Real-Time Data Services for Cyber Physical Systems

Kyoung-Don Kang
Department of Computer Science
State University of New York at Binghamton
kang@cs.binghamton.edu

Sang H. Son
Department of Computer Science
University of Virginia
son@cs.virginia.edu

Abstract

Cyber Physical Systems (CPSs) have grand visions with great socio-economic impacts such as blackout-free electricity supply and real-time disaster recovery. A key challenge is providing real-time data services for CPSs. Existing real-time data management techniques and wireless sensor networks (WSNs) fall far short to support timely, secure real-time data services for CPSs. In this paper, we present a novel information-centric approach to supporting these requirements in CPSs. In our approach, network-enabled real-time embedded databases (nRTEDBs) communicate with each other and control and communicate with wireless sensors in a secure, timely manner. Unlike sensor databases such as TinyDB, nRTEDBs collaboratively derive global knowledge of real world phenomena. Based on the collective information, they actively control a WSN to extract important data directly relevant to an event of interest. In this way, nRTEDBs considerably enhance the overall timeliness, security, and efficiency.

1 Introduction

Cyber physical systems (CPSs) interconnect the cyber world with physical world by embedding sensors and computational nodes to the physical world. CPSs envision significant socio-economic impacts. Applications of CPSs include, but are not limited to, electric grid management, assisted living, transportation management, disaster recovery, factory automation, smart spaces, military applications, and environmental science research. CPSs not only have rosy visions but also face challenges. One of the key challenges is providing real-time data services for CPSs: CPSs have to deal with large amounts of data in a timely, secure fashion. Although real-time data management and wireless sensor networks (WSNs) have been well studied separately, very little prior work has been done to integrate the two for CPSs.

To shed light on this problem, we propose a novel *approach for timely, secure information services* in CPSs. Specifically, we present network-enabled real-time embedded databases (nRTEDBs) and describe how to integrate them with wireless sensors. nRTEDBs are different from sensor databases such as TinyDB [14] and Cougar [3], since nRTEDBs actively derive value-added information from raw sensor data and control WSNs based on the observed real world phenomena. Also, nRTEDBs are designed to consider security and timeliness.

In this paper, we extend and integrate cutting-edge real-time data service techniques as well as real-time and secure routing protocols in WSNs. nRTEDBs have more energy, computational power, and communication bandwidth than sensor nodes. They communicate with each other to collectively detect important real world events, if any, using raw sensor data. Wireless sensors transmit data to one of the nRTEDBs in the vicinity in a timely, secure way under the guidance of nRTEDBs.

The expected benefits of our architecture for real-time data services in CPSs and system requirements are discussed in Section 2. Section 3 describes an initial high level system design for real-time data services in CPSs. Related work is discussed in Section 4. Finally, Section 5 concludes the paper and discusses future work.

2 System Requirements

By deploying multiple nRTEDBs in the field, the number of hops from a sensor node to a nRTEDB is reduced. As a result, the end-to-end (E2E) packet transmission delay and packet delivery rate can be considerably improved. For example, if the one-hop delivery rate is 95%, the end-to-end delivery rate drops below 60% after 10 hops. In contrast, the rate is over 77% when the communication path between a sensor and

nRTEDB has 5 hops. Further, a sensor node alternates between multiple nRTEDBs in the neighborhood for load balancing.

nRTEDBs extract information from raw sensor data and exchange them with each other to build a global view of real world phenomena. As a whole, nRTEDBs and WSNs are *information-centric* rather than being data-centric. Based on the derived knowledge, nRTEDBs significantly enhance the efficiency of sensing. For example, nRTEDBs can derive the perimeter of a fire and predict the movement of a fire based on sensor data such as the temperature, humidity, and wind direction to wake up sensors in areas to which the fire is expected to move or expand.

Moreover, nRTEDBs enhance WSN security. Surprisingly, a large body of WSN security protocols consider a single base station, and simply assume that the base station cannot be compromised. Thus, the base station is a single point of failure/attack in these approaches. By collaborating with each other, nRTEDBs alleviate this problem and other key security concerns.

To support these desirable features, nRTEDBs need to meet the following system requirements:

- **Real-Time Data Processing:** Given sensor data, a nRTEDB needs to process queries within certain deadlines or response time bounds.
- **Reliable Event Detection:** nRTEDBs need to support highly accurate event detection despite potentially noisy, faulty, or compromised sensor data.
- **Real-Time Routing:** Sensor data must be delivered to nRTEDBs in a timely fashion. Also, nRTEDBs need to efficiently exchange information with each other, while controlling sensors.
- **Security and Robustness:** nRTEDBs need to avoid the single point of attack problem discussed before in addition to data confidentiality, integrity, and authenticity. Overall, it is required that timeliness, security, and reliability requirements need to be considered together at design time.

3 Real-Time Sensor Data Services

A nRTEDB supports query processing, data fusion, routing, load balancing, and security. A nRTEDB has random access memory and flash memory bigger than sensor nodes by several orders of magnitude. For wireless communication, it supports spread spectrum techniques such as frequency hopping. Thus, nRTEDBs can communicate with each other, while avoiding interference and radio jamming attacks. A nRTEDB has much more energy than a sensor node and it may recharge the battery using ambient energy such as solar energy or vibration. To avoid interference, sensor nodes use multiple wireless channels to transmit data to different nRTEDBs, similar to [11]. This approach is feasible, since low-end sensors such as MICA-Z motes support multichannel wireless communication. Key components for real-time data services are discussed next.

3.1 Real-Time Data Processing

Although real-time databases have been studied for more than a decade, very few existing RTDB systems can support the desired timeliness in the presence of dynamic workloads [17]. To meet deadlines or response time bounds for real-time data services in the presence of dynamic workloads, we propose to apply feedback control as shown in Figure 1.

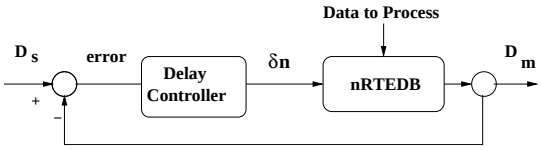

Figure 1. Feedback Control of Data Service Delay in One nRTEDB

In our real-time data service model, continuous queries and data fusion functions are registered. A continuous query or fusion function is periodically executed for a specified TTL (time to live). On the other hand, the number of incoming data to process may vary in time due to the dynamic network and real world status. Thus, the number of data to process is essentially disturbance to real-time data services as shown in Figure 1. To support the desired data service delay D_s such as 0.1s, the feedback controller in Figure 1 computes the required number of data adjustment δn. When the measured delay $D_m > D_s$, $\delta n < 0$ to reduce the number of data to process and vice versa. We have implemented a preliminary version of this control model in our real-time database system called Chronos [7]. We have verified that the model supports the desired response time for bursty transactional workloads.

To actually adjust the number of data in a WSN, a nRTEDB adapts the data similarity threshold [10]. For example, when the similarity threshold is 1%, a sensor reading that is not different from its previous version by more than 1% is dropped. This threshold is raised according to the control signal, if necessary, to support D_s under overload. Alternatively, a nRTEDB may drop sensor data with relatively low confidence

levels for event detection. Reliable event detection is discussed in Section 3.2.

After adaptation via feedback control, a nRTEDB disseminates a new similarity or confidence threshold to sensor nodes in the physical world to let them avoid unnecessary data transmissions in the first place. To minimize the overhead, control messages to adapt behaviors of sensor nodes are piggybacked to periodic beacon messages for real-time routing. Beacon messages are also used for load balancing. A detailed discussion is given in Section 3.3.

We will extend Chronos to support data and information storage in flash memory. Flash memory is rapidly replacing disks not only in real-time embedded systems, but also in high-performance servers [19]. Unlike disks, the access time to flash memory is not affected by mechanical parts. Thus, flash memory is several orders of magnitude faster than disks and highly predictable. These desirable characteristics make flash memory more suitable for nRTEDBs. However, flash memory still has high access time compared to volatile memory such as SRAM. Table 1 compares the overhead of flash memory to SRAM.

Type	Read	Write	Erase
SRAM	10ns	10ns	N/A
NAND Flash	10μs	200μs	2ms

Table 1. Characteristics of memory devices [15].

Such gaps in access time are critical for nRTEDBs, because high I/O overheads for flash memory access may incur deadline misses, if data objects are not properly managed. System designers may provide buffer memory to mitigate the high overhead of flash memory accesses. However, in many cases, workloads are unknown at design time and they may change dynamically. Thus, it may not be feasible to provide enough buffer space to satisfy all timing constraints. In particular, embedded systems cannot afford to have large buffers due to cost and space constraints.

We have developed a deadline miss ratio management architecture where a feedback control loop prevents overload both for I/O and CPU [8]. A key issue in real-time data processing with different kinds of resources, e.g. I/O and CPU, is to find out which resource is the bottleneck causing deadline misses. A straightforward approach would be to build separate models for I/O and CPU. However, our experiments show that CPU and I/O deadline miss ratios are coupled and affect each other, requiring multiple-input multiple-output (MIMO) modeling of the system. Thus, we model a nRTEDB as a MIMO system to capture this coupling of control inputs and system outputs. The current MIMO model is verified via simulations. This will be actually implemented and evaluated in Chronos. Further, we will extend the MIMO model to consider wireless link deadline miss ratio in addition to CPU and I/O deadline miss ratios. For real-time sensor data services, we also support event detection, real-time routing, and security as follows.

3.2 Reliable Event Detection

Due to noisy, faulty sensor readings in the physical world, event detection could be error prone. A way to address this problem is for sensor nodes to compute the confidence level of a detected event, e.g., a wild fire, in a certain range such as [0,1] [12]. In this paper, sensor nodes forward sensor data related to the event to a nearby nRTEDB, only if the confidence level exceeds the threshold, e.g., 0.7, specified by the nRTEDB. nRTEDBs receiving event data collaboratively derive an *event map* such as the perimeter of a fire. In contrast, sensors detecting no event or events with low confidence may sleep, periodically waking up to receive control messages from nRTEDBs. In this way, they save precious energy, while reducing unnecessary contention for wireless medium. The confidence function developed in [12] is specialized for fire detection. We will investigate data fusion algorithms for reliable event detection in more general WSN applications. As data fusion algorithms such as the Kalman filter and Bayesian networks are computationally more demanding than simple confidence functions [12], a low-end sensor node may not be able to fully execute data fusion algorithms. In this case, partially processed data or aggregated raw sensor data are forwarded to a nRTEDB, which finishes data fusion to derive more reliable information from sensor data.

3.3 Efficient Real-Time Routing

In our approach, there are two categories of wireless communications: (1) communication between sensor nodes or communication between a nRTEDB and sensors; and (2) communication between nRTEDBs. These two categories of communications need to use different frequency bands to avoid interference.

3.3.1 Communication between Sensors and nRTEDBs

A number of real-time routing protocols in WSNs such as SPEED [5] and MMSPEED [4] rely on greedy geographic forwarding (GF) [9], which is lightweight and

scalable. Unfortunately, GF fails in the presence of a void. Back-pressure routing is supported in SPEED and MMSPEED. However, back propagations in the presence of a void could be too late, incurring deadline misses. Also, simply implementing void traversal is inappropriate for real-time routing due to the high computational cost.

In our approach, multiple nRTEDBs are deployed in the area of sensing to reduce the number of hops, and therefore, enhance the E2E timeliness and reliability as discussed before. nRTEDBs periodically broadcast beacon messages to let sensors build or repair *shortest paths* to nRTEDBs in the proximity, similar to TinyOS's shortest routing protocol. When a sensor node receives a beacon message with the shortest delay, it marks the sending node as the parent and rebroadcasts the beacon message with an incremented hop count. In this way, each node knows to which parent it has to forward a packet to reach the corresponding nRTEDB via the shortest path. When a nRTEDB is overloaded, it increases the beacon period. As a result, paths from sensors to this nRTEDB will expire early, reducing loads for this nRTEDB. Workloads are distributed to other nRTEDBs in the vicinity, which broadcast beacons frequently. In this way, *load balancing* is performed between nRTEDBs in a close area without requiring centralized control.

Even in the absence of a void, we have experimentally shown that shortest path routing outperforms GF in terms of the E2E deadline miss ratio [13]. Essentially, GF is not optimal but greedy. A drawback of shortest path routing is that it implicitly assumes link bidirectionality: A node assumes that it can reach the parent node with the shortest delay, since the parent forwarded it the beacon message first. However, link bidirectionality does not necessarily hold in wireless networks. To address this problem, we extend shortest path routing via link quality estimation. A nRTEDB periodically broadcasts a beacon message. When a sensor node receives $k(> 1)$ duplicate beacon messages from the nRTEDB, it records all the k one hop neighbors that forwarded the beacons to itself. When the node transmits a packet towards the nRTEDB, the node randomly forwards a data packet to one of the k one hop neighbors and measures the actual delay and reliability, i.e., delivery ratio, of the link from itself to the selected one hop neighbor. After sending a certain number of packets to these neighbors, the node forwards data to a neighbor, which provides a high quality link, with a high probability. We consider a probabilistic approach to support resilience to network dynamics, while distributing loads between links.

To measure the link quality, we extend a well known link quality metric called ETX (Expected Transmission Count) [2]. ETX indicates how many (re)transmissions are necessary to deliver a single packet through the observed link. The higher is the ETX, the lower is the network throughput. The ETX of a path consisted of n links $l_1, l_2, ..., l_n$ is equal to $\sum_{i=1}^{n} ETX_i$ where ETX_i is the ETX of l_i. A limitation of ETX is that it does not consider the delay but only considers the delivery ratio. Thus, for real-time routing, we compute the *estimated transmission delay* ETD_i for link i:

$$ETD_i = delay \cdot ETX_i \tag{1}$$

where the delay is the exponentially weighted moving average of one hop transmission delay of l_i. We ignore the propagation delay, since radio signals travel at the speed of light. The one hop speed supported by l_i is:

$$S_i = 1/ETD_i \tag{2}$$

and the required speed for a link on the path with n hops is:

$$S_r = n/\text{E2E deadline}. \tag{3}$$

A node probabilistically chooses a neighbor node i if $S_i \geq S_r$ as the next hop. If S_i is higher, then node i is more likely to be selected.

In addition, our information-centric approach is different from most existing real-time routing protocols where data or event semantics are not considered:

- nRTEDBs serve as *proxies* for sensor nodes. They answer certain queries based on the derived data and information rather than flooding every query to sensor nodes and collecting data from them.
- Based on the real world status, sensor nodes *adjust reporting periods*. A set of nRTEDBs in a region indicates which sub-regions are important for event detection. This control information is piggybacked to beacon messages from nRTEDBs. Thus, sensors in unimportant areas increase their reporting periods or sleep in a distributed manner.
- The speed of packet delivery is *differentiated*, similar to MMSPEED [4]. However, unlike MMSPEED, packets are differentiated according to the importance and confidence of the detected event. Unimportant or low confidence data are dropped at the source. In contrast, an important event data with high confidence is transmitted using high quality links. Further, a critical data is forwarded to more than one nRTEDB in the proximity.

Thus, nRTEDBs significantly increase the efficiency of event detection and network lifetime (or time-to-recharge), while reducing variations in service delays.

3.3.2 Communication between nRTEDBs

To derive information from raw sensor data, nRTEDBs communicate with each other via an on demand ad hoc routing protocol. In our approach, when a nRTEDB detects an event, it communicates with the closest nRTEDBs first. Thus, the nRTEDBs in only one hop range are invited in the first phase. It incrementally extends the communication scope to two hop nRTEDB neighbors after exchanging a certain number of event messages with one hop neighbors and so on. A variation of this protocol disseminates event information in a differentiated manner. If a detected event such as a fire is more critical, the information is disseminated faster among nRTEDBs by increasing the scope in proportion to the criticality such as the intensity of the detected fire. In this way, nRTEDBs collaborate as necessary considering the importance of detected events, if any. Also, this protocol supports mobility of nRTEDBs. When a nRTEDB in an important region fails, for example, another nRTEDB in a region with no interesting event may move to the important area. If nRTEDBs cannot communicate with each other due to network problems, nRTEDBs locally process sensor data and store extracted event information to support real-time information services when the network recovers. During the network partition time, a nRTEDB directly transmits locally processed information to a nearby user (if any) such as a rescue person with a personal digital assistant in a disaster area or an unmanned aerial vehicle flying over a battlefield. Thus, real-time information can be available even when the network is partitioned.

3.4 Security and Robustness

By taking an information-centric approach, we address several security issues. In our approach, multiple nRTEDBs in a region derive a *consensus* for event detection based on sensor data received from a set of sensor nodes in the region. If a nRTEDB gives a significantly different result from others in event detection, this nRTEDB could be under attack. As long as a majority of nRTEDBs are not compromised, this approach can detect malfunctioning nRTEDBs, if any.

Similarly, nRTEDBs can alleviate the problem of *false data injection* by compromised or malicious sensor nodes. For example, consider that a small set of sensors continuously reports a fire, but nearby sensors do not detect the reported fire or spread of the fire for a long period of time. In this case, the set of sensors could be faulty or colluding for false event reports.

The problem of *single point of attack* is alleviated by our approach, as load balancing is performed under failure or attack by adapting the beacon period. Also, another nRTEDB, if available, takes over when a nRTEDB in an important area fails as discussed before. nRTEDBs rely on spread spectrum techniques to communicate with each other. Also, sensor nodes use different radio frequencies to communicate with different nRTEDBs as discussed before. Thus, an adversary has to jam and overhear the entire broadband for *jamming and traffic analysis*.

By reducing the number of hops between a source and sink, nRTEDBs reduce the chance of information leakage. nRTEDBs reduce the overhead due to repetitive link-layer encryption and decryption at intermediate nodes needed for in-network data aggregation. As a result, the timeliness of secure routing is enhanced. Further, via an authenticated broadcast protocol such as μTESLA [16], nRTEDBs securely broadcast queries and their deadlines. Each node can verify that a query and its deadline are actually originated from a nRTEDB without being altered in transit. Thus, it is hard for an adversary to make sensor nodes misbelieve it as a nRTEDB or increase/decrease deadlines to disrupt real-time routing.

Further, we observe that WSN security support can be optimized by considering *application data semantics* [18]. For example, there is usually no need to support confidentiality of sensor data for scientific research such as a study of global warming, while the confidentiality of data must be supported for battlefield monitoring. False data injection, jamming, and traffic analysis are unlikely to happen for environmental science research too. Thus, excessive security measures should be avoided for timeliness and energy efficiency. Despite the importance, the key research issues described in this section have rarely been studied in WSNs.

4 Related Work

Real-time data management has been studied for more than a decade, producing fruitful research results [17]. However, most existing work only considers centralized real-time data management techniques. Very little prior work has been done for distributed real-time data management. Also, timely, secure communication is not considered.

Most existing sensor databases such as TinyDB [14], Cougar [3], and SINA [20] focus on the database metaphor and in-network processing. They do not consider security and real-time issues.

A plenty of work has been done for real-time routing including [1, 5, 4, 6, 13]. However, these protocols do not consider the information-centric approach and security issues discussed in this paper. On the other

hand, WSN security schemes such as [16] do not consider real-time requirements. Hence, they cannot be directly applied to real-time data services in CPSs. In contrast, nRTEDBs enhance security in several ways, while improving the real-time performance and energy efficiency as discussed before.

5 Conclusions and Future Work

In this paper, we present a novel information-centric approach for timely, secure real-time data services in CPSs. In our approach, to derive global knowledge of real world phenomena, network-enabled real-time embedded databases communicate with each other, while controlling and communicating with WSNs in a secure, timely manner. Based on the collective information, WSNs are controlled to extract important data directly related to an event of interest. By taking the information-centric approach, nRTEDBs can considerably enhance the efficiency of sensing, while improving timeliness and security. In the future, we will investigate more efficient routing, event detection, data fusion, and security protocols.

References

[1] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo. An Implicit Prioritized Access Protocol for Wireless Sensor Networks. In *Proceedings of the IEEE Real-Time Systems Symposium*, 2002.

[2] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In *MobiCom*, 2003.

[3] A. J. Demers, J. Gehrke, R. Rajaraman, A. Trigoni, and Y. Yao. The Cougar Project: A Work-in-Progress Report. *SIGMOD Record*, 32(4), 2003.

[4] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural. Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks. *IEEE INFOCOM*, 2005.

[5] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A Stateless Protocol for Real-Time Communications in Sensor Networks. In *International Conference on Distributed Computing Systems*, 2003.

[6] Q. Huang, C. Lu, and G.-C. Roman. Mobicast: Just-in-time multicast for sensor networks under spatiotemporal constraints. *International Workshop on Information Processing in Sensor Networks (IPSN'03)*, April 2003.

[7] K. D. Kang, P. H. Sin, J. Oh, and S. H. Son. A Real-Time Database Testbed and Performance Evaluation. In *the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2007.

[8] W. Kang, S. H. Son, J. Stankovic, and M. Amirijoo. I/O-Aware Deadline Miss Ratio Management in Real-Time Embedded Databases. *28th IEEE Real-Time Systems Symposium (RTSS'07)*, 2007.

[9] B. Karp and H. Kung. GPSR: Greedy Perimeter stateless Routing for Wireless Networks. In *MobiCom*, 2000.

[10] T.-W. Kuo and A. K. Mok. Real-Time Database - Similarity Semantics and Resource Scheduling. *ACM SIGMOD Record*, 25(1), 1996.

[11] H. K. Le, D. Henriksson, and T. Abdelzaher. A Control Theory Approach to Throughput Optimization in MultiChannel Collection Sensor Networks. In *Information Processing in Sensor Networks*, 2007.

[12] S. Li, Y. Lin, S. H. Son, J. Stankovic, and Y. Wei. Event Detection Services using Data Service Middleware in Distributed Sensor Networks. *Telecommunication Systems, Special Issue on Information Processing in Sensor Networks*, 26(2-4), 2004.

[13] K. Liu, N. Abu-Ghazaleh, and K. D. Kang. JiTS: Just-in-Time Scheduling for Real-Time Sensor Data Dissemination. In *the 4th Annual IEEE International Conference on Pervasive Computing and Communications*, 2006.

[14] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TinyDB: An Acquisitional Query Processing System for Sensor Networks. *ACM Transactions on Database Systems*, 30(1):122–173, 2005.

[15] C. Park, J. Seo, D. Seo, S. Kim, and B. Kim. Cost-Efficient Memory Architecture Design of NAND Flash Memory Embedded Systems. *21st International Conference on Computer Design*, 2003.

[16] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar. SPINS: Security Protocols for Sensor Networks. In *MobiCom*, 2001.

[17] K. Ramamritham, S. H. Son, and L. DiPippo. Real-Time Databases and Data Services. *Real-Time Systems Journal*, 28(2-3):179–215, Nov.-Dec. 2004.

[18] E. Sabbah, K. D. Kang, N. Abu-Ghazaleh, A. Majeed, and K. Liu. An Application-Driven Approach to Designing Secure Wireless Sensor Networks. *Wireless Communications and Mobile Computing, Wiley Interscience*, 2007. To Appear.

[19] http://www.samsung.com/products/semiconductor/flash/.

[20] C.-C. Shen, C. Srisathapornphat, and C. Jaikaeo. Sensor Information Networking Architecture and Applications. *IEEE Personel Communication Magazine*, 8(4):52–59, 2001.