A Synergy of the Wireless Sensor Network and the Data Center System

Ke Hong Shuo Yang Zhiqiang Ma Lin Gu
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Email: khongaa@cse.ust.hk, syangag@ust.hk, {zma, lingu}@cse.ust.hk

Abstract—**In recent years, data centers have emerged to be an increasingly important computing infrastructure. It is shown that wireless sensor networks (sensornets) can provide fine-grained measurements in data centers, and achieve better control of the data center platform for energy efficiency. However, the usage of sensornets has so far been limited to auxiliary functions, such as sensory data collection across a data center. We argue that the combined computational and networking capability of a sensor network enables it to interact with the clusters in a much more sophisticated way and enhance essential functions in a data center. We have designed a Cluster-Area Sensor Network (CASN) to improve the cluster management and operational security in the system. Implemented with TelosB motes, CASN can be easily deployed in a cluster, with sensor nodes attached in an ad hoc manner to servers, and provides key system functions including cluster-wide command dissemination and verification of physical presence. Experimental results show that CASN has 85% success rate in verifying physical locations of servers with coarse-grained localization when the threshold is 3 meters, and incurs small latency in cluster-wide command dissemination.**

I. Introduction

Data centers have become increasingly important in today's computing technology, and a number of sensor networks have been developed for this critical infrastructure [1], [2], [3]. The cost of a sensor network is negligible compared to rack servers, high-speed network equipment, power distribution, and cooling facility in a data center, and it can provide useful services such as temperature measurements and emergency event detection. However, most sensor networks used in data centers operate as an auxiliary network for sensory data collection. It has not been shown whether the sensornet can actively participate in core computing functions in the data center and enhance the management of large clusters.

We notice that wireless sensor networks are, in fact, sophisticated in many technical aspects—they self-organize into an ad hoc network without *a priori* configuration, adapt to topology changes and node failures, and may localize their network nodes with minimal infrastructural support. Many functions can complement those currently running in data center systems, and improve the manageability and security of the compute clusters. We also observe several inefficacies in the current data center management technology, and believe that the sensornet technology opens a new design space for addressing these problems.

One crucial task in a data center is the management of software on a compute clusters—to modify the software state, e.g., system settings, OS components, application software and configuration files, on a large number of servers in the cluster. Typically, such tasks are performed on a *control station* or *management station* [4], and often require certain manual operations, such as specifying IP addresses and installing a basic system image. A sensor network, in contrast, emphasizes self-organization, and can use wireless reprogramming to update crucial system software. By bridging the sensornet-based wireless reprogramming mechanism to controlled server reprogramming, we can enhance the flexibility and security of disseminating control commands to a plurality of servers and, consequently, change their software state more efficiently.

Moreover, security of the data center platform has been a serious concern today. Data centers have attracted all sorts of attacks, and even leading Internet systems fall victim of hacking activities resulting in leakage of user data. In 2012, Yahoo! confirmed the exposure of credentials of 450,000 personal e-mail accounts due to a SQL injection attack [5]. Sony PlayStation Network and Qriocity services have also reported that 77 million customers' account information was exposed by hackers [6]. In addition to traditional authentication, it is necessary to deploy additional measures to effectively monitor the system and detect anomalies.

Using its capability of measuring physical properties including the wireless signal strength, the sensor network can provide additional security enhancement to the cluster system by capturing and verifying physical signatures of communicating entities. In particular, the sensor network can use radio-based localization to summarize the physical wireless signal strengths into a signature representing a communicating entity's approximate location, and verify that it is close to the entity's real location known to the system. An anomaly incident is to be reported to the system administrator if the verification fails. Although sensor networks have limits on the precision of localization, approximate location information can already serve our purpose and defend many exploits that forge a digital presence of a legitimate physical entity, such as IP spoofing. Intrinsically, the verification of such unforgeable physical properties mitigates the difficulty of establishing a correspondence between a computing object and a real-world entity, which is the root cause of unauthorized data access [7].

We develop a wireless sensor network called CASN (Cluster-Area Sensor Network) as an integral part of the data center computing environment, and design a few mechanisms for the servers to closely work with the sensornet and improve the functionality of the overall system. CASN provides two main functions: cluster-wide command dissemination and verification of physical presence. When implementing the verification of physical presence, we find that existing localization schemes have limitations in the indoor environment. To improve the precision and stability of indoor localization, we develop a new empirical RSSI (Received Signal Strength Indicator) based model for localization. The model takes the multipath effect into consideration and improves the localization accuracy in our empirical study.

It is worth clarifying that the verification of physical presence does not aim to replace existing cryptographic authentication mechanisms. Instead, by determining the approximate locations of the servers, CASN supplements existing authentication mechanisms by providing a probabilistic means to verify an unforgeable physical property, and performs anomaly detection in the system.

To the best of our knowledge, CASN is the first attempt to make the sensor network an "internal" system component that provides core system functions and works closely with other components in a cluster computing system. This design follows an observation on several interesting similarities between sensornet and data center systems, and also distinct features of these two systems that complement each other. Section IV provides more details on the rationales of our approach, application notes on the command dissemination and physical signatures, and discussions on power consumption and fault tolerance. Our work contributes to the sensornet technology and data center based computing in the following aspects.

- We have implemented a sensor network based control command distribution system for clusters.
- We have developed a coarse-grained localization system based on a new radio signal strength model for indoor environments, and evaluated its performance.
- We have created a verification mechanism based on physical presence, currently focusing on the physical location, this method can potentially be extended to other unforgeable physical properties.

The rest of the paper is organized as follows. Section II describes the design of CASN's reprogramming and authentication function. Section III presents the evaluation results of CASN's authenticated reprogramming. Section IV has additional discussions on the rationales and usage of CASN, as well as addresses concerns on power consumption and reliability. We discuss related work in Section V. Section VI summarizes this work.

II. Design

We design CASN to be an inexpensive, independent and integrated wireless sensor network that provides command dissemination through the wireless channel and verification of physical presence. CASN is designed to complement and strengthen, not replace, the hardware and software components in the data center computing environment. In the meantime, the sensor network provides functionality that traditional cluster systems cannot implement in a cost-efficient way.

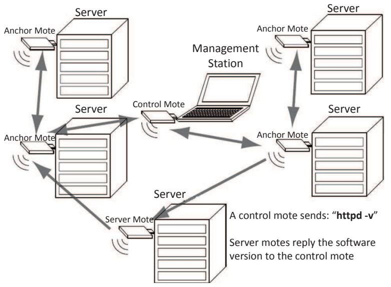


Fig. 1: Dissemination of control commands

A. System architecture

CASN comprises a number of sensor nodes attached to the compute servers or workstations. In a data center facility, a compute server is usually mounted in a server rack, and is often called a node in the system. To simplify naming, we use "mote" to refer to a sensor node, and "server" to refer to a compute server node. A small subset of servers and workstations are designated as *management stations*, and we require that only the management stations can issue critical control commands. Such specially designated management stations are used in many data centers, and should be easy to implement. Some other servers on critical data access paths, such as mail application servers, also perform sensitive data access operations and thus need to be verified for their identity. The motes attached to the management stations and servers on critical data access paths are called *control motes*. We assume the servers' locations are known, and this can be implemented either with a system-wide database or local records on individual servers. In the latter case, the servers need to declare their location when communicating with other servers. Some motes are designated as *anchor motes* and perform the function of anchors in localization algorithms. CASN should have four or more anchor motes for it to perform localization. One particular anchor mote is responsible for collecting localization data and conducting triangulation. We called this important anchor mote as the *anchor master*. If a mote is neither a control mote nor an anchor mote, we call it a *server mote*. Fig. 1 illustrates the deployment of the CASN system in a data center.

B. Sensornet-assisted command dissemination

CASN is built on existing wireless reprogramming capability in sensor networks. The data to be disseminated, however, are smaller than typical executable images, which makes the workload easier to handle in the sensornet and distribution latency shorter.

When a control station starts to send commands to the cluster, the control mote attached to the control station broadcasts a `CONTROL_COMMAND` message with its own ID in the wireless channel. Upon receiving `CONTROL_COMMAND`, the anchor motes retrieve the ID of the control mote from the message, and may conduct additional authentication for the control mote (refer to Section II-C). When it receives the CONTROL_COMMAND message, a server mote also retrieves the ID of the control mote, and queries an anchor mote with a SERVER_QUERY message containing the ID of the control mote to verify the validity of the command. The anchor mote confirms with an ANCHOR_RESPONSE message, contingent on the authentication result, if any, and all motes forward the command to servers they are attached to after receiving a positive ANCHOR_RESPONSE.

CASN provides a command-line interface for administrators to perform cluster control tasks. Servers are identified by the TOS_NODE_IDs of the server motes or anchor motes attached to them. Administrators can send a command to servers via the command-line interface, and CASN distributes the command to the motes. If authentication is successful, the motes deliver the commands to the servers to be executed. Although CASN currently disseminates only control commands for compute servers to be executed, given sufficient wireless bandwidth, the system can easily extend to also send small program files, configurations, and data over the sensor network to program compute servers.

C. *Verification of physical presence*

To obtain and verify the current locations of servers, CASN conducts localization periodically for actively communicating servers. The periodic localization process is launched by the anchor master sending out ANCHOR_QUERY messages to verify the physical presence of control motes. When a nearby control mote, which is attached to a management station or a server on a critical data access path, receives a ANCHOR_QUERY message, it replies with 10 CONTROL_DISCOVER message containing the ID of the control mote with 1-second intervals. The CONTROL_DISCOVER message is sent multiple times because anchor motes need to measure the RSSI of multiple CONTROL_DISCOVER messages as the physical properties. The mean and variance of the measurements are sent to the anchor master together with the IDs of the anchor and control motes. The anchor master then calculates the position of the control motes based on the collected data. If the localization result shows that a control mote is in the vicinity of a designated position, the anchor master adds the control mote to a *trusted mote list*, a list of currently verified control motes which is replicated to anchor motes. Membership in the trusted mote list is a soft state and needs to be re-authenticated periodically. This allows new control motes to be verified and stale information removed.

The additional verification of physical presence happens when the control mote sends commands to a server mote. With help from the anchor motes, the server mote looks up the control mote in the trusted mote list. The outcome of the verification is sent to the server mote with ANCHOR_RESPONSE messages. A summary of the communication process and messages involved is shown in Fig. 2. Note that the authentication procedure can be triggered not only by the ANCHOR_QUERY message sent by the anchor mote passively but also by a CONTROL_DISCOVER message sent by the control mote pro-actively. The control mote broadcasts a CONTROL_DISCOVER message after it starts. If an anchor mote receives this CONTROL_DISCOVER message, it replies with an ANCHOR_QUERY message, and the protocol proceeds as presented in Section II-B. This on-demand verification mechanism serves two purposes. First it allows a control mote currently not in the trusted mote list to be verified when communication starts; Second, it prevents a compromised server from impersonating a verified server whose control mote is in the trusted mote list.

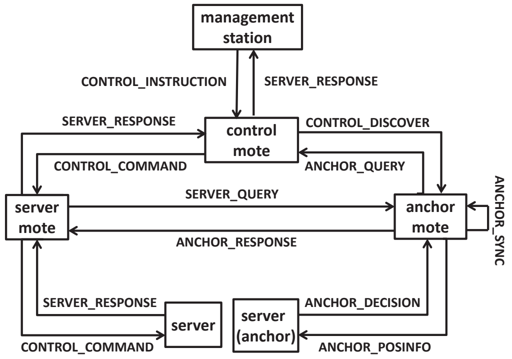


Fig. 2: Localization and verification of physical presence

The verification process relies on the correct operations of the sensornet itself, i.e., CASN itself must not be compromised. The characteristics of sensor networks make attacks to the sensor network inside a data center very difficult. First, the sensor network uses a specific wireless channel for communication, and is practically “detached” from the Internet. Second, the sensor node hardware usually supports disabling writing to its program memory. This makes it impossible to modify the sensor node’s behavior even if the adversary compromises the server to which the sensor node is attached. Finally, the data center facility is usually well protected for physical accesses. Hence, it is difficult for an external adversary to enter the data center and manipulate the sensor network.

D. *Coarse-grained localization*

Precise sub-meter localization is not necessary in the control mote verification. Suffices it for a control mote to be located in the vicinity of its designated position to pass the verification. Considering typical dimensions of racks and the data center, we estimate that a localization precision of 5 meters is still useful for CASN to construct physical signatures.

It may appear that simple ranging or even a wireless reachability test can achieve the physical authentication. In fact, the relatively large errors in RSSI-based ranging and the irregularity of wireless communication make the fidelity of these schemes too low to be useful [8]. Correlating a large number of measurements on numerous participating nodes, the localization process reduces the error, and makes the authentication more reliable. It is, however, still non-trivial to construct a viable localization system in real-world settings, especially in an indoor environment.

1) Signal strength model: RSSI-based ranging is challenging in an indoor environment. Fig. 3 compares the RSSI measurements in two indoor environments. Though the two experimental environments are next to each other, we can clearly observe the irregularities of the signal strength. First, in different environments, the RSSI measurements at the same reference distance are not the same. Second, at certain distances, instead of decreasing monotonically, the RSSI can increase. For example, in a corridor, P(6m) = -77.5dBm and P(9m) = -73dBm.

Such irregularity is also reported in earlier works on localization, and we find it difficult to leverage an existing localization model to provide reliable performance in the indoor environment. Fortunately, CASN requires only approximate localization, and can tolerate certain inaccuracy in trade of improved stability. This allows us to construct an empirical model to characterize the RSSI in an indoor environment where the multipath effect is prevalent. Though the model is an approximation to the real radio propagation phenomenon, we empirically verify it in realistic settings to make sure this approximation serves its purpose in this problem domain.

The new empirical model includes the contributions of indirect signals in radio propagation, and characterizes the received signal strength as

$$P(d) = P(d_0) - 10n \log\left(\frac{\sum_{i=1}^{k} r_i d_i}{d_0}\right) \tag{1}$$

In Eq. (1), d is the discretized distance between the transmitter and the receiver, $P(d)$ is the received signal strength, and $P(d_0)$ is the received signal strength at a reference distance. n is the loss exponent determined by the environment, normally between 2 and 4 ($n = 2$ in free space propagation). r_is are the amplitude coefficients of signal components, d_is are the corresponding distances. $\sum_{i=1}^{k} r_i d_i$ can also be viewed as the dot product of two vectors $\mathbf{R} = \begin{bmatrix} r_1 \ r_2 \ \cdots \ r_k \end{bmatrix}^{\mathbf{T}}$ and $\mathbf{D} = \begin{bmatrix} d_1 \ d_2 \ \cdots \ d_k \end{bmatrix}^{\mathbf{T}}$. $\mathbf{R} \cdot \mathbf{D}$ represents the composition process of the signals. k is an integer and, to simplify computation, we place a practical limit $1 \leq k \leq 11$.

Furthermore, we use Rician Distribution to model the amplitude drop of the indirect signals [9]. The probability density function of Rician Distribution is

$$R(x|\nu, \sigma) = \frac{x}{\sigma} e^{\frac{-(x^2+\nu^2)}{2\sigma^2}} I_0\left(\frac{x\nu}{\sigma^2}\right) \tag{2}$$

where $I_0(z)$ is the modified Bessel function of the first kind with order zero. The parameters ν and σ describe the relationship between the power of the direct signal and the indirect signal. Their values can be determined empirically. Suppose the discretized distance between the transmitter A and the receiver B is d_{AB}, using the Rician distribution, we define amplitude coefficient r_i as

$$r_i = \begin{cases} 0 & if\ d_i < d_{AB} \\ 1 & if\ d_i = d_{AB} \\ a_i \cdot R(d_i - d_{AB}) & if\ d_i > d_{AB} \end{cases} \tag{3}$$

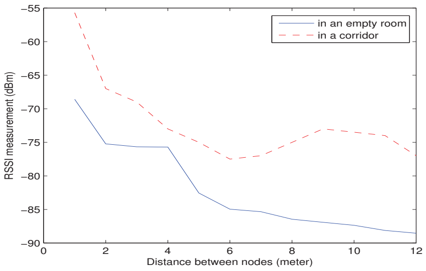


Fig. 3: RSSI measurements in an empty room and a corridor

2) Probabilistic ranging: In Eq. (3), $R(d_i - d_{AB})$ characterizes the attenuation of the signal when reflected signal travels longer distance than d_{AB}. a_i denotes the number of reflected signals that travel a certain distance, which can be specified as a fixed value in practice. We call reflected signals that travel the same distance as one type of reflected signals. Rounding the distances to integer values, we let $\mathbf{D} = \begin{bmatrix} 1 \ 2 \ \cdots \ 11 \end{bmatrix}^{\mathbf{T}}$. Then $\mathbf{R}$ would be, for example, if the physical distance between the transmitter and the receiver is 3m and $a_i = 1$, in the form of $\mathbf{R} = \begin{bmatrix} 0 \ 0 \ 1 \ \cdots \ R(8) \end{bmatrix}^{\mathbf{T}}$. From Eq. (1), we know

$$\mathbf{R} \cdot \mathbf{D} = d_0 \cdot 10^{\frac{P(d_0) - P(d)}{10n}} \tag{4}$$

in which $\mathbf{D}$, $P(d_0)$ and $P(d)$ (the RSSI measurement) are known. We can then compute $\mathbf{R}$ so that it satisfies Eq. (4), where $r_m = 1$ and $r_j = 0$ for all $j < m$, d_m is the physical distance. In real applications, d_{AB} is, however, unknown, and it is impossible to determine the number a_i of each type of reflected signals. Hence, given an RSSI measurement, we can compute multiple **R**s satisfying Eq. (4). To reduce the computational complexity, we further simplify the problem so that only the 5 shortest reflected signals are considered and that a reflected signal that travels 2 meters longer than the direct signal can be ignored. Therefore, Eq. (3) can be simplified to be

$$r_i = \begin{cases} 0 & if\ d_i < d_{AB}\ or\ d_i - d_{AB} \geq 2 \\ 1 & if\ d_i = d_{AB} \\ a_i \cdot R(d_i - d_{AB}) & if\ d_i > d_{AB}\ and\ d_i - d_{AB} < 2 \end{cases} \tag{5}$$

where $a_i \leq 5$. With appropriate configuration, Eq. (4) and (5) can map an RSSI measurement to several **R**s. Compared to models used in some systems where one RSSI measurement can only generate one distance value, the model used in the CASN gives an uncertain answer, but it reflects the multipath reality and gives the system more choices to determine the distance based on the RSSI measurement so that the result can be close to the physical reality.

3) Localization: After obtaining the probabilistic ranging results, the localization algorithm needs to compute the most plausible location information from the expanded ranging results. Suppose a system with H anchor motes. For each control mote, H RSSI measurements can be obtained for each transmission, and, if each RSSI measurement value maps to 10 **R**s, we would have 10^H possible combinations of estimates based on which we can derive the approximate location of

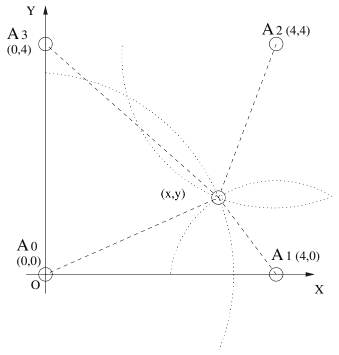


Fig. 4: Geometry constraints & trilateration positioning

the control mote. It is, however, computationally too costly to calculate all these possibilities. Therefore, we apply additional heuristics to reduce the number of possible distance estimates.

One way of narrowing down the distance values is applying geometric constraints. Not all distance values satisfy the triangle inequality. In this way, calculations of some invalid combinations can be avoided. Another way is utilizing the known distance between anchors. We can calculate **R**s for an RSSI measurement between a pair of anchor motes. One of these **R**s should yields the true distance. By analyzing this **R**, the multipath pattern around this pair of motes can be roughly characterized. For example, if **R** has one r_i with $a_i = 5$, it is reasonable to conjecture that other signals in the nearby area have similar compositions, and discard results that have too few reflected signal components.

Applying these techniques, the position of a target node (a control mote) can be calculated using trilateration. Three distance measurements can generate one position calculation in a 2-dimensional space. The position will be calculated four times using different combinations of anchor mote triads, then the centroid of the calculated locations will be taken as the location of the control mote.

We illustrate the localization process using a simple example based on measurements in a field test. With a control mote placed at the position (2,2) in a configuration shown in Fig. 4. A_0, A_1, A_2 and A_3 represent the anchor motes at (0,0), (4,0), (4,4) and (0,4), respectively. The RSSI measurements collected by anchor motes are shown below:

RSSI[dBm]	A_1	A_3	A_0	C	A_2
A_1	0	-62.00	-67.40	-56.00	-62.30
A_3	-69.10	0	-57.60	-58.00	-67.60
A_0	-80.10	-61.80	0	-55.90	-83.80
C	-62.30	-55.00	-58.10	0	-62.00
A_2	-71.75	-68.25	-80.25	-60.50	0

In the table, **C** is the control mote whose position is to be determined. CASN generates an interval for each RSSI measurement. For instance, the RSSI measurement between A_0 and **C** is -57dBm, which will be mapped to an interval [1.0, 2.2]. If we choose a step value of 0.4 to discretize this interval, 4 distance values, namely, 1.0, 1.4, 1.8, and 2.2, will be obtained. Similarly, other RSSI measurements generate more distance estimates. The final result of the average position of **C** is (1.6, 2.2).

III. Implementation and evaluation

We have implemented a prototype of CASN and deployed it in a research cluster with typical hardware used in data centers. Experimental results indicate that CASN is able to achieve efficient and reliable authenticated reprogramming. We use 6 motes to conduct experiments on reprogramming and verification of physical presence. One control mote provides control interface to the administrator, and one server mote receives commands from the control mote and performs corresponding operations. Four anchor motes gather RSSI strength values of signals sent from other motes. The RSSI strength is obtained from the CC2420 chip directly. The underlying routing protocol for the sensor network is CTP [10].

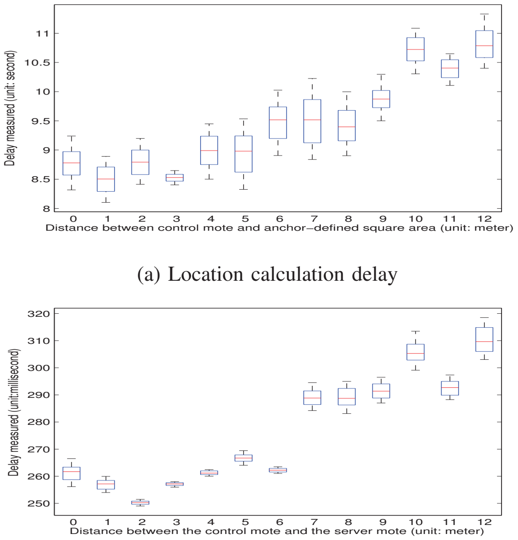
The wireless sensor mote platform used in CASN is TelosB [11]. The **nesC** programs running on sensor motes are developed using TinyOS version 2.1.1. To facilitate the management process, we also implement a user interface for administrator to perform tasks in a command line style. With a sensor mote attached to the administration work station, the administrator can launch the control program. In the control command line, administrator can input "send" followed by a system command that is recognizable by server machines to broadcast the command to all the server machines within range. Parameters can be added to send commands only to a specific server machine. The sever machine is identified by the `TOS_NODE_ID` of the sensor node attached to it.

A. System latency

Providing command dissemination and location verification operations, CASN introduces the following two types of delays: the *location calculation delay* from the time when the anchor mote receives the first `CONTROL_DISCOVER` to the time when the anchor mote receives `ANCHOR_DECISION` (refer to Fig. 2) about specific control mote and the *control delay* from the time when the control mote sends out a `CONTROL_INSTRUCTION` message to the time when the control mote receives the `SERVER_RESPONSE` message.

Fig. 5 shows the location calculation delay and control delay. Each data point centers on the mean of 100 repeated measurements. It takes 8–12 seconds to calculate the location of the control mote, as shown in Fig. 5(a). Since anchors must gather multiple RSSI measurements for calculation and the control mote sends `CONTROL_DISCOVER` once per second, position calculation usually takes a few seconds. Although position calculation itself happens every 30 seconds, if the control mote stays at a static position, the localization overhead is incurred only once. The longest delay we obtained during the experiment is around 12 seconds, which is still acceptable for cluster-wide operations.

The control delay against the distance between the control mote and the server mote is shown in Fig. 5(b). We can see that the administrator would only wait for less than 300ms if the control mote keeps close to the anchor motes and server motes. When the distance increase, the delay is also increased due to routing overhead. Overall, the control delay is low enough to make cluster-wide command dissemination very effective.



(a) Location calculation delay

(b) Control delay between the control and server mote

Fig. 5: Latency of authenticated reprogramming

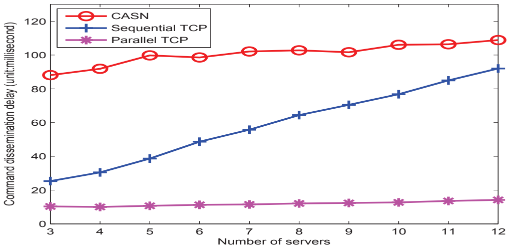


Fig. 6: Latency of unauthenticated reprogramming

To further understand the wireless scalability of command dissemination in CASN, we vary the number of compute servers to be reprogrammed on three racks in our testbed. A control mote is attached to a workstation located in one of the three racks. An *echo* command is issued by the workstation using unauthenticated wireless broadcast in CASN or wired TCP (in a sequential or parallel manner) to a number of compute servers residing in the three server racks. Fig. 6 shows the command dissemination delay, calculated from the time when the workstation issues the command to the time the *echo* response from all compute servers are collected, versus the number of compute servers to be reprogrammed. We see that the wireless broadcasting approach leveraged by CASN is scalable with comparable latency against wired TCP.

B. Localization precision

We conduct the localization experiment on our testbed and emulate the scenario shown in Fig. 4, where anchor motes are deployed at positions A_0, A_1, A_2 and A_3, to show that the localization algorithm is effective in correctly authenticating the physical presence of control motes within small error bound. A control mote is placed at different positions inside the square bounded by the anchor motes to obtain the localization results. Table I compares the real and estimated position of the control mote for each location, and computes the localization error **e**, defined as the Euclidean distance between the calculated position (x', y') and the real position (x, y) of a control mote. The localization error for each location is also illustrated in Fig. 7. Fig. 8 shows the localization correctness rate given a set of thresholds for the verification of physical presence. In other words, a localization result leads to correct verification outcome if its localization error is less than the threshold.

x	y	x'	y'	error (meter)	x	y	x'	y'	error (meter)
0	0	0	0	0	3	0	3.2	2.1	2.1095
0	1	1.1	2	1.4866	3	1	2.8	0.8	0.2828
0	2	-1.4	0.8	1.8439	3	2	3.5	3.7	1.7720
0	3	-1.9	0.8	2.9069	3	3	2.9	4.2	1.2042
0	4	0.1	-2.3	6.3008	3	4	4.7	4.3	1.7263
1	0	1.6	2	2.0881	4	0	5.2	-0.7	1.3892
1	1	2.3	2.3	1.8385	4	1	3.8	-1.1	2.1095
1	2	2.2	0.7	1.7692	4	2	4.6	-1.4	3.4525
1	3	2.8	2.9	1.8028	4	3	-2.8	3.5	6.8184
1	4	-1.1	4.3	2.1213	4	4	4.7	-0.4	4.4553
2	0	2.5	0.3	0.5831	2	3	2.7	3	0.7
2	1	3.1	1.2	1.1180	2	4	-0.9	4.7	2.9833
2	2	2.3	1.6	0.5					

TABLE I: Control mote's real vs. localized position

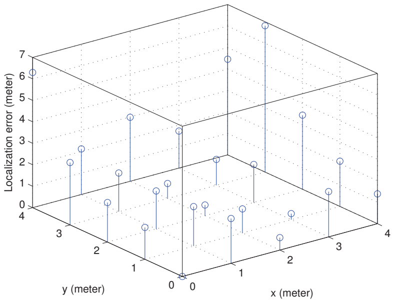


Fig. 7: Localization error

According to Fig. 7 and Fig. 8, 60% of the localization errors are lower than 2 meters. At the boundary of the square, the localization errors become larger, with the localization errors bounded by 7 meters. A partial cause lies in the stronger signal attenuation at some anchor motes. Overall, 88% of the localization errors stay within 5 meters, which meets our design goal. With more anchor motes, the precision can be further improved.

IV. Discussion

Involving a sensor network in core functions of cluster-based computing is a dramatic change in how we design and use a sensor network in a data center environment. In

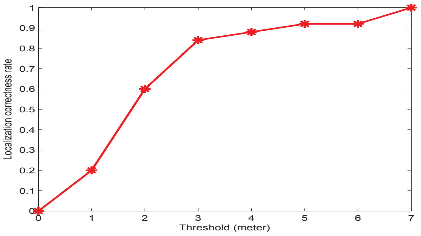


Fig. 8: Localization correctness rate with different thresholds

this section, we first analyze why sensor network and data center systems, in fact, share a number of similarities, then discuss how to use command dissemination and physical signatures in a data center, and finally address concerns on power consumption and reliability of CASN.

A. Sensornet and data centers

One of the typical work patterns of sensor networks is a large number of sensor nodes operating in an unattended manner for an extended period of time [12], [13], [14], [15]. This work mode is interestingly similar to how a cluster of compute servers in a data center works. It has been demonstrated that hundreds of sensor nodes can collaborate in one application with demanding time constraints [16]. A recent system, GreenOrbs, scales to 1000+-node deployments and operates for a long period of time [17]. Such a scale is capable of fully supporting a cluster of servers in a data center. Typically, a cluster comprises 100–5000 compute servers, and a data center may accommodate multiple clusters, each serving a specific purpose (e.g., indexing [18], MapReduce computation [19], or database storage [20]) or a combination of some services.

Built on inexpensive hardware and deployed in large numbers, sensornets must expect faults in the system and handle them automatically. The capability of adapting to faults and system dynamics is explicitly required or implicitly built in almost all layers of the system architecture, from communication [10], sensing [16], data management [21], to application logic [12], [22]. The same principle is applied in the data center environments, where faults are “norm” but the system must be always available. Dean reported that servers in Google’s data centers typically fail twice every year and, in addition, 1-5% hard drives fail yearly [23]. This means a dozen servers and disks fail every day in a cluster of several thousand machines. Hence, the software for data center computing must handle the faults.

Designed to work in demanding outdoor environments, the sensor nodes can work, in fact, reliably in the indoor data center environment. Given that the sensor node’s fault rate is similar to that of other components in the server, the faults from the sensor system will not compound the reliability issue to the extent that the fault-tolerant data center software is not able to handle. It is also very easy to replace a faulty sensor node in a cluster.

The similarities between the sensor network and data center systems indicate a compatibility of work mode that enables a sensor network to work together with a cluster. Furthermore, the sensornet technology provides unique strengths—self-organizing networking, wireless reprogramming and the ability to capture physical properties, which can improve the operations of a data center system. Built on these observations, CASN provides two main functions: dissemination of control commands and verification of physical presence.

B. Dissemination of control commands

Large distributed computer systems, such as data centers, routinely require management, software upgrade, and reconfiguration. Typically, an authorized user (or program) logs in to certain computers (management stations) and performs the management task (e.g., restarting servers, distributing new software) [4]. Usually, the system needs to install specific software “agents” to fulfill such tasks on many servers, and certain tasks, such as the configuration of static IP addresses, cannot be performed over the network and require manual operations.

In the sensornet technology, researchers have developed a number of solutions for wirelessly programming a sensor network, providing intelligent code dissemination, fault-tolerant operations, and network-wide operations [24], [25]. Reprogramming software, such as Deluge, has become a standard programming method in sensornet systems and are widely used [24]. In a cluster environment, administrators can send control commands to servers over the sensor network in a similar way to wireless reprogramming, as long as we build a wireless sensor network closely integrated with the servers.

C. Verification of physical presence

The sensor network provides the capability of capturing physical properties of the environment and conducting computation based on the perceived reality. CASN uses the wireless signal strength as an unforgeable physical property of the communicating servers, and supplements the existing authentication system by verifying the physical presence of certain servers. Since localization in sensor networks has been intensively studied, an integrated sensor network in the cluster shall be able to summarize the radio signal strength to obtain location information for the physical verification. Fingerprint-based localization in sensor network [26] may be an alternative approach for the physical authentication in CASN. However, it normally requires a more or less stable environment so that the radio map database obtained offline for referencing can be in long-term use. The range-based localization algorithm used in CASN is free of such restrictions.

The verification of physical presence is able to tackle some potentially hazardous attack scenarios. We give a few examples below.

- *Verification of management stations:* Usually, critical management tasks are initiated from a set of management stations that are protected with strong authentication and access provisioning procedures [4]. In CASN, physical presence of the management station can be localized to verify that control commands are indeed coming from where the management station locates. This enables the system to detect potential exploits of machines impersonating the management stations.
- *Detection of malicious attacks:* In public clouds, virtual machine instances make attacks possible from inside a data center. For example, it is reported that a Zeus botnet controller resided on and launched attacks from the Amazon EC2 platform [27]. A malicious virtual

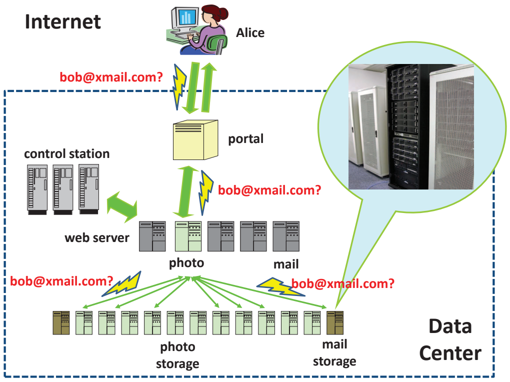


Fig. 9: Organization of data center based computation

instance inside a data center can generate IP-spoofed traffic in a shared network environment, launch a DDoS attack against other virtual instances and disrupt legitimate services. If we can verify the physical location of an IP address, such attacks can be easily detected. In another attack pattern, adversaries may create malicious services or virtual machine instances, and impersonate valid service providers [28]. With the verification of the physical location of virtual instances, such malicious virtual instances will inevitably leave some footprints in the data center, facilitating administrators to locate and confine the physical source of attacks.

- *Verifying the access path:* For security, data center systems often require data accesses follow certain paths [29]. For example, in Fig. 9, only the photo application server can access the photo storage, and only the mail application server can access the mail storage. Assuming the servers' locations are known, the verification of physical presence can probabilistically detect accesses originating from illegitimate servers, supplementing the conventional anomaly detection mechanism.

Currently, CASN conducts physical authentication of server access within one single data center of reasonable scale. By propagating physical authentication results out of one data center through the wide-area network, CASN may verify legitimate access among geo-distributed data centers. We leave this for future study.

D. Power consumption and fault tolerance

The power consumption of sensor motes is not an issue for CASN because the power of sensor motes are provided by the server machines through USB connectors. Furthermore, we have studied the effect of communication between the server and mote through the USB interface. Since we limit the period of RSSI sampling to be one second, this overhead is negligible to the compute servers.

Sensor nodes are supposed to work in outdoor environments, compared with which the data center environment is less stringent. Fault tolerance in wireless sensor networks has been intensively studied and a number of techniques have been proposed for fault detection and recovery [30], [31]. Even though a sensor mote itself might not be highly reliable, it is inexpensive and convenient to replace a faulty mote by a new one. If one mote in CASN becomes malfunctioning, it can be easily replaced or reprogrammed. A faulty anchor master can be replaced by software using a dynamic leader election protocol [32] or Paxos-based consensus [33], and this is one direction of future work for CASN.

V. Related Work

In the emerging cloud-based computing paradigm, data centers provide the fundamental functions of storing data for millions of users and exercising computation at an unprecedentedly large scale. Most commodity data center systems are similar in organization to the Google clusters [18]. Individual parts of the system, including the file system [34], execution engine [19], and energy-conscious organization [35] have been studied in great details.

Sensor networks represent another platform where the decentralized operations and dynamic network activities require a re-examination of the traditional network design. Although sensor nodes are limited in power, computing capacities and memory, their sensory capability, self-organizing behavior and fault-tolerant system design enable them to conduct computation in the physical world in a way not previously feasible. In recent years, sensor networks start to serve as a monitoring facility in data centers, and help achieve better environment control and energy efficiency. A sensornet system for data center temperature monitoring is presented in [3]. In [36], a near-optimal sensor placement scheme is proposed to detect hot servers in a data center. ThermoCast [2] is a cyber-physical system using sensors to collect air flow information in a data center for server overheating prediction. Furthermore, a robot is built to navigate in a data center and to collect temperature, humidity data for asset tracking and energy management [1]. Sensor networks can easily replace parts of the data center sensory infrastructure to collect temperature, humidity, and other physical measurements in the ambient environment. Hence, traditional use of sensor networks in data centers focuses on sensory data collection in the data center building and clusters. This is, however, different from our approach in terms of the purpose, methodology, and the extent to which the sensor network participates in the operations of the data center system.

In fact, the research and development on sensor networks have generated several pioneering technologies which data center systems can benefit from. Related to this work, wireless reprogramming is a basic operation in sensor networks. A standard reprogramming tools, Deluge, can disseminate large binary objects to many loosely-coupled nodes [24]. More advanced systems can further improve the efficiency of bandwidth usage [37], [25]. Radio-based localization techniques for sensor networks is also an active area of development. The basic physical and geometric models have been well studied [38], and many ranging based triangulation methods are proposed. While simple RSSI based ranging techniques have limited precision [39], advanced systems can achieve an average localization error as small as 3cm at a distance of 160 meters [40]. In our work, we still use the RSSI based method with a new signal strength model because the application requires only coarse-grained position information and the multipath effect is significant in the indoor environment.

VI. Conclusion

CASN enables the data center system to manage a large number of servers through an intelligent, independent, albeit low-bandwidth, sensor network. Based on a coarse-grained localization algorithm, CASN verifies the physical presence of servers to be legitimate entities within the data center, and enhances the security of datacenter management.

References

[1] K. Deland, J. Lenchner, J. Nelson, J. Connell, J. Thoensen, and J. O. Kephart, "Demo: A robot-in-residence for data center thermal monitoring and energy efficiency management," in *Proc. of the 9th ACM Conf. on Embedded Networked Sensor Systems (SenSys '11)*, pp. 381–382, 2011.

[2] L. Li, C.-J. M. Liang, J. Liu, S. Nath, A. Terzis, and C. Faloutsos, "Thermocast: a cyber-physical forecasting model for datacenters," in *Proc. of the 17th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD '11)*, pp. 1370–1378, 2011.

[3] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao, "RACnet: a High-Fidelity Data Center Sensing Network," in *Proc. of the 7th ACM Conf. on Embedded Networked Sensor Systems (Sensys '09)*, 2009.

[4] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance*. 2009.

[5] "Yahoo confirms hacker exposure of 450,000 e-mail passwords," *http://redmondmag.com/articles/2012/07/12/yahoo-confirms-hacker-exposure-of-450000-email-passwords.aspx*. [last access: 07/21, 2012].

[6] "Playstation network users fear identity theft after major data leak," *http://www.guardian.co.uk/technology/2011/apr/27/playstation-users-identity-theft-data-leak*. [last access: 07/22, 2012].

[7] D. D. Clark and D. R. Wilson, "A comparison of commercial and military computer security policies," in *IEEE Symp. on Security and Privacy'87*, pp. 184–195, 1987.

[8] G. Zhou, T. He, and J. A. Stankovic, "Impact of Radio Irregularity on Wireless Sensor Networks," June 2004.

[9] S. O. Rice, "Mathematical analysis of random noise," *Bell Systems Technical Journal*, vol. 23, 1944.

[10] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. of the 7th ACM Conf. on Embedded Networked Sensor Systems (SenSys '09)*, pp. 1–14, 2009.

[11] J. Polastre, R. Szewczyk, and D. Culler, "Telos: enabling ultra-low power wireless research," in *Proc. of the 4th Intl. symposium on Information processing in sensor networks (IPSN '05)*, 2005.

[12] C. Hartung, R. Han, C. Seielstad, and S. Holbrook, "Firewxnet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments," in *Proc. of the 4th Intl. Conf. on Mobile Systems, Applications and Services (MobiSys '06)*, pp. 28–41, 2006.

[13] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in *Proc. of the 7th Symp. on Operating Systems Design and Implementation (OSDI '06)*, pp. 381–396, 2006.

[14] T. Liu, C. M. Sadler, P. Zhang, and M. Martonosi, "Implementing software on resource-constrained mobile sensors: experiences with impala and zebranet," in *Proc. of the 2nd Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys '04)*, pp. 256–269, 2004.

[15] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Wireless sensor networks for structural health monitoring," in *Proc. of the 4th Intl. Conf. on Embedded Networked Sensor Systems (SenSys '06)*, pp. 427–428, 2006.

[16] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, T. He, A. Tirumala, Q. Cao, J. A. Stankovic, T. Abdelzaher, and B. Krogh, "Lightweight detection and classification for wireless sensor networks in realistic environments," in *Proc. of the 3rd ACM Intl. Conf. on Embedded Networked Sensor Systems (SenSys '05)*, 2005.

[17] "GreenOrbs project," *http://greenorbs.org*. [last access: 07/21, 2012].

[18] L. A. Barroso, J. Dean, and U. Hölzle, "Web search for a planet: The Google cluster architecture," *IEEE Micro*, vol. 23, no. 2, pp. 22–28, 2003.

[19] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," in *Proc. of the 6th Symp. on Operating Systems Design and Implementation (OSDI '04)*, 2004.

[20] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, pp. 1–26, 2008.

[21] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks," *Proc. of the 5th ACM Symp. on Operating System Design and Implementation (OSDI '02)*, 2002.

[22] G. Werner-Allen, K. Loricz, M. Welsh, M. Ruiz, J. Lees, J. Johnson, and O. Marcillo, "Deploying a sensor network on an active volcano," *IEEE Computing, Special Issue on Data-Driven Applications in Sensor Networks*, Mar./Apr. 2005.

[23] J. Dean, "Designs, lessons and advice from building large distributed systems (keynote)," in *The 3rd ACM Intl. Workshop on Large Scale Distributed Systems and Middleware (LADIS)*, Oct. 11-14 2009.

[24] J. Hui, "Deluge 2.0 - TinyOS network programming," July 2005.

[25] P. Levis, N. Patel, S. Shenker, and D. Culler, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor network," *In Proc. of the 1st USENIX/ACM Symp. on Networked Systems Design and Implementation (NSDI '04)*, 2004.

[26] K. Lorincz and M. Welsh, "Motetrack: a robust, decentralized approach to rf-based location tracking," *Personal Ubiquitous Comput.*, vol. 11, pp. 489–503, Aug. 2007.

[27] "Zeus botnet controller," *http://aws.amazon.com/security/security-bulletins/zeus-botnet-controller*. [last access: 07/21, 2012].

[28] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono, "On technical security issues in cloud computing," in *Proc. of the 2009 IEEE Intl. Conf. on Cloud Computing (CLOUD '09)*, pp. 109–116, 2009.

[29] Google, "Security Whitepaper: Google Apps Messaging and Collaboration Products," 2010.

[30] M. Ding, D. Chen, K. Xing, and X. Cheng, "Localized fault-tolerant event boundary detection in sensor networks," in *Proc. of the 24th Annual Joint Conf. on Computer Communications (INFOCOM 2005)*, vol. 2, pp. 902 – 913 vol. 2, march 2005.

[31] N. Li and J. C. Hou, "Flss: a fault-tolerant topology control algorithm for wireless networks," in *Proc. of the 10th Annual Intl. Conf. on Mobile Computing and Networking (MobiCom '04)*, pp. 275–286, 2004.

[32] L. Luo, T. F. Abdelzaher, T. He, and J. A. Stankovic, "Design and comparison of lightweight group management strategies in envirosuite," in *Distributed Computing in Sensor Systems (DCOSS '05)*, June 2005.

[33] L. Lamport, "The part-time parliament," *ACM Trans. Comput. Syst.*, vol. 16, no. 2, pp. 133–169, 1998.

[34] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *Proc. of the 9th ACM Symp. on Operating Systems Principles (SOSP '03)*, pp. 29–43, 2003.

[35] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc. of the 34th Annual Intl. Symp. on Computer Architecture (ISCA '07)*, pp. 13–23, 2007.

[36] X. Wang, X. Wang, G. Xing, J. Chen, C.-X. Lin, and Y. Chen, "Towards optimal sensor placement for hot server detection in data centers," in *Proc. of the 31st Intl. Conf. on Distributed Computing Systems (ICDCS)*, pp. 899 –908, June 2011.

[37] W. Dong, Y. Liu, X. Wu, L. Gu, and C. Chen, "Elon: enabling efficient and long-term reprogramming for wireless sensor networks," in *Proc. of the ACM Intl. Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS '10)*, pp. 49–60, 2010.

[38] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust distributed network localization with noisy range measurements," in *Proc. of the 2nd ACM Conf. on Embedded Networked Sensor Systems (SenSys '04)*, Nov. 2004.

[39] E. Elnahrawy, X. Li, and R. Martin, "The limits of localization using signal strength: a comparative study," in *Proc. of the 1st Annual Conf. on Sensor and Ad Hoc Communications and Networks (SECON 2004)*, pp. 406 – 414, oct. 2004.

[40] M. Maróti, P. Völgyesi, S. Dóra, B. Kusý, A. Nádas, A. Lédeczi, G. Balogh, and K. Molnár, "Radio interferometric geolocation," in *Proc. of the 3rd Intl. Conf. on Embedded Networked Sensor Systems (SenSys '05)*, pp. 1–12, 2005.