. In Proceedings of the 6th International IFIP-TC6 Networking Conference, Atlanta, GA, USA, 14–18 May 2007; pp. 132–142.
11. Kim, H.-W.; Park, J.-H.; Song, E.-H.; Jeong, Y.-S. Parallel Processing Simulator for Separate Sensor of WSN Simulator with GPU. In Proceedings of the 29th International Conference on Advanced Information Networking and Application, IEEE, Gwangju, Korea, 24–27 March 2015; pp. 255–262.
12. Blum, J.; Ding, M.; Thaeler, A.; Cheng, X. Connected Dominating Set in Sensor Networks and MANETs. In *Handbook of Combinatorial Optimization*; Springer: New York, NY, USA, 2005; pp. 329–369.
13. Cho, H.-S.; Yoo, S.-J. Power, mobility and wireless channel condition aware connected dominating set construction algorithm in the wireless ad-hoc network. *J. Korean Inst. Commun. Inf. Sci.* **2005**, *30*, 274–286.
14. Hong, Y.-S.; Lim, H.-S. A Load-balanced Topology Maintenance with Partial Topology Reconstruction. *J. Korean Inst. Commun. Inf. Sci.* **2010**, *35*, 1188–1197.
15. Wu, J.; Dai, F.; Gao, M.; Stojmenovic, I. On Calculating Power-Aware Connected Domination Sets for Efficient Routing in Ad Hoc Wireless Networks. *J. Commun. Netw.* **2002**, *4*, 59–70. [CrossRef]