Mobility and Handoff Management in 5G

Er.Avni Sharma
Department of Computer Science and Engineering
Hamirpur, India
avnisharma910@gmail.com

Gourav Kumar Dhiman
Department of Computer Science and Engineering
Hamirpur, India
gourav.dhiman331@gmail.com

ABSTRACT

This paper presents a survey on the current state-of-the-art in Wireless Sensor Network (WSN) Operating Systems (OSs). In recent years, WSNs have received tremendous attention in the research community, with applications in battlefields, industrial process monitoring, home automation, and environmental monitoring, to name but a few. A WSN is a highly dynamic network because nodes die due to severe environmental conditions and battery power depletion. Furthermore, a WSN is composed of miniaturized motes equipped with scarce resources e.g., limited memory and computational abilities. WSNs invariably operate in an unattended mode and in many scenarios it is impossible to replace sensor motes after deployment, therefore a fundamental objective is to optimize the sensor motes' life time. These characteristics of WSNs impose additional challenges on OS design for WSN, and consequently, OS design for WSN deviates from traditional OS design. The purpose of this survey is to highlight major concerns pertaining to OS design in WSNs and to point out strengths and weaknesses of contemporary OSs for WSNs, keeping in mind the requirements of emerging WSN applications. The state-of-the-art in operating systems for WSNs has been examined in terms of the OS Architecture, Programming Model, Scheduling, Memory Management and Protection, Communication Protocols, Resource Sharing, Support for Real-Time Applications, and additional features. These features are surveyed for both real-time and non-real-time WSN operating systems.

Keywords— Wireless Sensor Network (WSN), Operating Systems (OS), embedded operating system, Real-Time Operating System (RTOS)

1. Operating Systems for Sensor Networks

I. INTRODUCTION

Advances in networking and integration have enabled small, flexible, lowcost nodes that interact with their environment and with each other through sensors, actuators and communication. Single-chip systems are now emerging that integrate a low-power CPU and memory, radio or optical communication , and MEMS-based on-chip sensors. The low cost of these systems enables embedded networks of thousands of nodes for applications ranging from environmental and habitat monitoring, seismic analysis of structures, and object localization and tracking .

II. Design Issues

A. **Architecture**

The organization of an OS constitutes its structure. The architecture of an OS has an influence on the size of the OS kernel as well as on the way it provides services to the application programs. Some of the well known OS architectures are the monolithic architecture, the micro-kernel architecture, the virtual machine architecture and the layered architecture.

A monolithic architecture in fact does not have any structure. Services provided by an OS are implemented separately and each service provides an interface for other services. Such an architecture allows bundling of all the required service together into a single system image, thus results in a smaller OS memory footprint. An

advantage of the monolithic architecture is that the module interaction costs are low. Disadvantages associated with this architecture are: the system is hard to understand and modify, unreliable, and difficult to maintain. These disadvantages associated with monolithic kernels make them a poor OS design choice for contemporary sensor nodes.

An alternate choice is a microkernel architecture in which minimum functionality is provided inside the kernel. Thus, the kernel size is significantly reduced. Most of the OS functionality is provided via user-level servers like a file server, a memory server, a time server, *etc*. If one server fails, the whole system does not crash. The microkernel architecture provides better reliability, ease of extension and customization. The disadvantage associated with a microkernel is its poor performance because of frequent user to kernel boundary crossings. A microkernel is the design choice for many embedded OS due to the small kernel size and the number of context switches in a typical WSN application is considered to be far fewer. Thus, fewer boundary crossing are required compared to traditional systems.

A virtual machine is another architectural choice. The main idea is to export virtual machines to user programs, which resemble hardware. A virtual machine has all the needed hardware features. The key advantage is its portability and a main disadvantage is typically a poor system performance.

A layered OS architecture implements services in the form of layers. Advantages associated with the layered architecture are: manageability, easy to understand, and reliability. A main disadvantage is that it is not a very flexible architecture from an OS design perspective.

An OS for a Wireless Sensor Network should have an architecture that results in a small kernel size, hence small memory footprint. The architecture must allow extensions to the kernel if required. The architecture must be flexible *i.e*., only application-required services get loaded onto the system

B. **Programming Models**

The programming model supported by an OS has a significant impact on the application development. There are two popular programming models provided by typical WSN OSs, namely: event driven programming and multithreaded programming. Multithreading is the application development model most familiar to programmer, but in its true sense rather resource intensive, therefore not considered well suited for resource constraint devices such as sensor nodes. Event driven programming is considered more useful for computing devices equipped with scarce resource but not considered convenient for traditional application developers. Therefore researchers have focused their attention on developing a light-weight multithreading programming model for WSN OSs. Many contemporary WSN OSs now provide support for the multithreading programming model and we discuss them in detail later.

C. **Scheduling**

The Central Processing Unit (CPU) scheduling determines the order in which tasks are executed on a CPU. In traditional computer systems, the goal of a scheduler is to minimize latency, to maximize throughput and resource utilization, and to ensure fairness.

The selection of an appropriate scheduling algorithm for WSNs typically depends on the nature of the application. For applications having real-time requirements, real-time scheduling algorithm must be used. For other applications, non-real-time scheduling algorithms are sufficient.

WSNs are being used in both real-time and non-real-time environments, therefore a WSN OS must provide scheduling algorithms that can accommodate the application requirements. Moreover, a suitable scheduling algorithm should be memory and energy efficient.

D. Memory Management and Protection

In a traditional operating system, memory management refers to the strategy used to allocate and de-allocate memory for different processes and threads. Two commonly used memory management techniques are static memory management and dynamic memory management. Static memory management is simple and it is a useful technique when dealing with scare memory resources. At the same time, it results in inflexible systems because run-time memory allocation cannot occur. On the other hand, dynamic memory management yields a more flexible system because memory can be allocated and de-allocated at run-time. Process memory protection refers to the protection of one process' address space from another. In early sensor network operating systems like TinyOS [3] there was no memory management available. Initial operating systems for WSNs assumed that only a single application executes on a sensor mote, therefore there is no need for memory protection. With the emergence of new application domains for WSNs, contemporary WSNs provides support for multiple threads of execution, consequently memory management becomes an issue for WSN OS.

E. Communication Protocol Support

In the OS context, communication refers to inter-process communication within the system as well as with other nodes in the network. WSNs operate in a distributed environment, where senor nodes communicate with other nodes in the network. All WSN OSs provide an Application Programming Interface (API) that enables application program to communicate. It is possible that a WSN is composed of heterogeneous sensor nodes, therefore the communication protocol provided by the OS must also consider heterogeneity. In network-based communication, the OS should provide transport, network, and MAC layer protocol implementations.

F. Resource Sharing

The responsibility of an OS includes resources allocation and resource sharing, which is of immense importance when multiple programs are concurrently executing. The majority of WSNs OSs today provide some sort of multithreading, requiring a resource sharing mechanism. This can be performed in time e.g., scheduling of a process/thread on the CPU and in space e.g., writing data to system memory. In some cases, we need serialized access to resources and this is done through the use of synchronization primitives.

III. Examples of Operating Systems

A. Emeralds

EMERALDS is an extensible microkernel written in C++ for embedded, real-time distributed systems with embedded applications running on slow processors (15 to 25 MHz) and with limited memory (32 to 128 kB). It supports multithreaded processes and full memory protection, which are scheduled using combined earliest deadline first (EDF) and a rate- monotonic (RM) scheduler. The device drivers are implemented at the user level, whereas interrupt handling takes place at the kernel level. EMERALDS uses semaphores and condition variables for synchronization with priority inheritance at the same time and provides full semaphore semantics to reduce the amount of context switching. Interprocessor communication (IPC) is realized based on message passing, mailboxes, and shared memory, optimized especially for intranode, intertask communication. EMERALDS does not use a mailbox; it uses global variables to exchange information between tasks, to avoid message sending. EMERALDS does not consider networking issues.

B. PicOS

One property of OS microcontrollers with limited RAM is to try to allocate as little memory as possible to a process or thread. PicOS is written in C for a microcontroller with limited on- chip RAM (e.g., 4 kB). In PicOS, all tasks share the same global stack and act as coroutines with multiple entry points and implicit control transfer, which is different from classical multitasking approaches. In PicOS, each task is like a FSM where the state transition is triggered by events. The FSM approach is effective for reactive applications whose primary role is to respond to events rather than to process data or crunch numbers. The CPU cycle is multiplexed among multiple tasks, but the tasks can be pre-empted only at the FSM state boundary. It has few resource requirements and supports multitasking, a flat structure for processes but perhaps not good for real-time applications.

C. **SenOS**

SenOS is a finite state machine (FSM)-based operating system. It has three components:

- A kernel that contains a state sequencer and an event queue. The state sequencer waits for an input from the event queue (a FIFO queue).
- A state transition table that keeps the information on state transition and the corresponding call-back functions, Each state transition table defines an application. Using multiple state transition tables and switching among them, SenOS supports multiple applications in a concurrent manner.
- A call-back library of call functions. An incoming event will be queued in the event queue. The first event in the event queue is scheduled, which triggers a state transition and correspondingly, invokes the associated functions.

The kernel and call-back library are statically built and stored in the flash ROM of a sensor node, whereas the state transition table can be reloaded or modified at runtime since it is application dependent. Since SenOS is FSM-based, it can easily realize concurrency and reconfiguration. It can also be extended to network management.

IV. Node Level Simulators

Node-level design methodologies are usually associated with simulators that simulate the behavior of a sensor network on a per-node basis. Using simulation, designers can quickly study the performance (in terms of timing, power, bandwidth, and scalability) of potential algorithms without implementing them on actual hardware and dealing with the vagaries of actual physical phenomena.
A node-level simulator typically has the following components:

A. **Sensor node model**

A node in a simulator acts as a software execution platform, a sensor host, as well as a communication terminal. In order for designers to focus on the application-level code, a node model typically provides or simulates a communication protocol stack, sensor behaviors (e.g., sensing noise), and operating system services. If the nodes are mobile, then the positions and motion properties of the nodes need to be modeled. If energy characteristics are part of the design considerations, then the power consumption of the nodes needs to be modeled.

B. **Communication Model**

Depending on the details of modeling, communication may be captured at different layers. The most elaborate simulators model the communication media at the physical layer, simulating the RF propagation delay and collision of simultaneous transmissions. Alternately, the communication may be simulated at the MAC layer or network layer, using, for example, stochastic processes to represent low-level behaviors.

C. **Physical Environment Model**

A key element of the environment within which a sensor network operates is the physical phenomenon of interest. The environment can also be simulated at various levels of detail. For example, a moving object in the physical world may be abstracted into a point signal source. The motion of the point signal source may be modeled by differential equations or interpolated from a trajectory profile. If the sensor network is passive—that is, it does not impact the behavior of the environment—then the environment can be simulated separately or can even be stored in data files for sensor nodes to read in. If, in addition to sensing, the network also performs actions that influence the behavior of the environment, then a more tightly integrated simulation mechanism is required.

D. **Statistics and Visualization**

The simulation results need to be collected for analysis. Since the goal of a simulation is typically to derive global properties from the execution of individual nodes, visualizing global behaviors is extremely important. An ideal visualization tool should allow users to easily observe on demand the spatial distribution and mobility of the nodes, the connectivity among nodes, link qualities, end-to-end communication routes and delays, phenomena and their spatio-temporal dynamics, sensor readings on each node, sensor node states, and node lifetime parameters (e.g., battery power).

V. Performance and Traffic Management Issues

Wireless Sensor Networks(WSNs) have matured to a point where they present a realistic technology for monitoring non critical systems in industrial, office and domestic environments. This in turn will lead to an increased number of applications using WSN technology, each requiring a unique response from the underlying network. Due to the nature of WSN communications these different network requirements are achieved using a variety of communication tools. With ever increasing number and complexity of tools available it becomes difficult to choose which tool is best suited for an application in a given deployment.

Wireless Sensor Networks (WSNs) are event-driven network systems consist of many sensors node which are densely deployed and wirelessly interconnected that allow retrieving of monitoring data. In Wireless sensor network, whenever an event is detected, the data related to the event need to be sent to the sink node (data collection node). Sink node is the bottleneck of network there may be chance for congestion due to heavy data traffic. Due to congestion, it leads to data loss; it may be important data also. To achieve this objective, soft computing based on Neural Networks (NNs) Congestion Controller approach is proposed. The NN is activated using wavelet activation function that is used to control the traffic of the WSN. The proposed approach which is called as Modified Neural Network Wavelet Congestion Control (MNNWCC), has three main activities: the first one is detecting the congestion as congestion level indications; the second one is estimated the traffic rate that the upstream traffic rate is adjusted to avoid congestion in next time, the last activates of the proposed approach is improved the Quality of Services (QoS), by enhancement the Packet Loss Ratio (PLR), Throughput (TP), Buffer Utilization (BU) and Network Energy (NE) . The simulation results show that the proposed approach can avoid the network congestion and improve the QoS of network.

VI. Performance Modelling of WSNs

A critical issue in wireless sensor networks is represented by the limited availability of energy within network nodes; therefore making good use of energy is a must. A widely employed energy-saving technique is to place nodes in sleep mode, corresponding to a low-power consumption as well as to reduced operational capabilities. In this work, we develop a Markov model of a sensor network whose nodes may enter a sleep mode, and we use this model to investigate the system performance in terms of energy consumption, network capacity, and data deliver delay. Furthermore, the proposed model enables us to investigate the trade-offs existing between these performance metrics and the sensor dynamics in sleep/active mode. Analytical results present an excellent matching with simulation results for a large variety of system scenarios showing the accuracy of our approach.

VII. Emerging Applications and Future Research Directions

Wireless sensor networking technology has been used extensively by both commercial and military applications for sensing and data collection purposes. The self-configuring, self-healing nature and the ease of deployment of these networks make them an attractive option to other centralized approaches. Most of the existing networking solutions for sensor networks focus on the communication aspects and do not address the data security concerns of these networks. Since sensor networks are being deployed for emerging applications involving sensitive data and are envisioned to be integrated with the cyber space, it is essential to address the security needs of wireless sensor networks. Designing security solutions for Wireless Sensor Networks is an extremely challenging task due to the resource constraints of sensor nodes and the distributed nature of network design. This chapter provides an overview of emerging sensor networks involving sensitive data and a discussion of some of the proposed security solutions.

The future developments in sensor nodes must produce very powerful and cost effective devices, so that they may be used in applications like underwater acoustic sensor systems, sensing based cyber physical systems, time critical applications, cognitive sensing and spectrum management, and security and privacy management. In this section we will look into all possibilities of further development in WSN applications.

2. Mobility and Handoff Management in 5G

I. Network deployment types

Though 5G has been standardized, it has a number of options. Two network operators can deploy 5G in very different ways. This choice of option depends on the spectrum licensed to an operator, the geographic area they serve (terrain and user density), capabilities of the equipment they use, and business factors (cashflow and decision making).

3GPP has defined options covering both 4G and 5G technologies with respect to Radio Access Network (RAN) and Core Network (CN). These options can guide operators as they migrate from current 4G deployments to 5G deployments.

It's generally expected that operators would first deploy 5G NR, let 4G RAN and 5G NR coexist, and finally deploy 5G Core. This implies that 4G+5G handsets would come out first and they would connect to both 4G eNB and 5G gNB.

In LTE, both RAN and CN had to use LTE standards. 5G gives more flexibility. For example, 4G RAN can be combined with 5G Core or 5G NR can be combined with 4G EPC. This gives rise to two broad deployment scenarios:

- **Standalone (SA)**: Uses only one radio access technology, either LTE radio or 5G NR. Both control and user planes go through the same RAN element. Deployment and network management is perhaps simpler for operators. Inter-RAT handover is needed for service continuity. Under SA, we have option 1 (EPC + 4G eNB), option 2 (5GC + 5G gNB), and option 5 (5GC + 4G ng-eNB).
- **Non-Standalone (NSA)**: Multiple radio access technologies are combined. Control plane goes through what's called the master node whereas data plane is split across the master node and a secondary node. There's tight interworking between 4G RAN and 5G NR. Under NSA, we have option 3 (EPC + 4G eNB master + 5G en-gNB secondary), option 4 (5GC + 5G gNB master + 4G ng-eNB secondary), and option 7 (5GC + 4g ng-eNB master + 5g gNB secondary).

II. Interference management in 5G

In the modern technological world, wireless communication has taken a massive leap from the conventional communication system to a new radio communication network. The novel concept of Fifth Generation (5G) cellular networks brings a combination of a diversified set of devices and machines with great improvement in a unique way compared to previous technologies. To broaden the user's experience, 5G technology provides the opportunity to meet the people's potential necessities for efficient communication. Specifically, researchers have designed a network of small cells with unfamiliar technologies that have never been introduced before. The new network design is an amalgamation of various schemes such as Heterogeneous Network (HetNet), Device-to-Device (D2D) communication, Internet of Things (IoT), Relay Node (RN), Beamforming, Massive Multiple Input Multiple Output (M-MIMO), millimeter-wave (mm-wave), and so on. Also, enhancement in predecessor's techniques is required so that new radio is compatible with a traditional network. However, the disparate technological models' design and concurrent practice have created an unacceptable intervention in each other's signals. These vulnerable interferences have significantly degraded the overall network performance. This review article scrutinizes the issues of interferences observed and studied in different structures and techniques of the 5G and beyond network. The study focuses on the various interference effect in HetNet, RN, D2D, and IoT. Furthermore, as an in-depth literature review, we discuss various types of interferences related to each method by

studying the state-of-the-art relevant research in the literature. To provide new insight into interference issue management for the next-generation network, we address and explore various relevant topics in each section that help make the system more robust.

In a small cell wireless cellular network, multi-tier interferences are predetermined due to each low power node's specific attributes . It generates and receives continuously unwanted signals from various nearby sources. It is known that the HD (Half Duplex) mode limits a radio communication network's performance since it is transmitted or received at the same frequency. Different from HD, FD (Full Duplex) transmission mode transmits and receives signals simultaneously on the same frequency and is supported by a multi-antenna system that enhances the network capacity and minimizes the round trip data delivery time. However, though HD shows the capabilities to avoid interferences and provide quality signal strength , the delay in transmitting and receiving a signal and inefficient use of spectrum makes it undesirable for new radio wireless communication. In contrast, the FD supports higher throughput with lower latency and efficient use of spectrum . Also, it enhances the ergodic capacity and the network secrecy ; however, its performance is extremely descended due to interferences. Therefore, a robust and concrete interference mitigation scheme in FD transmission is required to deliver significant results for the practical future mobile network. The most common interferences associated with radio networks are self-, adjacent channel-, intra-, and inter-cell interference. Nonetheless, the mobile network is not limited to only these interferences. Each network is affected by interferences endured by their respective deployment and transmission scenario.

III. Mobility management in 5G

Throughout last decades, cellular networks have become disorderly spread on the globe. To organize the wireless networks, mobility management (e.g. handover management) is utilized. To mention about mobility management in LTE networks, this type of network utilizes only hard handover. As mentioned before, hard handover process run in break-beforemake principle and it causes some significant issues in mobility management processes. In order to provide a continuous connection to UE, eNB need to support as LTE does not include an RNC (Radio Network Controller) entity . On the other hand, because of massive increase in data traffic, grown in demand the LTE structures will not be applicable for future network scenarios. These current methods will be inadequate for cases of 5G future networks

As the use of 5G networks is more adopted, the main differences between 4G and 5G networks will be the outstanding benefits owing to mm-wave frequency bands, beam directional antennas, higher data rates, wider coverage, lower costs, higher capacities, etc. The mobility management services in 5G can be provided on cloud systems. 5G technology that is a packet switched system with outstanding results provides a more efficient and higher performance communication opportunities. And, users are able to utilize the technology and broadband internet connection by their mobile phones.

IV. Dynamic network reconfiguration in 5G

Botnets are one of the most powerful cyberthreats affecting continuity and delivery of existing network services. Detecting and mitigating attacks promoted by botnets become a greater challenge with the advent of 5G networks, as the number of connected devices with high mobility capabilities, the volume of exchange data, and the transmission rates increase significantly. Here, a 5G-oriented solution is proposed for proactively detecting and mitigating botnets in a highly dynamic 5G network. 5G subscribers' mobility requires dynamic network reconfiguration, which is handled by combining software-defined network and network function virtualization techniques.

The incoming fifth generation (5G) mobile technology aimsto offer huge data bandwidth and high networking capabilitiesto bring superb users' experiences on mobile communications.Meeting the demands of 5G is more than merely increasing thecomputational and bandwidth resources of the infrastructure,although these improvements still represent crucial driver. Although not every detail of 5G has been disclosed yet, newsecurity challenges will arise as discussed .The announced number and heterogeneity of the 5G devices,with high mobility capabilities, entail an even evolved threatslandscape. One of the most powerful threats in 5G will mainthe malicious actions lead by botnets, as in existing networks.A botnet is a network of thousands, millions of compromiseddevices known as bots, infected by an unconsciously installedmalware, going on to be controlled by a Command and Control(C&C) server remotely. Typically, recruited bots ask from timeto time to the C&C if they should trigger actions. How botnetsbehave, their architecture, and communication patterns amongbots and C&C

servers are widely reviewed. Among the potential malicious actions that a botnet ownercan request, Distributed Denial of Service (DDoS) attacks areoften the most commonly used today. Kaspersky Lab reportedfor Q3 2016 that the botnet-assisted DDoS attacks supposed78.9% of all detected attacks, where the largest number wasobserved on 3 August with 1,746 attacks. As real examples,the Mirai and Leet botnets launched in 2016 crippling DDoSattacks, reaching up to 650 Gbps of network traffic to disruptservices of Amazon and Netflix, among others.

REFERENCES

[1] Muhammad Omer Farooq and Thomas Kunz : Operating Systems for Wireless Sensors Networks
[2] Modelling the Performance of a WSN with Regard to the Physical Features Exhibited by the Network Declan T. Delaney & Gregory M. P. O'Hare
[3] Traffic Management in Wireless Sensor Network Based on Modified Neural Networks, June 2014 : Iraqi Journal for Computers and Informatics
[4] Modeling the performance of wireless sensor networks , April 2004 : Proceedings - IEEE INFOCOM
[5] Wireless Sensor Networks 'Future trends and Latest Research Challenges' Dr. Deepti Gupta
[6] Wireless Sensor Networks: Emerging Applications and Security Solutions
[7] https://devopedia.org/5g-deployment-options
[8] Interference Management in 5G and Beyond Network: Requirements, Challenges and Future Directions
[9] Mobility Management in 5G A.N. Kasim, Istanbul Technical University
[10] Dynamic Reconfiguration in 5G Mobile Networks to Proactively Detect and Mitigate Botnets