International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.7 May - June – 2023
DOI:10.46647/ijetms.2023.v07i03.131 ISSN: 2581-4621

Theoretical Study on Embedded Processor and Networking

Shreya Mane[1]
[1]*Research Fellow, Department of Research and Development, ASTROEX RESEARCH ASSOCIATION, Uttar Pradesh, India*

***Abstract*—** Embedded processors are specialized microprocessors designed to perform specific tasks within a larger system or device. They are commonly used in various applications such as consumer electronics, automotive systems, industrial automation, and Internet of Things (IoT) devices. This abstract focuses on the role of embedded processors in networking applications. Networking refers to the interconnection of devices and systems to enable communication and data exchange. Embedded processors play a crucial role in networking by providing the necessary computational power and functionality to handle networking protocols, data processing, and network management tasks. They are often integrated into network devices such as routers, switches, gateways, and network interface cards. The utilisation of networked embedded systems in contemporary network situations is undeniably significant, and they are receiving an increasing amount of attention. For network connectivity, data processing, and service delivery, the research community and industry are putting forth cutting-edge embedded solutions, frequently based on network processors. Despite this, it appears to be quite difficult to get quantitative performance comparisons of such systems.

***Keywords*— Networked systems, Embedded Sensor, Processor**

1. INTRODUCTION

Networking and embedded processors are two interrelated topics that are essential to the operation of contemporary technological systems. Microcontrollers and embedded processors are specialised computing devices created to carry out particular functions inside bigger systems or devices. They are frequently utilised in applications that demand real-time processing capability, small size, and low power consumption. Networking, on the other hand, refers to the interconnection of devices and systems to enable communication and data exchange. It encompasses a wide range of technologies and protocols that facilitate the seamless transmission of information between devices, enabling the creation of complex networks.

The integration of embedded processors with networking technologies has revolutionized various industries, ranging from consumer electronics to industrial automation and IoT applications. By leveraging the computational power and functionality of embedded processors, networking systems can efficiently handle data processing, protocol handling, and network management tasks.

More and more often, multimedia material (audio and image) is sent via wired or wireless networks using embedded networking software. Additional applications where adding network capabilities is desirable include control applications and sensor networks. The IP protocol stack is used to communicate by more or less all systems connected to the Internet [1,2].

S. Mubeen et. al. (2020) time requirements for many embedded systems in automobiles are very strict. As a result, while developing these systems, developers must not only control software complexity but also ensure time predictability. When all of the specified timing requirements can be shown or demonstrated at the time of design, a system is said to be timing predictable [3].

L. Lo Bello et. al. (2019) studied that recently, embedded software has been the primary enabler for cutting-edge functionality and features in automotive systems, including on-road vehicles like contemporary cars and off-road vehicles like construction, mining, forestry, and recycling cranes. The substantial growth in size and complexity of the automotive software, which makes its

DOI:10.46647/ijetms.2023.v07i03.131 ISSN: 2581-4621

development a difficult undertaking, is one of the key effects of the growing need for new software-based capabilities in these systems [4].
The connectivity requirements of very different devices vary greatly. However, in our perspective, there must be a "base level" of connectivity between objects, if not more. Even the most basic embedded sensing and computing gadgets should have access to this. We enable numerous new applications by offering just a little quantity of wireless connectivity that allows for communication. We refer to the provision of such connectivity as embedded networking.

II. EMBEDDED SOFTWARE DEVELOPMENT ENVIRONMENT

A strong embedded software development toolkit is needed to provide high-level descriptions of the algorithms executing on the processor in order to fully utilise the inherent capabilities of the optimised network processor architecture. This guarantees time to market, readable code, and simple portability [5]. The requirement that this be done without sacrificing performance [6] makes it somewhat more challenging. You can choose a mainstream processor to guarantee strong tool support. This does not, however, allow for application-specific instruction sets to achieve a greater performance outcome. A standard processor may occasionally be adopted and enhanced with particular packet-oriented instructions. In this scenario, productivity is decreased since C intrinsics or in-line assembly are required to take use of the additional instructions. A fully dedicated processor is an additional choice that offers faster packet processing. Only a basic toolkit is often offered in this situation. This has a significant effect on output.

III. EMBEDDED MOBILE NETWORKING

Embedded networking is the creation of a network that is so small and straightforward that nearly anything can use it. We hope to achieve a level of communication amongst common things through embedded networks that has not yet been accomplished. Electronic access management to buildings and highways, telephones, fax machines, photocopiers, printers, portable computers, and PDAs; environmental sensors that can monitor and control the environment; banking and public information terminals. In order to meet the rising need for real-time processing in networking applications, embedded processors are essential. Low latency and deterministic behaviour are necessary for time-sensitive operations including video streaming, voice communication, and industrial control systems. Real-time operating systems and embedded processors with specialised hardware accelerators enable the swift and accurate completion of these operations. Interoperability and smooth communication are made possible by the integration of networking protocols and standards with embedded processors. They enable devices to interact and share data across many networks by supporting a broad variety of communication protocols, such as Ethernet, Wi-Fi, Bluetooth, and cellular technologies.
As networking technologies continue to advance, embedded processors are evolving to meet the increasing demands of modern applications. They are becoming more powerful, energy-efficient, and capable of handling complex networking tasks at the edge of the network. This trend aligns with the rise of edge computing and the proliferation of IoT devices, where embedded processors are at the forefront of enabling distributed computing and decentralized networks. An embedded network prototype is being developed by the Piconet project at ORL. A low-rate, short-distance radio network known as Piconet. A Piconet node that we have created can be utilised to connect to this embedded network. A wide variety of mobile and embedded computer items can take advantage of Piconet's connectivity to and awareness of their environment. Active diaries, alerts, information points, and electronic business cards are just a few examples of Smart Information Services that are appropriate for embedded networking. These applications can be context aware thanks to the nearby connectivity that Piconet offers [7].

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.7 May - June – 2023
DOI:10.46647/ijetms.2023.v07i03.131 ISSN: 2581-4621

IV. TECHNOLOGY CHARACTERISTICS

The type of applications that we intend to enable with an embedded network like Piconet place specific demands on the technology that was utilised to create it. The network should primarily be pervasive, low-powered, and straightforward. It must have a reasonable short range so that connectivity can be used to determine closeness.

Ubiquitous and simple

A binary switch is the most basic component that a Piconet might connect. It might only occasionally transmit its condition over the wireless channel as its only function. The switch can be queried by other mobile devices connected to Piconet to learn its current state and what it denotes. Piconet must be incredibly straightforward and low-powered in order to make this practicable. The wireless medium needs to work in both indoor and outdoor settings, exposed and embedded, line-of-sight and diffuse. The communication protocols used must be very light-weight. Devices must share mechanisms for describing themselves to the outside world so that other devices can find them, comprehend them, and communicate with them.

Low-power, low-rate, low-range

At every stage of the system's design, the need for low power has an impact. We must embrace protocols that enable a device's network interface to be turned off for a large portion of the time in addition to selecting low power components. This is made simpler because we only need a low-speed connection between the devices and do not require the same amount of complexity that higher-speed networks require in their architecture.

Radio for embedded networking

In Piconet, radio technology is employed for communication. Radio has the qualities required for ad-hoc, peer-to-peer communication in almost any setup or situation. Communication must be unlimited, i.e., nodes must be able to communicate when in range, whether they are being carried in a briefcase, coat pocket, or car boot, indoors or out, in order to support our concept of interaction among Piconet nodes. Although infrared technology provides advantages over radio, including smaller component sizes, lower costs, and reduced power consumption, a large portion of this is due to infrared technology's maturity and standardisation efforts [8].

V. A NETWORK-CENTRIC APPROACH TO EMBEDDED SOFTWARE

In the technology that underpins the ongoing miniaturisation of processing and storage, the introduction of small, low-power wireless communication, sensors, and actuators is giving rise to whole new types of embedded systems and a fundamentally new genre of embedded software. In the past, embedded systems were carefully developed to perform a certain function. The new classes of embedded systems are dispersed, installed in settings where they might not have been intended to follow a specific control path, and frequently quite dynamic. Communication between groups of devices enables more advanced coordinated behaviour, which is the basic shift. Instead of being highly tailored to a specific stand-alone purpose on a device sized to suit, the new genre of embedded software is characterised by being nimble, self-organizing, critically resource restricted, and communication-centric on several small devices acting as a collective.

VI. ETHERNET-BASED IPV4 NETWORKS

When it comes to communications, there are always challenges in embedded systems. Creating one's own communications protocol is expensive and could require outside assistance. The operation of a communication stack used on Ethernet-based IPv4 networks will be covered in the parts that follow. It is connected to a Microchip ENC28J60 [11] using SPI (Serial Peripheral Interconnect) to get around the hurdle [9, 10]. Micro Internet Protocol (uIP) TCP/IP [12, 13] stack was employed in terms of software. Using the sockets library and the Perl programming language, the second issue was resolved. the prototype before this one. With the help of science and technology, the simple computer machine has become increasingly difficult to meet our needs. As a result, people from all walks of life are pursuing new types of embedded software and systems.

DOI:10.46647/ijetms.2023.v07i03.131 ISSN: 2581-4621

Examples of these systems include the TV home network, intelligent home appliances, GPS navigation systems, smart phones, MP3 players, and other digital intelligent devices. To accomplish the ideal fusion of computer hardware and software systems, certain software technology is translated into a curable component that is applied to the computer hardware system.

VII. THE GENERATION OF EMBEDDED SOFTWARE

Embedded systems are the foundation of embedded software, which is a type of computer application that is buried in the host device's microprocessor system. Embedded systems are frequently referred to as "application-centric" systems. Such device, often known as an implanted computer, is typically not of interest to computer users. It primarily consists of four components: embedded operating system, application software system, peripheral hardware devices, and embedded microprocessors. Similar to a computer's operating system, embedded software's importance cannot be overstated. Because embedded systems typically require only a single installation of all software (including the operating system and application software), the embedded operating system is actually more significant than a computer's operating system.

VIII. THE CONCEPT AND CHARACTERISTICS OF EMBEDDED SOFTWARE

The so-called embedded software is a technical development of the operating system or other software embedded in hardware that is applied to the industrial production process and is embedded in the software. This process starts with chip design and development, then moves on to re-designing embedded system software before moving on to the production of embedded electronic devices. The embedded system, which is a computer system with a standalone function and contains a main memory, microprocessors, microcontrollers, and other components, is used as a foundation and emphasises how hardware and software must work in harmony to achieve a programme job. The main characteristics of embedded software are: practicality, usability, which refers to its functional complexity needs of users, embedded software, which is integrated into the computer system, but also for the services provided by the computer system, which are closely related to the hardware, and embedded type software development, which is primarily driven by user demand, market demand, and industrial development orientation [14].

IX. THE SITUATION ANALYSIS OF EMBEDDED SOFTWARE TECHNOLOGY

There are numerous problems in the current embedded software technology development process, necessitating the use of technology that can bind a variety of development needs to innovation and change. The embedded software that has been widely employed in people's working lives, learning, and so forth, has had a rather quick rate of development in China at the moment [15]. From the perspective of applications, embedded software is widely used in communication engineering, consumer electronics, and industrial manufacturing. Smart phones and GPS are key components of communications engineering; consumer electronics, including digital television, digital cameras, home gateways, and other components; and industrial production, including CNC machine tools, manufacturing facilities, and other components.

X. THE CHALLENGES FACING THE EMBEDDED SOFTWARE TECHNOLOGY

Information appliances, which stand in for embedded devices from the Internet era, not only infused new life into the embedded industry but also created new problems for embedded systems technology, particularly software technology. Support for the expanding functional density, adaptable network connectivity, lightweight mobile applications, and processing of multimedia information are some examples of this. Of course, we also have to contend with more ferocious market rivalry. The network is now an established fashion. Equipment must have an embedded communication interface to meet external networking requirements, and this interface must be supported by TCP/IP protocol stack software. A new generation of embedded devices must have

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.7 May - June – 2023
DOI:10.46647/ijetms.2023.v07i03.131 ISSN: 2581-4621

IEEE1394, USB, CAN, Bluetooth, or IrDA communication interfaces in addition to the necessary physical layer driver software and communications networking protocol software.
Support low cost, micro-power, compact size, and small electronic device technologies. The necessary specifications for embedded device designers restrict processor performance, memory capacity, and interface chip reuse in order to support this feature. The embedded software must be increased in accordance with this design. Pick the finest Java programming paradigm and continuous improvement method, for instance. As a result, skilled software developers are required, as well as more modern tools for developing embedded software, such Java and the Web.

XI. EMBEDDED WIRELESS SENSOR NETWORKS (EWSNS)

Sensor nodes with embedded sensors that can gather information about a phenomenon make up embedded wireless sensor networks (EWSNs), which connect neighboring sensor nodes via wireless links. A wide range of sensors, including acoustic, seismic, temperature, and, more recently, image sensors and/or smart cameras, must be incorporated in the sensor nodes for many emerging EWSN applications (such as surveillance and volcano monitoring). Traditional EWSNs with scalar sensors (such as temperature and humidity sensors) transmit the majority of the sensed data to a sink node (base station node), but information-hungry applications with a variety of sensors, such as image sensors and/or smart cameras, are making this sense-transmit paradigm unworkable. Emerging applications have processing and transmission needs that are greater than what can be handled by conventional EWSNs. Consider a military EWSN deployed on a battlefield as an illustration. This system needs a variety of sensors, including optical, acoustic, and electromagnetic ones. Since it is unable to transmit high-resolution pictures and video streams from sensor nodes to the sink node over bandwidth-constrained wireless networks, this application poses a number of difficulties for existing EWSNs. Furthermore, standard EWSNs made up of single-core embedded sensor nodes are unable to process multimedia data (audio, image, and video in this case) meaningfully in real-time [19, 20], necessitating the use of more potent embedded sensor nodes to realize this application.

XII. MULTI-CORE EMBEDDED WIRELESS SENSOR NETWORK ARCHITECTURE

Our suggested heterogeneous hierarchical MCEWSN design is shown in Fig. 1, and it fits the expanding in-network processing needs of new EWSN applications. Numerous single-core embedded sensor nodes and multiple multi-core embedded sensor nodes are integrated under the umbrella of the architecture's heterogeneity. We point out that large EWSNs (EWSNs made up of a lot of sensor nodes) have homogeneous hierarchical single-core EWSNs discussed in the literature [21, 22].

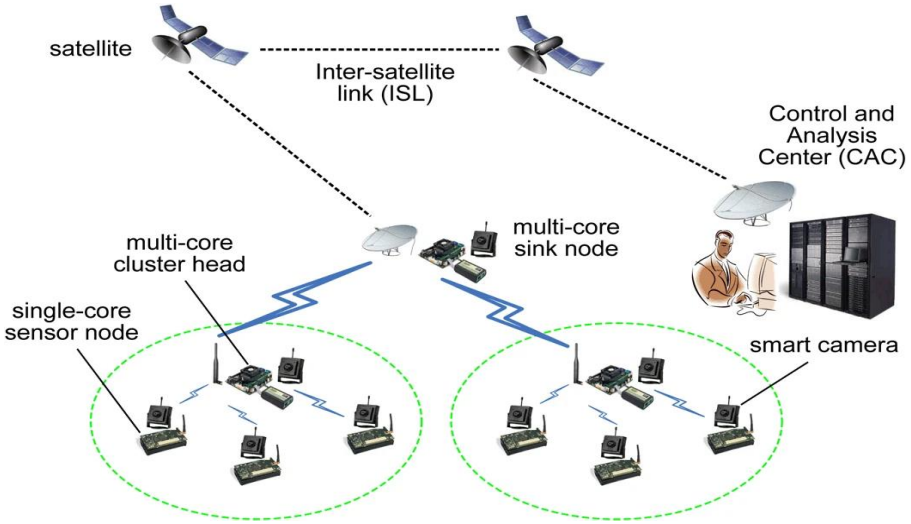

Figure 1. A heterogeneous multi-core embedded wireless sensor network (MCEWSN) architecture

DOI:10.46647/ijetms.2023.v07i03.131 ISSN: 2581-4621

A cluster head and numerous leaf sensor nodes make up each cluster. Leaf sensor nodes are in charge of sensing, pre-processing sensed data, and communicating sensed data to the cluster head nodes. They have a single core processor. The multi-core processor found in cluster head nodes is in charge of efficiently consolidating and fusing the data collected from leaf sensor nodes for transmission to the sink node. Due to bandwidth limitations, sending all the collected data from the cluster heads to the sink node is not practical for EWSNs. This necessitates complex processing and information fusion to be performed at cluster head nodes, and only the concise processed information is transmitted to the sink node [23].

The sink node has a multi-core processor and is in charge of converting high-level user requests from the control and analysis centre (CAC) into network-specific directives, requesting the required information from the MCEWSN, and providing the necessary information to the user/CAC. The multi-core CPU on the sink node makes it easier to post-process the data gathered from various cluster heads. Information fusion and event detection using network-wide data from all of the sensor nodes are part of the postprocessing done at the sink node. The CAC conducts additional analysis of the data obtained from the sink node and sends inquiries and control directives to the sink node.

XIII. APPLICATIONS

Routing and Switching

Embedded processors are utilized in routers and switches to analyze network traffic, make forwarding decisions, and manage the routing tables. They perform tasks like packet filtering, routing protocol processing (e.g., OSPF, BGP), and Quality of Service (QoS) management.

Network Security

Embedded processors are employed in network security appliances such as firewalls and intrusion detection/prevention systems. These processors handle tasks like deep packet inspection, cryptographic operations, and traffic analysis to enforce security policies and protect the network against threats.

Network Monitoring and Analysis

Embedded processors are used in network monitoring tools to capture and analyze network traffic, identify performance bottlenecks, monitor network health, and troubleshoot issues. They enable functions such as packet capture, protocol analysis, and statistical analysis of network traffic.

Wireless Networking

Embedded processors are found in wireless access points and wireless routers to manage wireless communication protocols, handle security mechanisms (e.g., encryption), and coordinate the transmission and reception of wireless signals.

Internet of Things (IoT)

In IoT applications, embedded processors are used in devices that connect to the internet, such as smart home devices, industrial sensors, and wearable devices. These processors enable data collection, communication with other devices or the cloud, and local processing of sensor data. Embedded processors used in networking applications vary in complexity and capabilities, ranging from simple microcontrollers to more powerful system-on-chip (SoC) designs incorporating multiple processing cores, memory, and specialized hardware accelerators for networking tasks. The choice of processor depends on the specific requirements of the networking application, such as throughput, latency, power consumption, and cost.

XIV. CONCLUSION

Embedded processors and networking are tightly intertwined fields that enable the efficient functioning of interconnected systems. Embedded processors provide the computational power, real-time processing capabilities, and protocol support necessary for networking devices and applications. They play a crucial role in data processing, network management, and real-time tasks, driving advancements in networking technologies across various industries.

DOI:10.46647/ijetms.2023.v07i03.131 ISSN: 2581-4621

References

[1]. Deborah Estrin (2001). Embedded Everywhere: A Research Agenda for Networked Systems of Embedded Computers, National Academy Press, ISBN 0-309-07568-8.
[2]. Gregory Pottie and William Kaiser (2005). Principles of Embedded Networked Systems Design, Cambridge University Press, ISBN 9780521840125.
[3]. S. Mubeen, E. Lisova, A.V. Feljan, Timing predictability and security in safety-critical industrial cyber-physical systems: A position paper, in: Applied Sciences–Special Issue ‘‘Emerging Paradigms and Architectures for Industry 4.0 Applications’’, Vol. 10, (3125) 2020, pp. 1–17.
[4]. L. Lo Bello, R. Mariani, S. Mubeen, S. Saponara, Recent advances and trends in on-board embedded and networked automotive systems, IEEE Trans. Ind. Inf. 15 (2) (2019).
[5]. N. Cravotta, “OC-48, OC-192 and beyond”, EDN Magazine, Nov. 9, 2000, pp. 61-68.
[6] L. Gwennap, “NPUs tease, don’t deliver”, Electronic Engineering Times Magazine, Nov. 20, 2000, p. 49.
[7]. Schilit94 Context-Aware Computing Applications. Bill N. Schilit, Norman Adams and Roy Want. IEEE Workshop on Mobile Computing Systems and Applications. December 8-9, 1994.
[8]. IrDA96 Infrared Data Association specifications, version 1.1. June 1996.
[9]. Atmel Atmega168 Datasheet.
[10] Atmel, “Atmel: Atmega168 Product Card.” http://www.atmel.com/dyn/products/Product_card.asp?part_id=3303 Web. 18 Dec. 2009.
[11] ENC28J60 Datasheet – Microchip Ethernet Controller.
[12] A. Dunkel, “uIP Main Page.” http://www.sics.se/~adam/uip/index.php/Main_Page. Web. 21. Dec. 2009.
[13] A. Dunkel, “uIP Documentation.” http://www.sics.se/~adam/uip/uip-1.0-refman/. Web. 16 Nov. 2009.
[14]. Huifang Zhou: Computer Application, Vol. 6 (2004) No 53, p.25-26.
[15] Hongli Zhang: Information Technology, Vol. 12 (2005) No 27, p.74-76.
[16] Qin Guo: Computer and Network, Vol. 1 (2006) No 33, p.11-14.
[17] Jieming Liu: Guangxi Normal University, Vol. 3 (2007) No33, p.121-124.
[18] Qiliang Hu: Guangxi Sciences Academy, Vol.21 (2010) No18, p.60-64.
[19]. M. Hamdi, N. Boudriga, and M. Obaidat, “Bandwidth-Effecitve Design of a Satellite-Based Hybrid Wireless Sensor Network for Mobile Target Detection and Tracking,” IEEE Systems Journal, vol. 2, no. 1, pp. 74–82, March 2008.
[20] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, “Wireless Multimedia Sensor Networks: Applications and Testbeds,” Proceedings of the IEEE, vol. 96, no. 10, pp. 1588–1605, October 2008.
[21]. S. Keckler, K. Olukotun, and H. Hofstee, Multicore Processors and Systems. Springer, 2009.
[22] TILERA, “TILEPro Processor Family,” March 2013. [Online]. Available: http://www.tilera.com/products/processors/ TILEPro Family.
[23]. A. Y. Dogan, D. Atienza, A. Burg, I. Loi, and L. Benini, “Power/Performance Exploration of Single-Core and Multi-core Processor Approaches for Biomedical Signal Processing,” in Proc. of the Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), Madrid, Spain, September 2011.