University Library

MINERVA ACCESS

A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
SRIDHARA RAO, A;Gubbi, J;Ngo, T;Mendis, P;Palaniswami, M

Title:
Internet of Things for Structural Health Monitoring

Date:
2016-05

Citation:
SRIDHARA RAO, A., Gubbi, J., Ngo, T., Mendis, P. & Palaniswami, M. (2016). Internet of Things for Structural Health Monitoring. Epaarachchi, A (Ed.). Chanaka Kahandawa, G (Ed.). Structural Health Monitoring Technologies and Next-Generation Smart Composite Structures, (1), pp.89-120. CRC Press.

Persistent Link:
https://hdl.handle.net/11343/302080

Internet of Things for Structural Health Monitoring

Aravinda S. Rao, Jayavardhana Gubbi, Tuan Ngo, Priyan Mendis and Marimuthu Palaniswami

Contents

1 Smart Cities

Australia is an urbanized society. Happily, Australian cities are regularly rated as world's most liveable. Like most developed countries, major challenges to urban landscape are emerging: higher (and aging) populations, environmental and resource stresses, infrastructure capacities, security issues, etc [36]. These challenges need to be addressed now to prepare our cities for tomorrow: future cities have to be 'smart'-digitally smart - if they are to be sustainable. Present urban universe is neatly partitioned into autonomous spheres: (a) real-where people, systems, services and environments lie and (b) virtual-where information and communication lie. In smart cities this divide does not exist; the real and the virtual are seamlessly integrated. Technologically, this integration is predicated on the creation of some very new disruptive paradigms on the data-knowledge-action axis of real-virtual symbiosis. The Internet revolution led to the interconnection between people at an unprecedented scale and pace. The next revolution will be the interconnection between objects to create a smart environment. Currently, there are 9 billion interconnected devices and it is expected to reach 24 billion devices by 2020 [59]. According to the Groupe Speciale Mobile Association (GSMA), this amounts to $1.3 trillion revenue opportunities for mobile network operators alone spanning vertical segments such as transport, infrastructure, health and entertainment [20]. The Digital Europe strategy of the European Union [12] points to some of these new paradigms: the Internet of Things [50], Big Data [4] and Cloud computing [6]. The first integrates the information highways and the physical world through ubiquitous sensing and actuation, the second is to extract actionable knowledge from constantly generated massive data from diverse sources (human and machine), and the third is to manage the required heterogeneity of computational platforms and goals in pervasive environments.

The emerging technology is applicable in almost all areas of city management. Some of the key application areas include:

Environmetal Monitoring The large-scale high-resolution environmental data will enable continuous monitoring of the city environment.

Video Monitoring Video sensor networks which integrate image processing, computer vision and networking frameworks will help develop a new challenging scientific research area at the intersection of video, infrared, microphone and network technologies. Surveillance, the most widely used camera network application, helps track a person, identify suspicious activities, detect left luggage and monitor unauthorised access.

Structural Health Monitoring As cities age, it is important that the critical infrastructure are regularly monitored and appropriate measures taken in order to extend the longevity. This includes measuring strains in structural components using advanced sensor technologies.

Traffic Congestion and Impact Monitoring Urban traffic is the main contributor to traffic noise pollution and a major contributor to urban air quality degradation and greenhouse gas emissions. Improved dynamic traffic information will improve freight movement, allow better planning and improve scheduling. The applications areas within traffic include dynamic traffic management and control strategies involving interactions amongst vehicles; vehicles and infrastructure; and different modes of transport.

In this chapter, we narrow our focus on structural health monitoring, particularly on critical infrastructure monitoring. Critical infrastructures are those that are essential for everyday living. Examples include physical structures such as long bridges connecting major locations for people to commute, tall buildings where region's workforce is mostly located; communication, electricity and water supply networks; information technologies; food and supply chains etc. Here, the focus is on civil infrastructure that mainly includes bridges and buildings. Integrity of the civil infrastructures are very important to maintain safety of everyone in and around such structures. In order to ensure that the structural integrity is maintained, health monitoring of these structures are carried regularly based on approximate estimates.

2 Structural Health Monitoring

According to Sohn *et al.* [46], “The process of implementing a damage identification strategy for aerospace, civil and mechanical engineering infrastructure is referred to as *Structural Health Monitoring (SHM)*.” Structural health monitoring implies monitoring of the state of the structures through sensor networks in an on-line mode and are pertinent to aircraft and buildings[55]. Structural health monitoring can be further divided into two categories [7]: (a) global health monitoring and (b) local health monitoring. Global health monitoring is the process where the damage to the structural integrity is determined by identifying damage to the structure. Whereas, local health monitoring deals with identifying the specific locations in the structures where the damage has occurred.

Damage to a structure can be defined as altering the true state of the stricture that may hamper the performance of the structure. The structural damage can be due to the changes in material composition, physical constraints such as load, and also due to environmental factors and ageing. Detection and evaluation of damages to structures are part of most of the engineering systems. The five key areas where damage detection is primarily focused are: (1) Structural Health Monitoring (SHM-[55]), (2) Condition Monitoring (CM-[5]), (3) Non-destructive Evaluation (NDE- [44]), (4) Statistical Process Control (SPC-[35]), and (5) Damage Prognosis (DP-[15]).

Structural Health Monitoring (SHM) is performed by using acoustic sensors, ultrasonic sensors, strain gauges, optical fibres etc. Condition Monitoring (CM) is assessment of the damage to the rotating machinery, where vibration, temperature and accelerations provide insight about the damages to the structure. Non-destructive Evaluation (NDE) is used to assess the local characteristics of the structure once the damage has been localized. Test such as ultrasonic, shear stress, magnetic particle inspection are some of the NDE methods. Statistical Process Control (SPC) evaluates the quality of the process where statistical methods are used. The monitoring process provides stability and minimum quality standards to the structures depending on the materials and the process used. Once the damage to the structure has been detected, Damage Prognosis (DP) is typically used to assess and predict the end time of the structure. SHM and CM are typically performed on-line, whereas NDE, SPC and DP are usually offline. Figure 1 shows the schematics of SHM using traditional sensing techniques.

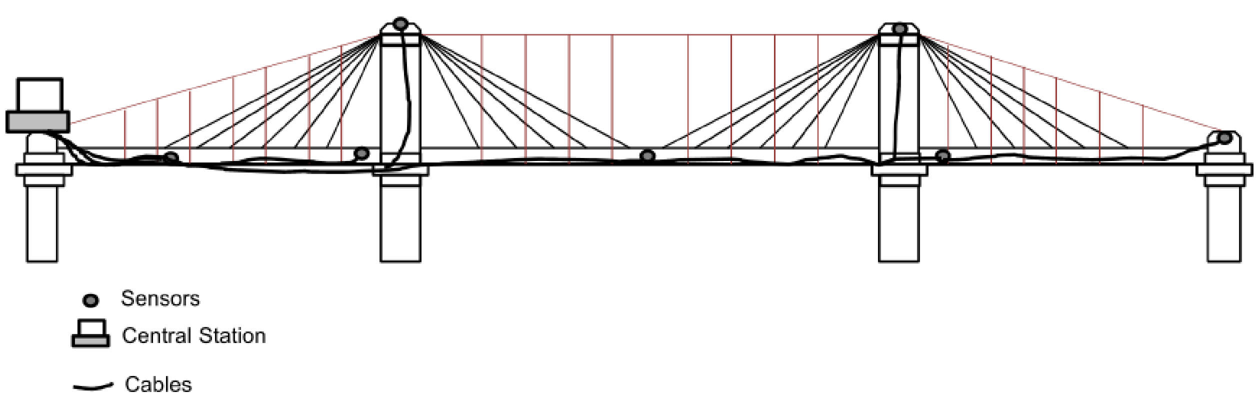


Figure 1: SHM using traditional sensing modalities. The central station collects the data from all the sensors using long cables.

3 Sensors and Sensor Networks

The development of advanced sensors, miniaturizations in integrated circuits, advancements in design engineering, reduction of power consumption and cost acted as catalysts to the development of sensors during the late 1990s. Further impetus was provided the much-needed communication network, the Internet. The integration of these two technologies spawned a new widespread technology commonly known as sensor networks. The ability of the sensor networks to send data over the Internet further enhanced the scope and usage of the sensor networks. The capability of wireless communication by the sensor nodes along with computation power and sensing proved to be a benign technology. Sensor networks with wireless communications are referred to as Wireless Sensor Networks (WSNs).“Smart Dust“ [39], [25], the pioneering node with 5 square millimeters [58] in size included Micro Electro-Mechanical Systems (MEMS) sensor, wireless communication and processing.

3.1 Traditional Sensors and Usage

Sensor networks can use wide range of sensors. Typically the sensors are chosen based on the application and end-user requirements. Power consumption and cost are also other important factors that affect the selecting of the sensors. Some of the most commonly used sensors are [11], [2]: temperature, pressure, humidity, soil condition, movement of vehicles, noise levels, light levels, detection of specific objects, mechanical stress, characteristics of an object such as speed, direction and size can also be monitored; event detection, continuous monitoring and remotely controlling actuators make the sensor networks attractive solutions toward added smartness to the human interactions. The application areas of sensors and sensor networks are numerous. These can be broadly categorised as [2]: military applications, environmental applications, healthcare, home automation, and other industrial applications. The focus here will be on the use of WSN on civil engineering and infrastructure applications, in particular, on SHM.

3.2 Limitations of Traditional Monitoring

SHM allows to assess and estimate the damage to the structures. This maybe due to structural ageing, environmental effects or the damage due to the structural material. The key factors required in assessing the damage are [27]: (1) temporal variations in structural integrity, and (2) degree of damage to structural integrity. Visual inspection, X-ray, and accelerators are some of the commonly used traditional methods. Specifically, accelerators and strain gauges are connected to a computer with appropriate software to collect the sensor data. Many times, the structures are too long and unreachable. Additionally, tall buildings puts lives of many people at risk in collecting and monitoring the sensor data. There is also unease of carrying the computer and connecting the long wires. It is also important to note that the length of senor cables connecting to the computer affects the sensor readings and sometimes become corrupted. Some of the major drawbacks are: (1) high cost of measurement due to long cables and sophisticated protection equipments, (2) high cost of maintenance as both experts and non-experts from sensors to structures are required, (3) only localized sensor data is obtained (i.e. the long-term behaviour of the structural changes are not known, and (4) as a consequence, the estimation of lifetime of the structure becomes rather difficult with sparse data. There is a need for new sensors, sensing methodologies and also methods in constructing the structures.

3.3 Carbon Nanotubes for Structures

Carbon nanotubes have been in popularity since 1991, after the synthesis of needle-like structures that produce fullerenes [23], although initial developments were made in 1976 by growing carbon fibres [38] . The application of carbon nanotubes are widespread due to their excellent properties. The applications of nanotubes include [24]: electronics, energy applications, mechanical applications, sensors, fields emission and lighting applications, biological applications. Nanotubes have Young’s modulus higher than diamonds [52], [42]. Theoretical [57] and experimental [13] evidence show that nanotubes can undergo huge deformations and yet recover elastically. With a 63 GPa tensile strength, they are two magnitude order greater than steel [61]. Undoubtedly, nanotubes can be used as a materials for formation of civil structures. For instance, Li *et al.* [31] found that 20 mm long ropes of single-walled carbon nanotube (SWNT) ropes had the tensile strength of 3.6 ± 0.4 GPa, which is comparable to carbon fibres. They concluded that for high performance composite materials, SWNT make a ideal reinforcement candidate. With their low-density (1.3 g/cm ^{3}), they have the highest specific strength $> 46{,}000$ kN-m/kg [8]. In [8], provides the study of tensile strength of kilometer long bridges made of carbon nanotubes. Carbon nanotubes can be used to new sensors as well as building strong structures with their excellent properties. Thus, they form the core of the future structures.

3.4 Continuous Monitoring Using Wireless Sensor Networks

However, as mentioned before, there are several drawbacks in conventional way of performing SHM. WSNs offer a wealth of opportunities to conduct the monitoring of structural health. They have the ability to continuously monitor the over long periods of time without human intervention. WSNs provide a important platform for long-term and continuous monitoring. The temporal variations of structural integrity and the damage severity can be measured in high resolutions (i.e data can be collected in seconds). WSNs also eliminate the need for the personnel to be present during sensing, effectively reducing the life risks of the workers. WSNs are low-cost and low-power devices that would reduce the installation and maintenance costs of the SHM. They can form long networks by wirelessly communicating with neighboring/designated nodes. This reduces the need for long cables running over the bridges and tall buildings. By continuous monitoring of the structures, the integrity of the structures can be accurately estimated. Figure 2 shows the schematics of SHM using WSN.

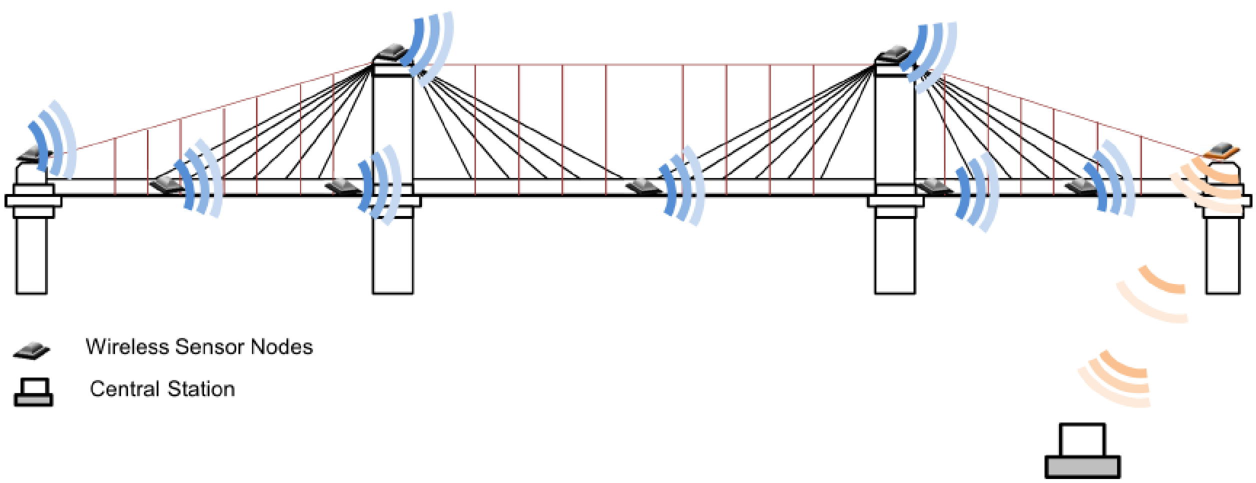


Figure 2: SHM using Wireless Sensor Networks. The central station collects the data from all the nodes through gateway router node.

However, WSNs face several challenges in continuous monitoring. Energy available to the sensor nodes are limited. They must operate at a very low power. Computation must also be reduced in order to reduce power. However, some of the computations such as processing sensor data and wireless transmissions are a must. The distance of communication also poses a severe challenge. If the distance is increased, then the power of transmitter must be increased or the packets will fail. Any distance beyond communication range will never be reached. Placing sensor nodes close to each other would overcome the communication problem to some extent, but the cost increases since the number of nodes in a given area increases with reduced distance. Furthermore, sometimes the packets get corrupted and the data will becomes void. Also, there are times when packet loss can be noticed due to environmental factors. Since the sensor nodes are made of low-cost hardware, the clocks on the devices have low-end oscillators that cause the clocks to drift. This introduces the problem of time synchronization. This is regarded as one of the important problems in WSNs. Because of time error, the analysis of the sensor data may deceive the assessment to falsely predict the integrity of the structures. Some of the core challenges in WSNs are : power management, low-power design, programming limitations, computational power imitations, real-time communication, low-bandwidth, network failure, security and privacy of WSN data, data fusion, interpolation of data, big data analysis, visualization of data, policymaking, and citizen engagement.

Factors that influence the design of WSN are [2]:

- Reliability - sensor nodes fail due to physical damage, environmental effects, or power outage.
- Scalability - algorithms for network creation and communication must be scalable. Algorithms must be designed so that ad hoc sensor nodes can still work with existing networks and network does not breakdown because of large number of sensor nodes.
- Cost - hardware design and production of sensor nodes should be such that cost of single nodes are reduced. Sophisticated sensor nodes might cost high compared to tractional sensor nodes deployment. Maintenance of senor nodes and network for continuous monitoring is another major challenge.

4 Time Synchronization in WSN

Time synchronization is an inherent problem in any communication network; for instance, two computers connected to a common Local Area Network (LAN) will have a difference in their clock timings if both are left untethered to the Internet for a significant amount of time (few days). This may be attributed to physical, mechanical and electrical properties of the quartz crystal oscillators residing in each of the computers. However, computers connected to the Internet, maintain their timings equal, as they are synchronized to higher-level stratum time server using Network Time Protocol (NTP) [34]. To maintain accurate timings with moveable devices, Global Positioning System (GPS) can be used. GPS consists of a constellation of 27 satellites positioned in geosynchronous orbits such that at least 3 satellites are visible for a device on any geographical position on the Earth. These satellites use atomic clocks to maintain accuracy of 10 ns [3]. In case of low-cost and low-power networks such as wireless sensor networks, sensor nodes use lower-end quartz crystal oscillators as their clock source. As a consequence, the clocks experience time drifts and also skewness.

In order to address time synchronization problem in WSNs, numerous algorithms and protocols have been proposed by the research community. An early comprehensive survey can be found in [49]. Reference Broadcast Synchronization (RBS) [9], Timing-sync Protocol for Sensor Networks (TPSN) [16], Flooding Time Synchronization Protocol (FTSP) [33], Lightweight Time Synchronization for Sensor Networks (LTS) [19], Tiny-sync [45] and Mini-sync [45] are few of the most commonly known protocols. In this work, we are presenting a new algorithm to compensate the time error between nodes in an energy-efficient way. In this manner, we would be able to use sensor networks for different applications and also with the inclusion of error compensation (time synchronization), it will provide a twofold benefits to the system as a whole. The tests were conducted using Castalia-3.2 Simulator. A hierarchial network structure was created and the algorithm was compared against Timing-sync Protocol for Sensor Networks (TPSN) and Lightweight Tree-based Synchronization (LTS) methods (centralized LTS). The simulation was conducted for a period of 30,000 seconds with two re-synchronization intervals: 300 seconds and 1000 seconds.

4.1 Literature

Physical time plays a crucial role for WSN application [10], [28] and is important that the deployed nodes have identical time stamps. Due to the absence of real time clock on the WSN motes, a software scheme is required in order to synchronize their on-board clock. The importance of physical time with respect to WSNs are detailed in [41], where three scenarios are discussed : (a) interaction between the sensor network and the observer, (b) interaction between nodes, and (c) interface between nodes and the real world. Another problem with time synchronization is that once a time is set for a node, the node-time drifts over a period of time due to frequency drift. Hence, there is a need for clocks to be reset at pre-specified intervals or dynamic intervals such that the re-synchronization interval does not contribute to the inaccurate timings of the collected data. So far, many solutions are available in literature to tackle the problem: (a) devising software clocks, (b) adopting unidirectional synchronization method, (c) estimation of round trip synchronization time, and (d) synchronization using reference signals. All of these have their own merits and demerits for a given scenario, network, accuracy and constraints. There exist other schemes to reduce error by collecting samples at higher data rate [41].

Clock drift is shown to be a major source of time synchronization error [29]. As a result, frequent synchronization of the clock is necessary. However, frequent synchronization of the clocks come at the cost of energy. Krishnamurthy *et al.* [29] have also reported the synchronization accuracy requirement for modal analysis of about 0.6 to 9 μs. Wang *et al.* [54] have reported a drift of up to 5 ms in 6 seconds period, which definitely leads to a significant timing error if a network decides to synchronize less frequently to save energy. As pointed out by Krishnamurthy *et al.* [29], there is a need to synchronize the clock more frequently than originally intended and this will have a bearing on the lifetime of the network. In this context, a few time synchronization methods proposed in literature have been identified to be very useful and are outlined below.

Reference Broadcast Synchronization (RBS) [9] is a simple and effective way to eliminate the nondeterministic delay between the sender and the receiver. In this technique, a reference node sends a reference message and the receivers exchange their messages to calculate the offset. Elson *et al.* conducted the experiment under laboratory conditions and achieved the clock precision of 1.85 ± 1.28 μs between sender and the receivers. Receivers reported their received time by triggering a General Purpose Input-Output (GPIO) pin, which is in turn connected to an external Logic Analyzer to record the time from each of the associated receivers. Elson *et al.* have also reported a precision of 3.68 ± 2.57 μs for 4 hops. RBS is mainly devised on measuring the receiver-to-receiver synchronization error. Although authors claim that as simply as a periodic reference provider, a closer look into RBS reveals that it is equivalent to external reference time provider that provides signal periodically such as GPS. Furthermore, a reference node alone is incapable of providing timing signals to all the nodes because of limited wireless range. To little extent, communication can be achieved by increasing radio range up to maximum radio limits while sacrificing its energy. Additionally, if the number of reference nodes are increased to support the entire network coverage, there exists a problem of providing timing information to all the reference nodes themselves. Moreover, as the number of nodes increases, the computation and time required to calculate relative error increases: A node has to estimate the relative error among all the other nodes. Because of increased number of nodes it takes a fair amount of time to exchange timing information between nodes. Indeed, each node has to wait until the currently active node completes its message exchange. Therefore, each node has to keep a large random back-off time to allow sufficient time with currently active nodes. An inherent problem with this methodology is the re-synchronization interval has to be determined externally and therefore lacks system dynamism.

Timing-sync Protocol for Sensor Networks (TPSN) [16] is a hierarchical structure-based algorithm that aims to provide network-wide synchronization. It is built on the notion of “always-on" model. TPSN works in a two-step process termed as “level discovery phase" and “synchronization phase", to synchronize all network nodes to reference node. First, it creates a hierarchical structure of sensor nodes and later, synchronizing between each pair along the edges. Hierarchical structure is created by assigning node levels to each of the nodes. “Root node" is always at level 0, and node at level i is ensured to have communication ability with at least a node belonging to level $i-1$. Next, root node initiates the synchronization phase in which level at level i synchronizes to a node at level $i-1$. Ganeriwal *et al.* have reported an average accuracy of 20 μs between neighboring nodes tested on Berkerly motes. Figure 3 depicts the typical two-way message exchange between two nodes. TPSN estimates the drift between clocks and the propagation delay as given by equations (1) and (2) respectively.

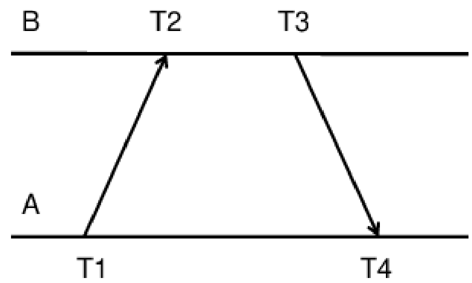


Figure 3: Two-way message communication between two nodes [16]

$$\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \quad (1)$$

$$d = \frac{(T_2 - T_1) + (T_4 - T_3)}{2} \quad (2)$$

The experiment setup included programming of two motes to toggle a particular pin (connected to the Digital Analyzer) every 8ms after two-way message exchange. Both the motes ran unsynchronized clocks. The error between the motes were measured by observing the phase shift between two waveforms produced by the motes. For 200 independent trials on both TPSN and RBS techniques, authors [16] report TPSN as roughly twofold better performing candidate than RBS. However, in a typical sensor network scenario, “always-on" model is collectively unattainable because of limited power reserve: sensor nodes run out of battery after specified amount of energy is expended. Additionally, due to unforeseen circumstances, a glitch that causes a sensor node to reboot might hinder the process of executing the algorithm successfully. In case of multi-hop scenario, sender synchronizes to receiver at every hop before data is transported. A careful analysis reveals that if the receiver at level $i-1$ is not synchronized to receiver at level $i-2$, then a data sent from level i will lose its true time because of time differences between consecutive receivers at levels $i-1$ and $i-2$, and further up to root node. In other words, first, the immediate root node neighbors must send the data and next, the nodes at subsequent levels must send the data in an increasing hierarchical order. Therefore, a node at level i must be aware of synchronizing times of nodes hierarchically above it. This may be lead to data latency by a leaf node as compared to data from higher-ups. Lightweight Time Synchronization (LTS) [19] is similar to TPSN where in, with the aid of spanning tree algorithms, a spanning tree is created and later pair-wise synchronization is performed along the edges of the spanning tree for multi-hop networks. The scheme requires three messages to perform a pair-wise synchronization, and error properties of the synchronization are modeled as Gaussian. The method requires $n-1$ pair-wise synchronization be performed for a set of n nodes. It has both centralized and distributed synchronizing schemes. Greunen *et al.* have shown that, for a multi-hop network, the communication complexity and accuracy depends on construction and related depth of the spanning tree; clock drift and accuracy are attributed to re-synchronization interval. The work carried out was aimed at reduced computation and complexity at the cost of relaxed accuracy. One major assumption of Greunen *et al.* is that the drift of the clock is bounded. Pair-wise synchronization is performed as shown in Figure 4.

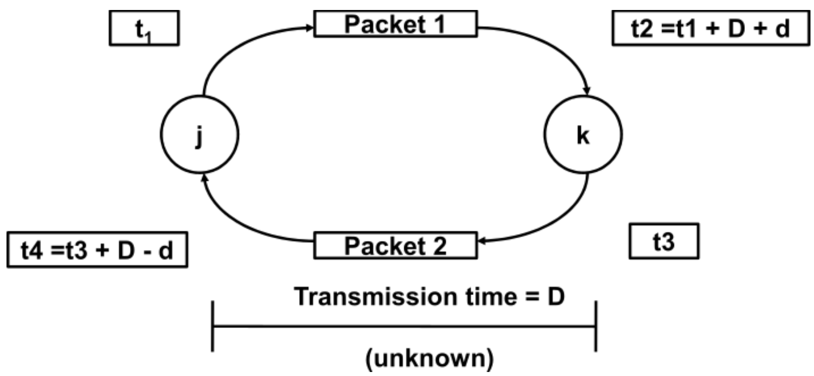


Figure 4: Pair-wise synchronization between two nodes [19]

Two nodes j and k are synchronized upon exchanging packets 1 and 2 and calculating the offset. A third message is used to convey the offset calculated by node j relative to node k. The offset d is calculated using unknown transmission time D calculated as given by equation (4)

$$t_2 - t_4 = t_1 - t_3 - D + D + 2d \quad (3)$$

$$d = \frac{(t2 - t4 - t1 + t3)}{2} \quad (4)$$

Flooding Time Synchronization Protocol (FTSP) floods the network with synchronization packets to the neighbors [33]. A global root node is elected and all other nodes synchronize their clocks to this node. The root node is dynamically reelected. A sender transmits its time by using byte alignment so as to reduce errors. In FTSP, a node is synchronized only upon having enough data (8 former data from the sender) to perform linear regression. A shortcoming with this approach is that if a node receives the data at less intervals, then it will not be in a position to complete the linear regression analyzes. Thereby, it deters any progress toward time synchronization : a situation my arise where in a node will never be able to synchronize.

In the recent past, very limited work has been focused on both time synchronization and energy efficiency simultaneously. More recently, Kim *et al.* [26] have developed an algorithm to increase efficiency by reducing the number of packets exchanged thereby conserving power. This again is a combination of RBS, TPSN and LTS methods but reduces the number of packets exchanged. Again, the topology used is identical to TPSN and the best achievable error rate is affected by the number of hops. Of all the methods discussed above, less importance has been devoted toward energy efficiency. Shahzad *et al.* [43] proposed Energy-Efficient Time Synchronization Protocol for Wireless Sensor Networks (ETSP) algorithm that combines RBS and TPSN for achieving better efficiency in energy. Although these methods improve energy efficiency, however, in large multihop network applications—such as real-time traffic management, health monitoring of a structure—with 8-10 multihop levels, the synchronization error is higher than required for critical infrastructure monitoring.

4.2 Clock Model

Clock can be modeled as a function of linearly increasing function with time i.e. $f(x) = m \times x + c$. When we say it is linear, it is assumed that the clock function will not fluctuate randomly rather maintains linear slope. This property was also observed during simulations. Let the subscript I represent the ideal clock (i.e. having slope $m_I = 1$). Then the clock function ($f(X_I)$) of an ideal clock is given by equation (5)

$$f(X_I) = m_I \times x_I + c_I \quad (5)$$

An imperfect clock has the slope $m > 1$ or $m < 1$. This implies that the clock is either faster or slower as compared to ideal clock.

The synchronization error is the time difference between two clocks at a given time t. The error (also offset) is caused by the skewness (first-order derivative of clock with respect to real time t). Therefore, the offset $f(O)$between ideal clock $f(X_I)$ and an imperfect clock $f(X_C)$ can modeled as:

$$f(O) = f(X_I) - f(X_C) \quad (6)$$
$$= [m_I \times x_I + c_I] - [m_C \times x_C + c_C] \quad (7)$$
$$= [(m_I - m_C) \times x_I] + [c_I - c_C] \quad (8)$$

because $(x_I = x_C = t;\ \forall x_I \in t, \forall x_C \in t)$ is true w.r.t. time t. The subscript C refers to a child node. The skewness is caused by the factor $m_C \neq 1$. c_I represents the offset at the starting of the timer. If $c_I\ or\ c_C > 0$ implies that the clock had a residual time. However, in general, $c_I = 0$ and $c_C = 0$. The following conditions are true for two clocks:

1. *Condition 1*—If $m_C > 1$, then $f(O) < 0$, indicating the child clock has a skew that is greater than m_I
2. *Condition 2*—If $m_C < 1$, then $f(O) > 1$, highlighting the fact that the child node has a clock that is running slower than the clock y_I measured at time t
3. *Condition 3*—if $c_I = 0$ and $c_C > 0$ results in a negative offset $(f(O) < 0)$at time $t = 0$ when measured with respect to ideal clock, inferring that the child clock has started before ideal clock
4. *Condition 4*—if $c_I = 0$ and $c_C < 0$, then this condition results in a positive offset with respect to ideal clock $(f(O) > 0)$ pointing that child clock is slower. This also directs that the child clock will have a positive value when $m_C * t > c_C$

A sensor node clock can be of any one of the three types (ideal, fast or slow). Therefore, the relative offset between clocks may be positive ($\frac{d}{dt}C_i(t) > \frac{d}{dt}C_j(t)$) or negative ($\frac{d}{dt}C_i(t) < \frac{d}{dt}C_j(t)$) for a node i with respect to node j depending on whether i is faster or slower than j. Figure 5 shows the linear variation of fast, ideal (perfect) and slow clocks.

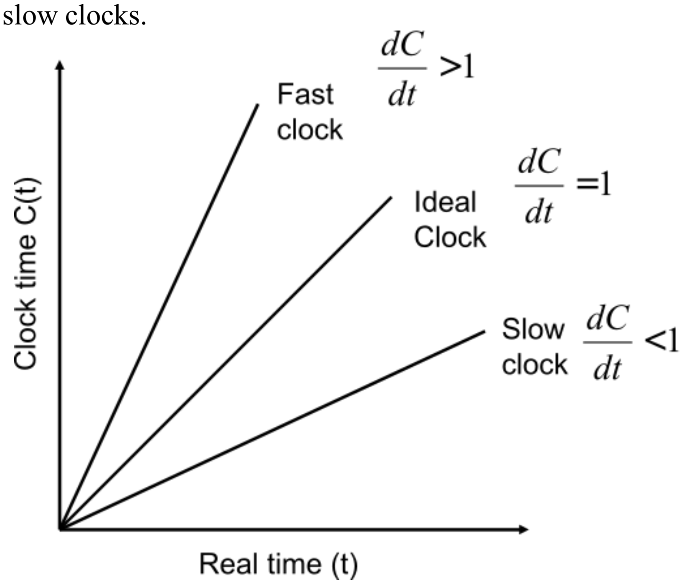


Figure 5: Fast, Ideal and Slow clocks [49]

It is very important to note that because of skewness of the clock, a fast clock samples faster than a slow clock. Thus, when a node is allowed to collect data, if it is not ideal, then both the collection time and collected data represent different timings.

5 Virtual Clock-based Time Synchronization (VCTS)

Our aim of time synchronization is to maintain a global time across the network through external synchronization (i.e. to maintain a precise time with respect to external source). For the purpose of synchronization we make use of implicit synchronization technique and adopt hierarchical structure-based network. The network is dynamically formed and nodes can be added at anytime. The communication mode is broadcast and communication distance is multi-hop. Our objective is to minimize the synchronization error while reducing the energy consumption.

Consider N nodes deployed as a group of sensor networks for monitoring certain activities. It is assumed that each of the N is in communication range from at least one other node and there exists one node among them that acts a root node (connected to external time source). A root node is a node that maintains the real time through external synchronization or through any other means. Every node has its own unique identification (ID) tag and are aware their tags. The sensor network time synchronization can be achieved by having a hierarchical structure followed by data transportation from leaf nodes, intermediate nodes to root node. This network creation is accomplished in two stage process: network creation and time synchronization. The flowchart for network creation and time synchronization are shown in figure 6 and the steps involved are described as follows.

5.1 Network Creation

1. The sensor network deployed creates its hierarchical structure being root node at level i and its immediate neighbors being at level $i+1$ (similar to TPSN, but not entirely). The immediate neighbors of level $i+1$ become level $i+2$ peers and so on. The process of network creation continues dynamically until all the nodes in the network have their levels determined—each node knows its parent (except root node) and each node knows its child (except leaf nodes).
2. Parent node sends a broadcast signal to determine neighbors
3. Parent node waits for t_w time units to receive “request" signals from child nodes.
4. After t_w time units, parent node estimates its children count and sends “level" signal for the child node. This “level" is sent only once upon determining a child for the lifetime(until child node requests “parent" as mentioned in step 6. Child simply receives the parent level and increments it by one and stores. The message also contains the current time of the parent. This child node now acts as parent and repeats steps from 1. Parent node processes each child according to ascending (or descending) order of identification number. After a child node is processed (by sending “level"), it processes the next child until all the child nodes are processed.
5. A node can sense a physical activity only after having a “level" signal from its parent and can send data to its parent whenever a data is available.
6. If a new node is added to (or change in topology is made—node displaced, node power cycled) the network, then it has to broadcast a request “parent" signal so that nodes nearer to it will respond. Each of the nodes will respond with their level numbers. The new child has the ability to chose the node with lowest level (closet to the parent) number as its parent node. After this an acknowledgment is sent to the parent. The parent will send a “level" signal to the new node member. Here acting-parent nodes responding to the new node respond only if they think that they can handle additional load from this newly added node and hence the new network. In this way, nodes can avoid additional onus being experienced.
7. Every time a data is received form a child, time synchronization is performed. Parent node updates the data received time as described in Section 5.2

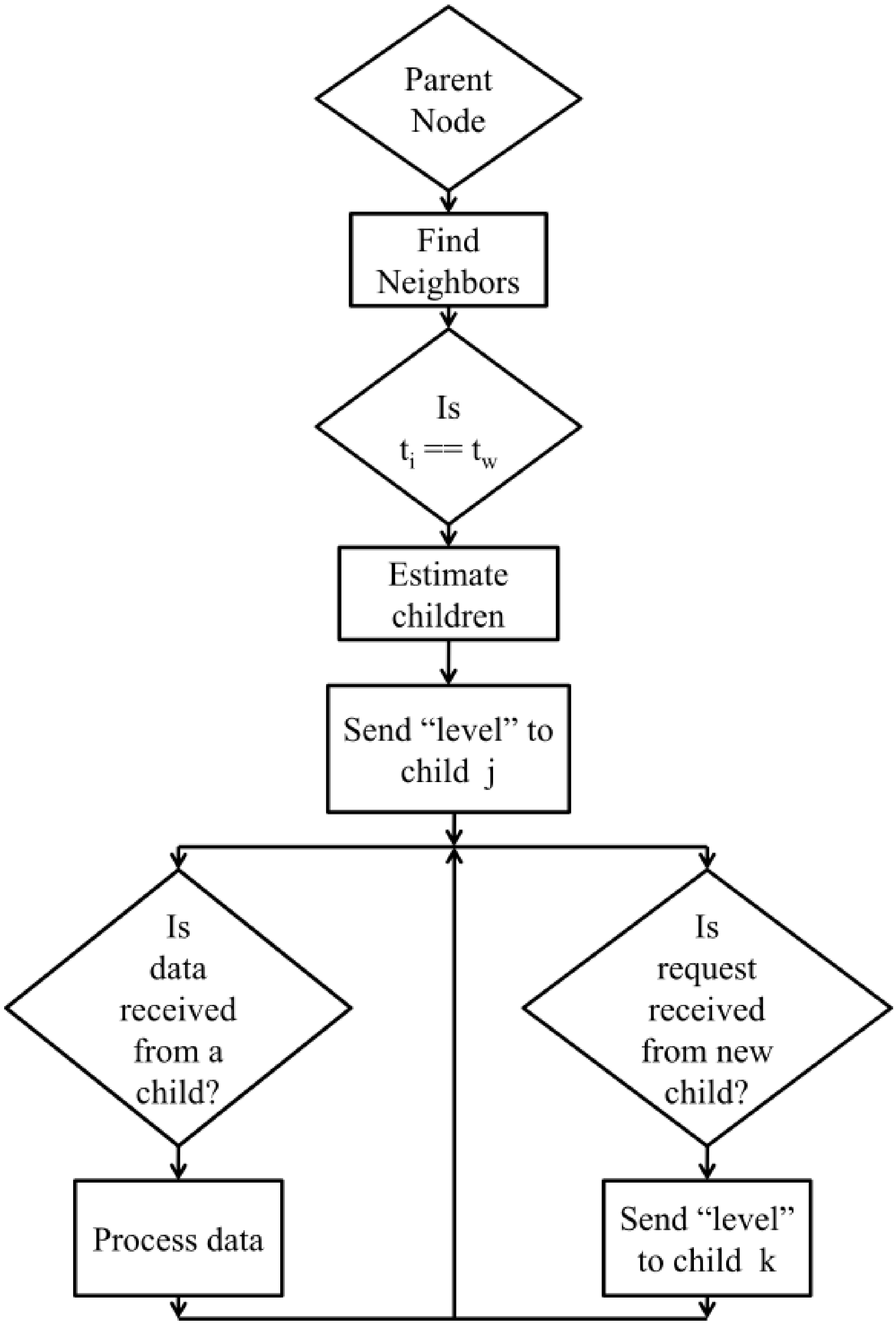


Figure 6: Procedure for network creation and time synchronization

5.2 Time Synchronization

Let t_i represent the local clock time of each the node and $i = 0,1,2, \ldots . N$. For simplicity, the algorithm described below is between any two nodes; that is, a parent and child node. This can be extended to single-parent multi-child situations. Let i represent a parent node and j represent child node among N set of nodes. Let t_{ij} represent the relative clock time difference (offset) between nodes t_i and t_j, where $t_{ij} = t_i - t_j$ w.r.t. to its parent i. In this paper all the time difference conventions are from parent's perspective. As the time elapses, there will be time differences between parent and child: if the derivative of clock source of the child is fast clock ($\frac{d}{dt} C_j(t) > 1$), then $t_{ij} < 0$, and $t_{ij} > 0$, for $\frac{d}{dt} C_j(t) < 1$. Therefore there is a need for manipulation of child's time individually by the parent. The relative synchronization error at any point in time between two nodes is given by:

$$t_{re} = t_{ij} \quad (9)$$

$$= t_i - t_j \quad (10)$$

The above set of equations are devoid of adaptively corrective mathematical terms that foster the sensor nodes to keep track of relative time. The relative time from parent's perspective can be obtained as:

$$t_{j_{re}} = t_i - [(t_i - t_{i-1}) - (t_j - t_{j-1}) + t_{j_{te}}] \quad (11)$$

$$= t_i - [(t_p) - (t_c) + t_{j_{te}}] \quad (12)$$

where tj_{re} symbolizes relative error, $t_{j_{te}}$ represents total relative error measured by parent i for child j up to time t_i. t_{i-1} represents the parent time when a message was previously received from the child node with the child time as t_{j-1}, which was time stamped along with the message. t_p corresponds to the time elapsed since the current and previous message as observed by parent by its self-timing $t_p = t_i - t_{i-1}$, for child with timings $t_c = t_j - t_{j-1}$. The relative error is updated every time a new message is received by using (13)

$$t_{j_{te}} = t_{j_{te}} + t_{j_{re}} \quad (13)$$

Note that $t_{j_{te}}$ could be positive or negative. It is observed that, when there is momentary variations in child or parent clock, $t_{j_{re}}$ swings drastically, but maintains a steady state after 2-3 successive measurements. In physical terms, $t_{j_{te}}$ is the sum of all relative time differences between a child and a parent until recent observation. In other words, parent node i is maintaining a cumulative account of relative offsets for child node j. The relative offset error is a combination of both clock offsets due to drift and nondeterministic delays. In equation (12), parent i is correcting child's clock at each message communication instances. Therefore, relative error between parent i and child j remains extremely precise. Using this method, the relative offset reaches a steady state error value for all t as shown in Figure 7, for a parent and child node. $t_i \pm \beta$ represents the relative offset between parent and child. Here, parent assumes itself as ideal clock (for the purpose of calculation). $t_0 \pm \alpha$ would provide actual time (real time) between parent and child. It is to be noted from Figure 7 that for a fast clock, relative to parent ($t_i + \beta$), the actual time is ($t_0 - \alpha$), and for a slow clock ($t_i - \beta$), the actual time is ($t_0 + \alpha$), measured at actual time t_0. $(t_p - t_c)$ corresponds to $t_i \pm \beta$ shown in figure 7. $(t_p - t_c) > 0$ for slow child clock, whereas $(t_p - t_c) < 0$ for fast child clock. $(t_p - t_c)$ provides an estimate of child clock skew for the time period $t_i - t_{i-1}$. Our algorithm implements equations (12) and (13). The following sections provide improvements that can be undertaken to further improve the accuracy: (a) Section 5.3 is about improving the accuracy taking packet size in to consideration, (b) Section 5.4 is for predicting approximate time to estimate synchronization interval and time transformation as post-facto synchronization, and (c) Section 5.5 highlights about short-term and long-term synchronization techniques.

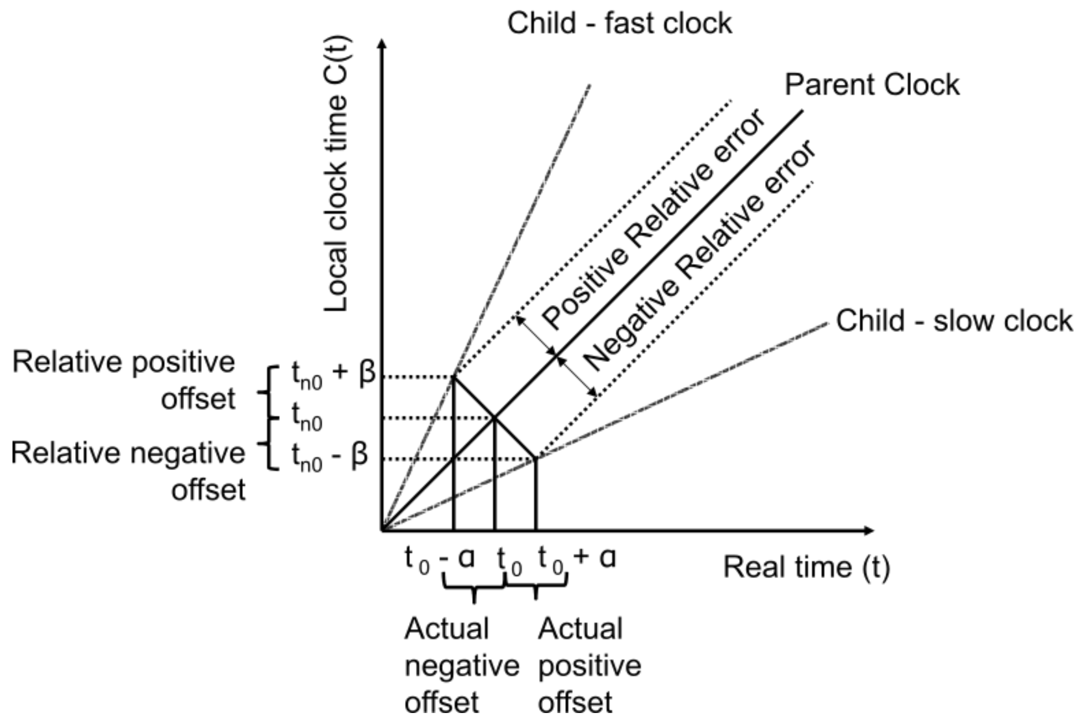


Figure 7: Relative Error stabilized using equation (12)

5.3 Algorithmic Improvements

Algorithmic improvements can be implemented to further lessen the synchronization error by considering the notion of time of reception that it takes more time for a sensor node to receive large-sized packet data, as against small-sized message packet. Given two nodes i and j, there is time associated with the length of the message. Let t_i denote the reception time of data at node i and let t_j represent the sending time at node j. Under ideal conditions, if node j is commanded to send k number of $n1$ bytes of data to node i at regular intervals t_s and also to send k number of $n2$ bytes of data, where $n2 > n1$, then the reception time difference t_d at node i is given as :

$$t_d(i) = t_s(j) - t_s(j-1)$$
$$t_d(i+1) = t_s(j+1) - t_s(j)$$
$$t_d(i+2) = t_s(j+2) - t_s(j+1)$$

.

.

.

$$t_d(i+k) = t_s(j+k) - t_s(j+k-1)$$

If we subscript i by i_{n1} and i_{n2}, then, since t_s is constant for all messages, it is found that $t_d(i_{n1}+k) \neq t_d(i_{n2}+k)$, for $n2 > n1$. Therefore, in sensor network communication, for different size of packets, equation (14) holds true $\forall t_s = K$, where K is data sending interval.

$$t_d(i_{n2}) > t_d(i_{n1}) \tag{14}$$

Using equation (14), equation (12) can be rewritten as:

$$t_{j_{re}} = t_i - [(t_p) - (t_c) + t_{j_{te}}] - t_d \tag{15}$$

where t_d represents time delay introduced due to sending time, access time, propagation time and reception time (we consider propagation time to be negligible for WSNs wireless range compared to radio waves' speed). More information can be found in [40].

5.4 Time Prediction and Transformation

At any point in time, if parent node has the information of the slope of the child clock, then parent can compute its child's predicted offset (P_o) by using equation (16).

$$P_o = [m_I - m_C] \times t \tag{16}$$

$$t_i = (t_j - t_{j-1}) + t_{j_{te}}, \tag{17}$$

where t_j is the current time, t_{j-1} is the time when $t_{j_{te}}$ was computed for node j. In general, equation (17) can be used to send the current time of a parent node to child node. If the child node is a new node, then $t_i = t_j$, else it has already been known to parent j and can make use of equation (17). In case of multi-hop scenario, to determine the time of node at level L distance is given by (18)

$$P_o(n) = [1 - \sum_{n=1}^{L} m_n] \times t, \tag{18}$$

where $m_n = m_1 + m_2 + \cdots + m_{n-1} + m_n$ up to nth node, m_1 is the skew between parent and its child at level 1; m_2 is the skew between child at level 1 and child at level 2 and so on, up to L levels.
If the slopes of all the nodes are available (considering nodes maintained their slopes ever after their power cycle), then time transformation can be applied on the data available from all the nodes at the base station. If δ_{max} and δ_{min} represent maximum and minimum offset limits, then, base station node can determine the exact occurrence of an event with respect to its time by estimating the occurrence (P_{ce})of an event using equation (19)

$$P_{ce} = t \times (m_b - m_n), \tag{19}$$

where n represents node number (excluding base station slope m_b).

5.5 Computing Re-synchronization Intervals

In order to lower the relative error between children and respective parents, a dynamic inter-synchronization interval scheme is essential. A predetermined, fixed inter-synchronization interval may often satisfy the error requirements; however, as the system scales into bigger entity, it is less likely that this static inter-synchronization interval to efficiently confine to the error limits.

At any instance, the time predicted for a child j by parent node i is given by equation (17). The re-synchronization interval is estimated using upper limit of time differences (δ_{max}). Let δ be a threshold limit, a measure of time differences between the parent and the child, then $\delta = |t_i - t_{j_{te}}|$, gives the threshold value. If $\delta > \delta_{max}$, then parent unicasts a message containing the child time. Therefore, using δ_{max}, synchronization interval can be set. The reason for having t_p and t_c is to measure the drifts in short terms, whereas $t_{j_{te}}$ is a measure of long-term (re-synchronization) interval. This helps to detect any large deviations in the child's clock. In turn, child's clock can be corrected using equation (17). Using this method, it adds dynamism to calculate re-synchronization intervals.

5.6 Determining Consumed Energy

Energy consumed by a sensor node, in general, is a combination of two cardinal factors: (1) energy consumed linearly when the sensor node is functioning (processor—executing operations during awake and sleep times, with other member components), and (2) energy expended in transmitting and receiving a packet.

Let the energy consumed by a node for processing operations be E_P Joules and E_C Joules due to communication activities. Let the data transmission rate of a node (i.e. radio) be d_{tx}, then the time it takes for a byte to be transmitted is $t_{tx} = \frac{1}{d_{tx}}$. Let p such bytes form a transmitting packet size and q bytes form a receiving packet size. Additionally, for generality, let the data reception rate be $t_{rx} = t_{tx}$, same as data transmission rate. Let the current required for transmitting a packet be C_{tx}, for receiving a packet C_{rx} and the battery voltage be V_{bat}. Energy expended by a node is given by equation (20) – (22).

$$E_T = E_P + E_C \tag{20}$$
$$= [E_{pw} + E_{ps}] + [E_{tx} + E_{rx}] \tag{21}$$
$$= [E_{pw} + E_{ps}] + [(C_{tx} \times V_{bat} \times t_{tx} \times p) + (C_{rx} \times V_{bat} \times t_{rx} \times q)], \tag{22}$$

where, E_T is the total energy consumption of the node, E_{pw} is the energy expended due to node's awake status and E_{ps} is due to node's sleep status, and $[E_{tx} + E_{rx}]$ due to transceiving operations.

5.7 Simulation Results

Our results were verified using Castalia-3.2 simulator. OMNeT++ is an discrete event simulation environment. It is an extensible, modular, component-based C++ simulation library and framework essentially build simulators [53]. At the core of the OMNeT++, it provides option for component architecture model that can be programmed using C++. At higher-level, these components can be linked together (using .NED) [53]. Castalia is a discrete event simulator that runs on OMNeT++ simulation framework. It can be used to simulate WSNs, Body Area Networks (BANs) and also low-power embedded devices. It is a not specific to any sensor and provides a range of realistic node behaviors. It provides flexibility in its modularity, reliability and celerity of execution that are partly attributed to OMNeT++ support [37].

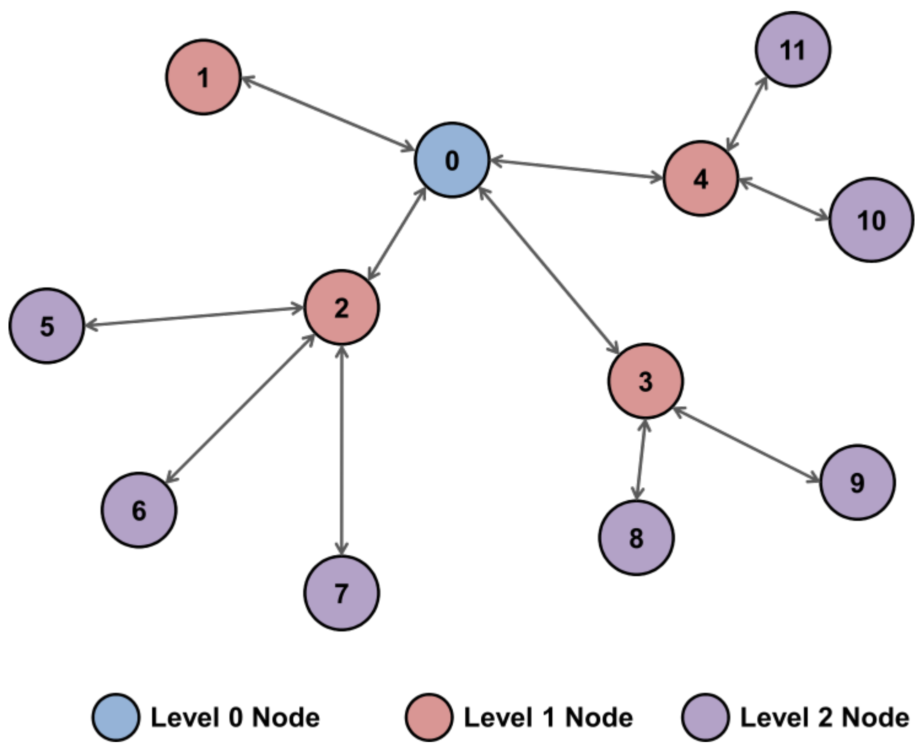
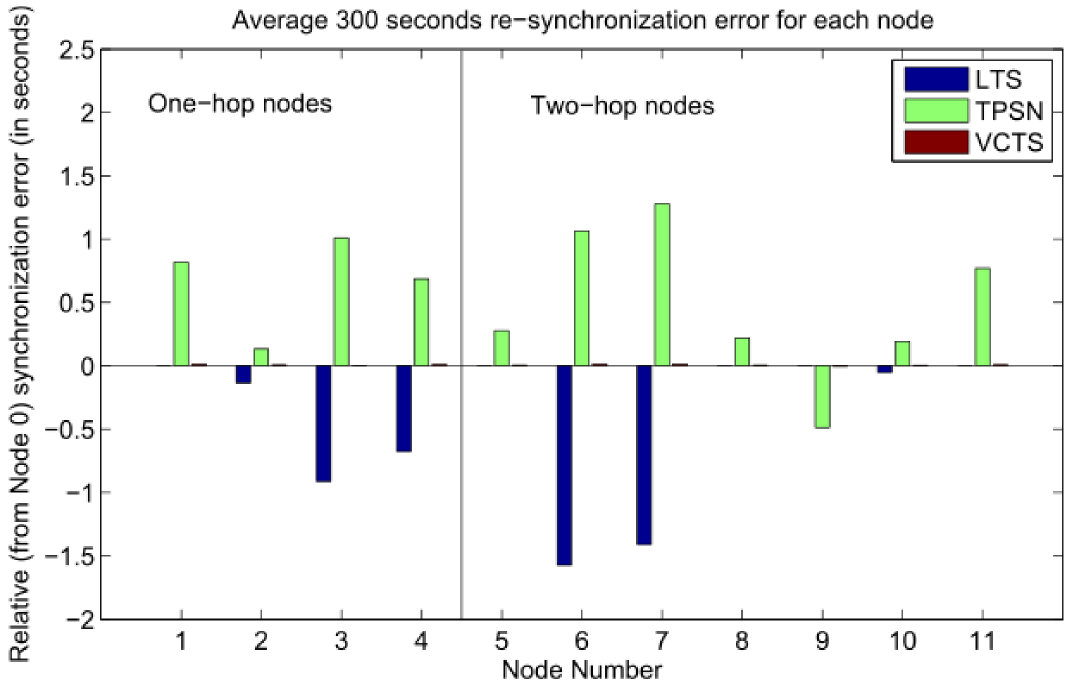
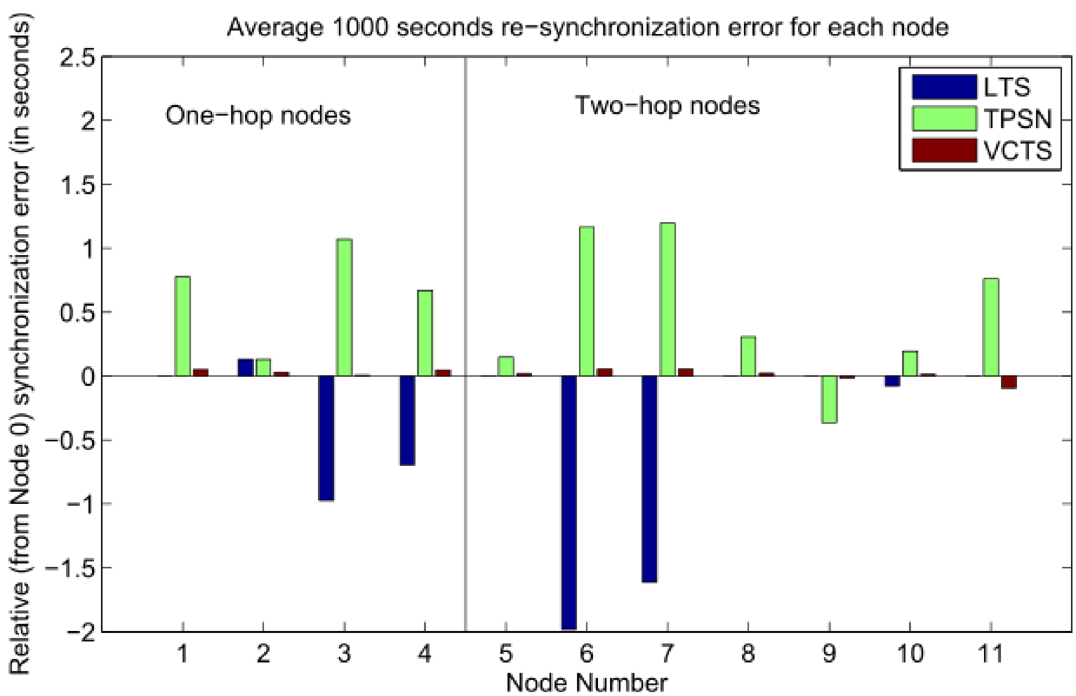


Figure 8: A common hierarchical structure for TPSN, LTS and VCTS

Simulations were conducted for a total of 30,000 seconds. A total number of 12 nodes were included to form a cluster with node 0 being the cluster head as shown in figure 8. All the nodes were defaulted to receive state upon completion of transmission. CC2420 radio parameters, 'IDEAL' conditions as specified in Castalia Simulator were used. We implemented radio disk model with a radius of 22 m. Neither MAC nor any routing protocol was used. Application layer time stamps were used instead of MAC layer. Synchronization error obtained for three algorithms (TPSN, LTS and VCTS) are shown in figures 8 and 8, for a re-synchronization interval of 300 seconds and 1000 seconds respectively. Total energy consumed and energy consumption for communication only, by these methods are shown in figures 9 and 9. Energy for the communications was calculated off-line using E_c of equation (20) and combined with E_P, obtained from simulator for 30,000 s. For calculating E_C, all data messages were set to 50 bytes in length, $d_{tx} = 250$ kbps, $V_{bat} = 3.6$ V, $C_{tx} = 17.4$ mA and $C_{rx} = 18.8$ mA.

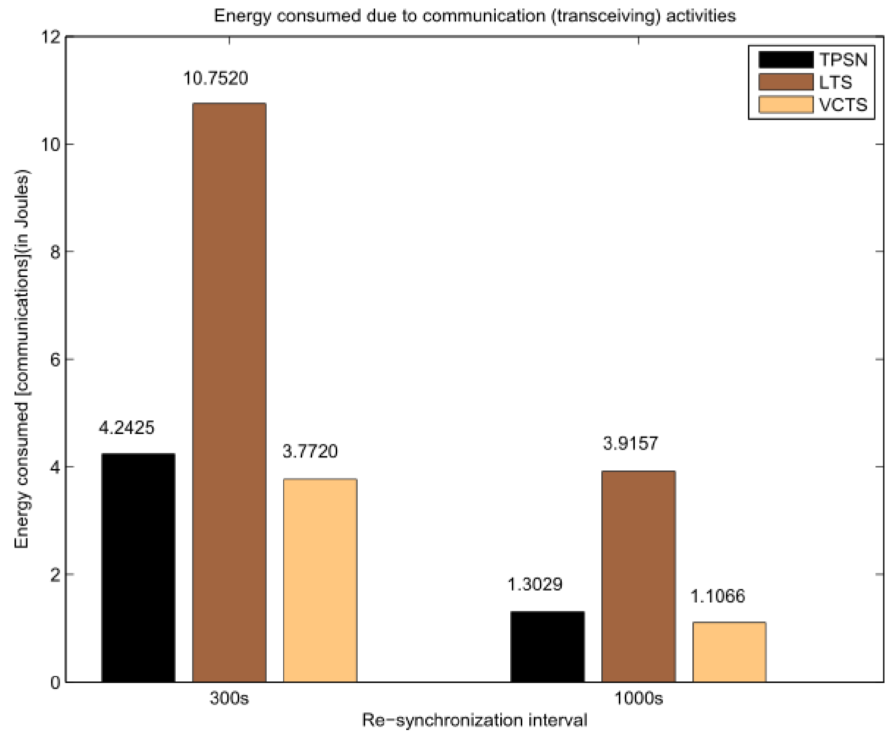


(a)

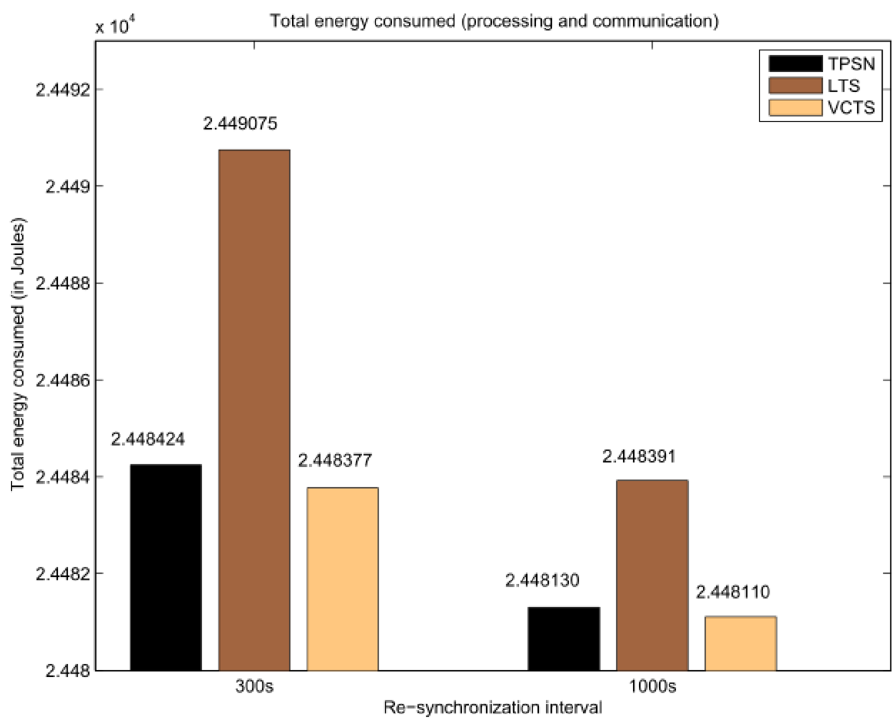


(b)

Figure 9: (a) Synchronization error for 300 seconds re-synchronization interval and (b) synchronization error for 1000 seconds re-synchronization interval.



(a)



(b)

Figure 10: (a) Energy expended by three different protocols (for transceiving operations), and (b) total energy expended by three different protocols: TPSN, LTS and VCTS.

5.8 Discussion

Initially, synchronization error due to different re-synchronization intervals will be discussed followed by their energy consumption.

First, for 300s re-synchronization interval, TPSN maintains a moderately increased error from one-hop (0.526633 s) to two-hop (0.613173 s) distanced node. The one-hop error (0.573529 s), for LTS, increased steadily to two-hop distance error (1.011589 s). On the other hand, VCTS maintained a relatively unvaried error from one-hop (0.009964 s) to two-hop (0.009074 s). Next, for 1000 s re-synchronization interval, TPSN performed better by declining its error to 0.590545 s from 0.660078 s. However, LTS followed its former position to reach a maximum error of 1.223768 s, during two-hop, from almost half the one-hop error. Nevertheless, VCTS had a moderate increase in error from 0.032712 s (one-hop) to 0.038614 s (two-hop). From figures 8 and 8, it can be inferred that, while TPSN and LTS maintained a steady trend in error values, VCTS quadrupled its error from 300 s to 1000 s re-synchronization interval. However, these errors are much below the values of both TPSN and LTS.
The energy consumed due to combined activities—nodes'awake state, sleep state and communication—is shown in figure 9, and that the energy spent in communication only is depicted in figure 9. The energy consumed by the entire network for different protocols, for 30,000s is approximately 24,480 J, and the remaining is contributed by message exchanges. For 300s re-synchronization interval, energy expended by TPSN is 4.2425 J, LTSN is 10.7520 J and VCTS is 3.7720 J. Likewise, energy consumed for 1000s interval are 1.3029 J, 3.9157 J and 1.1066 J respectively.

The proposed method performs better than TPSN and LTS, both in terms of synchronization error and energy efficiency. It is evident from the results that, LTS relaxes accuracy constraints while reducing accuracy, confining to authors claim. It is important to note that energy can be saved at the cost of sacrificing synchronization error in all the three methods. Low-Energy Adaptive Clustering Hierarchy (LEACH) is a clustering-based energy distribution protocol that distributes the load of local cluster head for communication between cluster head and the base station. It uses randomized-rotation of cluster head to balance the local cluster head energy [22]. Numerous LEACH-based clustering algorithms are also implemented [21], [30], [32], [56], [60], [1], [14], [51]. The combination of LEACH (for cluster-energy distribution) and VCTS (for time synchronization between nodes in a given cluster) can assist to design better energy-efficient time synchronization approaches as shown in Figure 11.

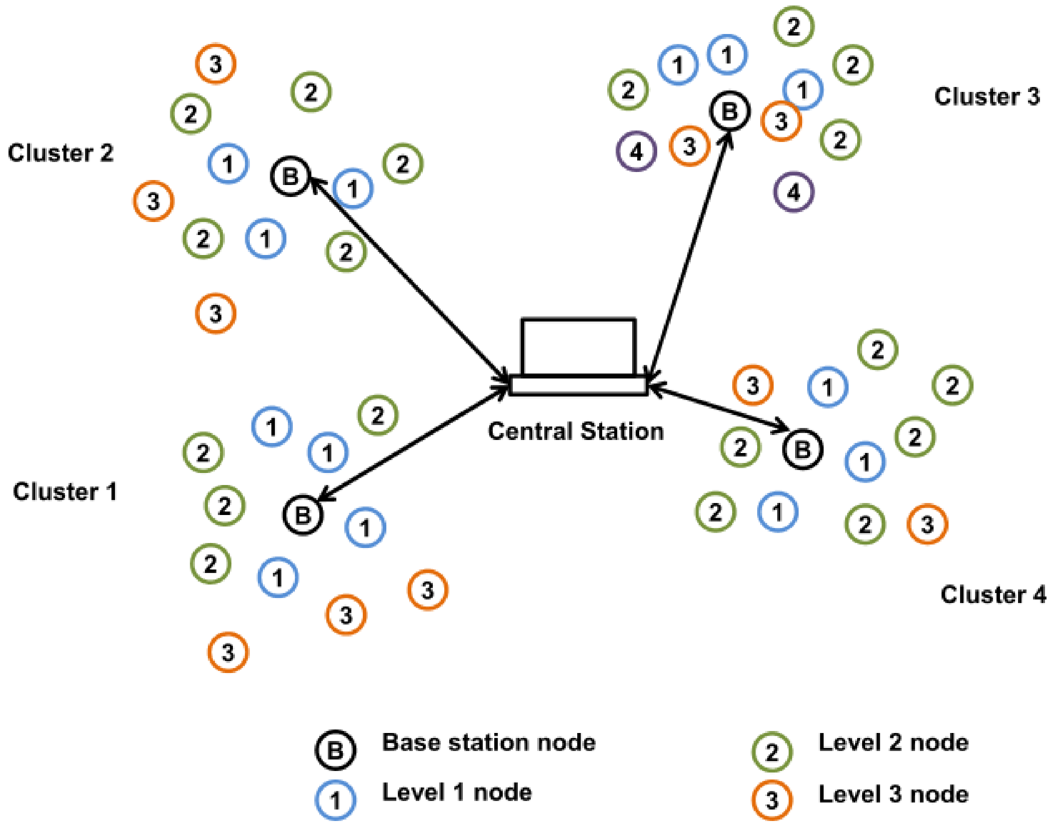


Figure 11: Schematic of cluster formation using LEACH. Numbers within the node represent hierarchical levels. B is the chosen cluster-head in any given round by LEACH.

An appropriate time synchronization algorithm is instrumental in devising an efficient sensor network. Maintaining the reliability of the network data, mainly the occurrence of the event, helps to detect an event of interest and act upon rightly. A data that has anomalies in its time-stamp due to synchronization error is of less conducive. From an SHM perspective, this is crucial for civil and infrastructure engineers. Hence, it is of great importance that, we minimize the synchronization error introduced by the low-cost solution. The use of precision clocks, certainly, accounts for expensive solution. Instead, a low-cost solution, at the expense of computation would be a remarkable progress. Therefore, time synchronization while adhering to stringent energy awareness is what is required. The accuracy of synchronized time among nodes, coupled closely (in time), ameliorates the system's ability to determine an event. Eliminating relative drifts and nondeterministic delays (send time, access time, propagation time, receive time), considered simultaneously, would provide promising results in synchronizing time. In the proposed approach, we are maintaining a relative offset errors of child nodes, for a parent node to use the data from the child and accommodate it to its local clock timings. Energy can be reduced by means of reducing communication overhead for synchronizing tasks. Also, time of a child can be predicted and error can be maintained at minimum level. The performance of the proposed algorithm is significantly better than TPSN and LTS.

6 Structural Health Monitoring using Emerging Internet of Things

Internet of Things (IoT) is an emerging technology that unifies sensors, sensor nodes and the Internet backbone into a single platform. The embedded devices with sensors can communicate with each other using the Internet. Users can also control these sensors in real-time over the Internet. Cloud computing technology has also immensely fostered the growth of (IoT). The state-of-the-art IoT infrastructure includes cloud platform for storage and retrieval of sensor data. Many useful application programming interfaces

(APIs) have been made available open to users for ease of use and open source. For instance, Libelium [65] provides IoT based solutions, where the embedded devices (Waspmotes) are programmable using Integrated Development Environment (IDE) that significantly reduce the development time. They also have a platform called Meshlium, which is Linux machine that has the Internet connection (3G, 4G), database server, memory storage, wireless transceiver (ZigBee and DigiMesh to receive data from Waspmotes). The users can further install the required packages as the platform is a Linux operating system. One can store the data in local database or send the data to external database using different protocols. Cloud platforms like Xively [66] provides modern data transfer services with Representational State Transfer (REST) APIs, can be used to upload data to cloud server for visualization of the sensor data.

6.1 Flexibility with Networking Similar to the Internet

The embedded devices with sensors have the ability to communicate mutual and form a network among themselves. The network can be formed dynamically on the availability of the resources (such as power, node failure). This feature provides a flexibility in networking. The same versatility is extended when connecting to the Internet. Each of the nodes can be made to connect to the Internet and communicate. This is highly advantageous since the failure of gateway node in otherwise static network, the critical sensor information will be lost. The ability of ad hoc network formation overcomes the problem of node failure and hence the data loss. Depending on the energy availability, any node can become a cluster head in controlling the network as observed in LEACH [22].

6.2 IPV6 Addressing

The Internet ("network of networks") uses unique address to identify the devices connected to the network. Usually we use names, for instance, http://www.google.com to access services from Google. However, devices on the Internet do not understand this language. The machines use Internat Protocol version 4 (IPv4) to resolve the names. A Domain Name Server (DNS) matches the names with IPv4 address and helps the devices on the Internet to communicate with each other. IPv4 address (32 bit addressing scheme) can handle $2^{32} \simeq 4.3$ billion unique devices. Since the inception of the Internet, there are an estimated 3 billion users (refer to Figure 12) covering 40% of papulation, with at least a million users being added per year. Additionally, there are many devices ("objects" or "things") connected to the Internat. A staggering 8.7 billion devices were reported to be connected to the Internet [63], already exceeding the IPv4 limit of 4.3 billion. Due to networking strategies (subnetwork mask), many devices can be connected to the same IP address using Network Address Translator (NAT). However, with IoT devices expected at 50 billion devices by 2020 [63] and unique identifiers to connect to the devices, new addressing schemes are required. In order to handle more number of unique devices, IPv6 standards [62] was proposed, which has the addressing (128-bit) capability of handling $2^{128} \simeq 3.4 \times 10^{38}$ addresses. many devices also have the mechanism to understand both IPv4 and IPv6 addressing, making them to communicate during transition from IPv4 to IPv6. The IoT devices posses low power and ad hoc networking, hence having NAT strategy is not ideal. The future IoT devices are being envisioned to have IPv6 address so that the devices can be uniquely identified.

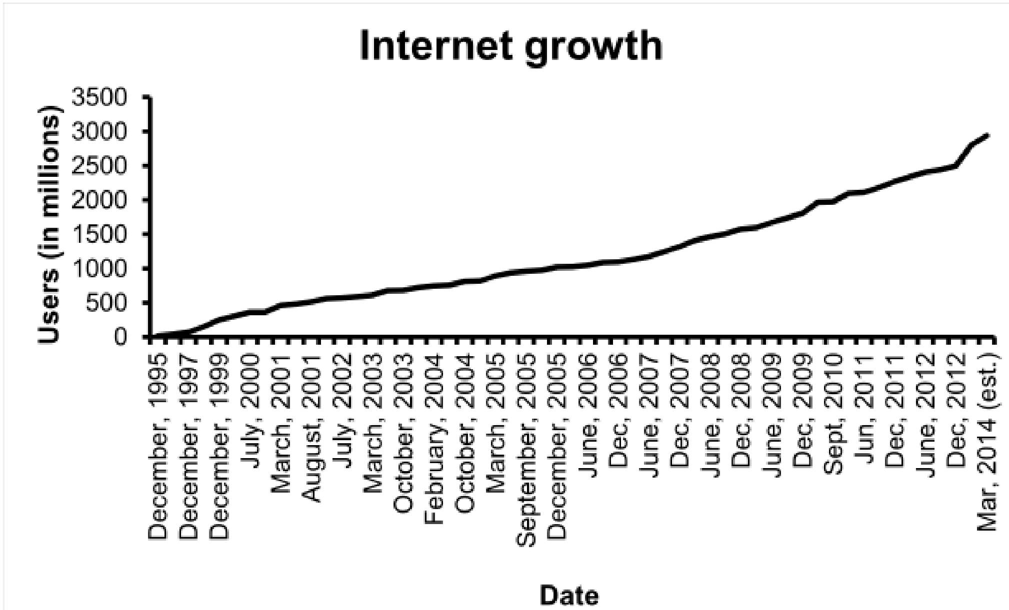


Figure 12: Growth of the Internet (in terms of users) since 1995 [64]

6.3 New Sensors with the Internet Connectivity

With IPv6 addressing, the new devices and sensor themselves can be assigned unique address. This provides an utility for the users a high-resolution and finer-scale knowledge of the senors and controls. The devices can connect to the Internet in ad hoc mechanism and eliminates many networking issues when there is a failure with individual devices. The device themselves have radio frequency comminations, which is used for ad hoc networking among the nodes. The dynamic nature of nodes having the communication and networking strategies almost provides a vision of machine to machine (M2M) communicating architecture. This adds increased flexibility in networking among the nodes and also to and from the nodes. The flexibility will be similar to the Internat. The bridges and the buildings can be monitored in an unprecedented way. The data resolution for scientific and engineering study will be unparalleled compared to the traditional monitoring.

6.4 Big Data Analytics

As number of sensors and sensor locations increase, we run into the problem of "Big Data". Deluge of data from different sensors and different modalities will be problematic for conclusive analysis. Big data analytics deals with handling multi-modal and heterogenous data for meaningful analysis. The existing data analysis algorithms prove to be inefficient in terms of processing time and processing techniques. A new line of analysis is the Big Data analysis. The algorithms are handle sophisticated homogenous, heterogenous and multi-modal data effectively to provide a better analytics. There has been considerable efforts by the research communities in addressing Big Data problem.

6.5 A Case Study of SHM

In this section, a case study of continuous monitoring of SHM using WSN have been presented. The aim of the study [17] was to address the following challenges in SHM: (1) a three-dimensional 14-bay truss structure with existing damage was verified, (2) continuous online SHM to measure ambient vibrations, (3) SHM through distributed computing using densely deployed sensors.

In a traditional SHM system, a central base station would acquire the data from sensors deployed at different locations of the structure. Those sensors that are close to damage were likely to be more pronounced in their outputs. Additionally, the long wires running would complicate the measurement process. The data acquisition becomes further burdened with large amounts of data to be transported and processed at the central station. With the advent of smart sensors [47], [48] — sensors with programmable microprocessors, memory and processing — has reduced load of central data processing, communication overhead while proving continuous SHM status.

The damage to the structure is detected by using flexibility-based damage localization method. The damage load vectors (DLVs) help determine the structures with damages. The sensors for undamaged structure with static forces produce zero stress at the sensor locations. This helps to localize the local structural damage. To verify this, a 5.6 m long three-dimensional truss structure was used. Steel tubes with 1.09 cm inner diameter and 1.71 cm external diameter formed the member of truss. The truss was excited vertically using Ling Dynamic Systems shaker. The response of the systems were measured using accelerometers (sensitivity of 100 mv/g, $1 - 4000$ Hz and range of ± 50). Spectrum analyzers were used to measure the responses.

In traditional SHM, the excitation requires external input and also the installation of machinery becomes expensive. Continuous online SHM would be an ideal solution. However, the DLVs for online method is not easily obtained. To overcome this, modal normalization constants were obtained for the construction of flexibility matrix. However, the construction of the flexibility matrix for a damaged structure is not feasible. To this end, extension of DLV was proposed to handle online damage [18].

To monitor large structures, the traditional way of using long cables is cumbersome and expensive. Also, from a safety and reliability perspective, continuous monitoring is necessary. WSN approach demands more bandwidth as more sensor nodes are deployed. To overcome this problem, distributed computing system (DCS) was employed. Small Clusters of sensor nodes work in tandem to process the sensor data and the “manager” sends the essential and limited data about the structure to the central station. Neighboring managers need to communicate with each other before sending the data. If the structure is not undamaged, “Ok” message about the location of the structure is sent to the central station by the manager. Extended DLV was employed by the nodes in the clusters to determine the damage to the structure in the locality.

6.6 A Demonstrable Application of Internet of Things

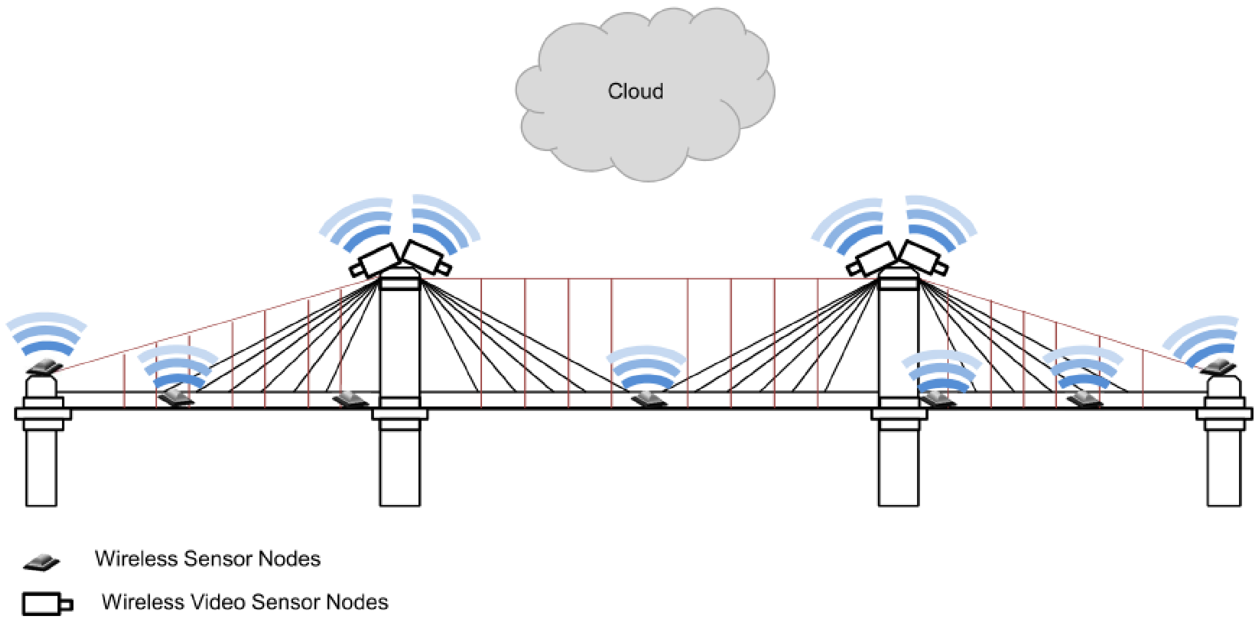


Figure 13: SHM using Internet of Things (IoT). All devices are addressable using IPv6 standards. Deviceswill make use of cloud services to store data and use analytics for real-time processing of the data at the node level.

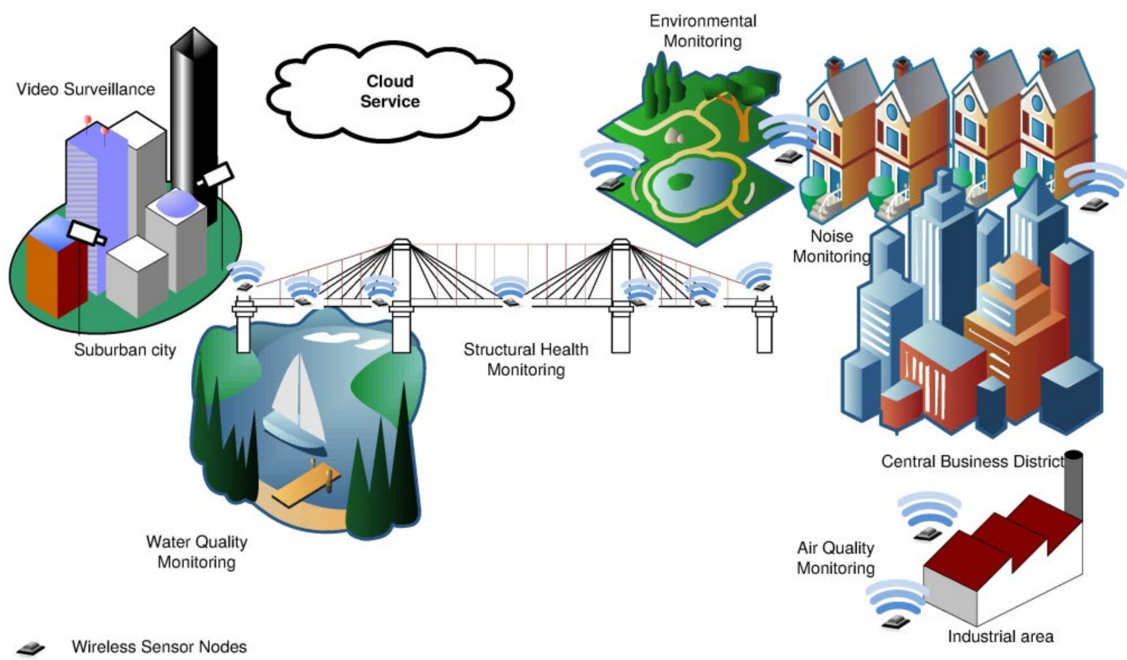


Figure 14: SHM using Internet of Things (IoT)

Internet of Things (IoT) can further ameliorate the above-mentioned approach by providing cloud-based services such as Software as a Service(SaaS), Platform as a Services (PaaS), and Infrastructure as a Service (IaaS). The IPv6 address to individual sensor nodes make the sensors authentic before providing such services. Figure 13 shows the SHM scheme using IoT. Devices (or "things") will be able to directly interact with the cloud services for data uploading and analytic for real-time processing at the node level. A wide range of analytics in real-time can be provided to the engineers, scientists and the users simultaneously. The central station can also be eliminated.

Further, carbon nanotubes as structure will be enable different sensing modalities based on their structural properties. The "smartness" will be embedded into the structure itself. This would help reduce the number of sensors required and consequently the cost. Additionally, with sufficient advancements, carbon nanotubes would be help to detect the local damages to the structure through global measuring techniques because of their intrinsic properties.

Video cameras can also be used for SHM. With low-coast cameras and advanced video processing algorithms, some of the visible signs of the structural damage can be easily detected using computer vision algorithms. Figure 13 shows the four video cameras monitoring the bridges. Additionally, usage of bridges by pedestrians and traffic can be monitored in real-time. Pedestrian monitoring aids in detecting the loitering behaviour, occupancy of the bridges by pedestrians at different times of the day and their general behaviour. Computer vision algorithms can also assists in monitoring traffic by : (1) analysing the vehicles for lane departure, (2) categorizing the vehicles based on size, load, and passengers, (3) detection of vehicle hazards such as smoke and reckless driving, (4) as early warning systems for incoming traffic about status of the traffic flow on the bridges and underground tunnels, and (5) also in automatic collision avoidance systems between vehicles and structures. Video camera networks will add rich source of information to the SHM to the existing SHM sensor data.

7 Conclusion

Internet of Things is an emerging technology that has promising future in continuous monitoring and controlling the connected devices. SHM can be achieved in real-time and rich analytics. However, sensor nodes face the problem of time synchronization error. An appropriate time synchronization algorithm is instrumental in devising an efficient sensor network. Maintaining the reliability of the network data, mainly the occurrence of the event, helps to detect an event of interest and act upon rightly. A data that has anomalies in its time-stamp due to synchronization error is of less conducive. From an SHM perspective, this is crucial for civil and infrastructure engineers. Hence, it is of great importance that, we minimize the synchronization error introduced by the low-cost solution. The use of precision clocks, certainly, accounts for expensive solution. Instead, a low-cost solution, at the expense of computation would be a remarkable progress. Therefore, time synchronization while adhering to stringent energy awareness is what is required. The accuracy of synchronized time among nodes, coupled closely (in time), ameliorates the system's ability to determine an event. Eliminating relative drifts and nondeterministic delays (send time, access time, propagation time, receive time), considered simultaneously, would provide promising results in synchronizing time. In the proposed approach, we are maintaining a relative offset errors of child nodes, for a parent node to use the data from the child and accommodate it to its local clock timings. Energy can be reduced by means of reducing communication overhead for synchronizing tasks. Also, time of a child can be predicted and error can be maintained at minimum level. The performance of the proposed algorithm is significantly better than TPSN and LTS.

References

[1] Hanady M. Abdulsalam and Layla K. Kamel. W-leach: Weighted low energy adaptive clustering hierarchy aggregation algorithm for data streams in wireless sensor networks. In *2010 IEEE International Conference on Data Mining Workshops*, pages 1–8. IEEE, 2010.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[3] David W Allan and Marc Abbott Weiss. *Accurate time and frequency transfer during common-view of a GPS satellite*. Electronic Industries Association, 1980.

[4] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[5] Donald E Bently and Charles T Hatch. *Fundamentals of rotating machinery diagnostics*. Amer Society of Mechanical, 2002.

[6] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.

[7] Peter C. Chang, Alison Flatau, and S. C. Liu. Review paper: Health monitoring of civil infrastructure. *Structural Health Monitoring*, 2(3):257–267, 2003.

[8] Stéphanie Damolini. *Carbon Nanotubes and thier Applicatoion to Very Long Bridges*. PhD thesis, Massachusetts Institute of Technology, June 2009.

[9] Jeremy Elson, Lewis Girod, and Deborah Estrin. Fine-grained network time synchronization using reference broadcasts. In *Proceedings of the 5ht Symposium on Operating Systems Design and Implementation (OSDI '02)*, volume 36, pages 11–19. ACM, December 2002.

[10] Jeremy Elson. *Time Synchronization in Wireless Sensor Networks*. PhD thesis, UCLA, 2003.

[11] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: scalable coordination in sensor networks. pages 263–270. ACM, 1999.

[12] Directive EU. A digital agenda for europe, 2010.

[13] MR Falvo, GJ Clary, RM Taylor, V Chi, FP Brooks, S Washburn, and R Superfine. Bending and buckling of carbon nanotubes under large strain. *Nature*, 389(6651):582–584, 1997.

[14] Muhamnmad Omer Farooq, Abdul Basit Dogar, and Ghalib Asadullah Shah. Mr-leach: Multi-hop routing with low energy adaptive clustering hierarchy. In *2010 Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pages 262–268. IEEE, 2010.

[15] Charles R Farrar and Nick AJ Lieven. Damage prognosis: the future of structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851):623–632, 2007.

[16] Suarabh Ganeriwal, Ram Kumar, and Mani B Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pages 138–149. ACM, November 2003.

[17] Yong Gao and Billie F. Spencer, Jr. Structural health monitoring strategies for smart sensor networks. NSEL Report Series NSEL-011, University of Illinois at Urbana-Champaign, Department of Civil and Environmental Engineering, May 2008.

[18] Y Gao and BF Spencer Jr. Online damage diagnosis for civil infrastructure employing a flexibility-based approach. *Smart materials and structures*, 15(1):9, 2006.

[19] Jana Van Greunen and Jan Rabaey. Lightweight time synchronization for sensor networks. In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA '03)*, pages 11–19. ACM, September 2003.

[20] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[21] Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions On Wireless Communications*, 1:660–670, October 2002.

[22] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, volume 2, pages 1–10. IEEE, January 2000.

[23] Sumio Iijima. Helical microtubules of graphitic carbon. *nature*, 354(6348):56–58, 1991.

[24] Ado Jorio, Gene Dresselhaus, and Mildred S Dresselhaus. *Carbon nanotubes: advanced topics in the synthesis, structure, properties and applications*, volume 111. Springer Berlin Heidelberg, 2008.

[25] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for "smart dust". pages 271–278. ACM, 1999.

[26] Byoung-Kug Kim, Sung-Hwa Hong, Kyeong Hur, and Doo-Seop Eom. Energy-efficient and rapid time synchronization for wireless sensor networks. *Consumer Electronics, IEEE Transactions on*, 56(4):2258–2266, November 2010.

[27] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *6th International Symposium on Information Processing in Sensor Networks (IPSN 2007)*, pages 254–263. IEEE.

[28] Sukun Kim. Wireless sensor networks for structural health monitoring. Master's thesis, UC-Berkely, 2005.

[29] V. Krishnamurthy, F. Fowler, and E. Sazonov. The effect of time synchronisation of wireless sensors

on the modal analysis of structures. *Smart Materials and Structures*, 17, August 2008.

[30] Stephanie Lindsey and Cauligi S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. In *IEEE Aerospace Conference Proceedings*, volume 2, pages 1125–1130. IEEE, 2002.

[31] F Li, HM Cheng, S Bai, G Su, and MS Dresselhaus. Tensile strength of single-walled carbon nanotubes directly measured from their macroscopic ropes. *Applied physics letters*, 77(20):3161–3163, 2000.

[32] V. Loscrì, G. Morabito, and S. Marano. A two-levels hierarchy for low-energy adaptive clustering hierarchy (tl-leach). In *IEEE 62nd Vehicular Technology Conference*, pages 1809–1813. IEEE, 2005.

[33] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pages 138–149. ACM, November 2004.

[34] David L Mills. Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, 39:1482–1493, October 1991.

[35] Douglas C Montgomery. *Introduction to statistical quality control*. John Wiley & Sons, 2007.

[36] The Hon. Wayne Swan MP. Intergenerational report: Australia to 2050: Future challenges, australian government treasury report. Technical report, January 2010. [Online].

[37] NICTA. What is castalia ?, Jul 2011 (verified on Jul 10, 2011).

[38] A. Oberlin, M. Endo, and T. Koyama. Filamentous growth of carbon through benzene decomposition. *Journal of Crystal Growth*, 32(3):335–349, 1976.

[39] Kristofer SJ Pister, Joseph M Kahn, Bernhard E Boser, et al. Smart dust: Wireless networks of millimeter-scale sensor nodes. *Highlight Article in*, page 2, 1999.

[40] A. Rao, J. Gubbi, N. Ngo, J. Nguyen, and M. Palaniswami. Energy efficient time synchronization in wsn for critical infrastructure monitoring. In *Trends in Networks and Communications*, volume 197 of *Communications in Computer and Information Sciences*, pages 314–323, 2011.

[41] Kay Römer, Philipp Blum, and Lennart Meier. *Handbook of Sensor Networks: Algorithms and Architectures*, chapter Time Synchronization and Calibration in Wireless Sensor Networks, pages 199–237. Wiley and Sons, October 2005.

[42] Jean-Paul Salvetat, G Andrew D Briggs, Jean-Marc Bonard, Revathi R Bacsa, Andrzej J Kulik, Thomas Stöckli, Nancy A Burnham, and László Forró. Elastic and shear moduli of single-walled carbon nanotube ropes. *Physical Review Letters*, 82(5):944, 1999.

[43] Khurram Shahzad, Arshad Ali, and N.D. Gohar. Etsp: An energy-efficient time synchronization protocol for wireless sensor networks. In *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*, pages 971–976, 2008.

[44] Peter J Shull. *Nondestructive evaluation: theory, techniques, and applications*. CRC press, 2002.

[45] Mihail L Sichitiu and Chanchai Veerarittiphan. Simple, accurate time synchronization for wireless sensor networks. *IEEE Wireless Communications and Networks (WCNC '03)*, 2:1266–1273, 2003.

[46] Hoon Sohn, Charles R Farrar, Francois M Hemez, Devin D Shunk, Daniel W Stinemates, Brett R Nadler, and Jerry J Czarnecki. *A review of structural health monitoring literature: 1996-2001*. Los Alamos National Laboratory Los Alamos, NM, 2004.

[47] BF Spencer, M Ruiz-Sandoval, and Y Gao. Frontiers in structural health monitoring. In *Proceedings of the China-Japan Workshop on Vibration Control and Health Monitoring of Structures and Third Chinese Symposium on Structural Vibration Control, Shanghai*, 2002.

[48] Billie F Spencer, Manuel E Ruiz-Sandoval, and Narito Kurata. Smart sensing technology: opportunities and challenges. *Structural Control and Health Monitoring*, 11(4):349–368, 2004.

[49] Bharath Sundararaman, Ugo Buy, and Ajay D Kshemkayani. Clock synchronization for wireless sensor networks: A survey. *Ad Hoc Networks*, 3:281–323, December 2005.

[50] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. *Vision and challenges for realising the Internet of Things*. EUR-OP, 2010.

[51] Ma Chaw Mon Thein and Thandar Thein. An energy efficient cluster-head selection for wireless sensor networks. In *2010 International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pages 287–291. IEEE, 2010.

[52] MMJ Treacy, TW Ebbesen, and JM Gibson. Exceptionally high young's modulus observed for individual carbon nanotubes. 381:678–680, 1996.

[53] András Varga. What is omnet++?, July 2011 (verified on July 10, 2011).

[54] Y Wang, J. P. Lynch, and K. H. Kae. A wireless structural health monitoring system with multithreaded sensing devices. *Struct. Infrastruct. Eng.*, 3:103–120, 2007.

[55] K. Worden and J. M. Dulieu-Barton. An overview of intelligent fault detection in systems and structures. *Structural Health Monitoring*, 3(1):85–98, 2004.

[56] Fan Xiangning and Song Yulin. Improvement on leach protocol of wireless sensor network. In *2007 Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pages 260–264. IEEE, 2007.

[57] Boris I Yakobson, CJ Brabec, and J Bernholc. Nanomechanics of carbon tubes: instabilities beyond linear response. *Physical review letters*, 76(14):2511, 1996.

[58] Sarah Yang. Researchers create wireless sensor chip the size of glitter. http://berkeley.edu/news/media/releases/2003/06/04_sensor.shtml, June 2003.

[59] Lu Yan, Yan Zhang, Laurence T Yang, and Huansheng Ning. *The Internet of things: from RFID to the next-generation pervasive networked systems*. CRC Press, 2008.

[60] L.-S. Yan, W. Pan, B. Luo, J.-T. Liu, and M.-F. Xu. Communication protocol based on optical low-energy-adaptive-clustering-hierarchy (o-leach) for hybrid optical wireless sensor networks. In *2009 Asia Communications and Photonics Conference and Exhibition (ACP)*, volume 7633, pages 763311–1–763311–6. IEEE, 2009.

[61] Min-Feng Yu, Oleg Lourie, Mark J Dyer, Katerina Moloni, Thomas F Kelly, and Rodney S Ruoff. Strength and breaking mechanism of multiwalled carbon nanotubes under tensile load. *Science*, 287(5453):637–640, 2000.

[62] Ip version 6 addressing architecture. https://tools.ietf.org/html/rfc4291, 2006. [Online].

[63] Connections counter: The internet of everything in motion. http://newsroom.cisco.com/feature-content?type=webcontent articleId=1208342. [Online].

[64] Internet world stats : Usage and population statistics. http://www.internetworldstats.com/emarketing.htm. [Online].

[65] Libelium. http://www.libelium.com/. [Online].

[66] Xively. https://xively.com/. [Online].