Quality of Service in Embedded Networks

Elena Suvorova, Saint-Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia
Yuriy Sheynin, Saint-Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia
Nadezhda Matveeva, Saint-Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia

ABSTRACT

Embedded systems are widely used in many fields, from space industry to medicine. In this paper we consider Quality of Service (QoS) in embedded networks. Different QoS are analyzed. The authors consider three structures and implementations of the network layer for providing QoS, compare their implementation characteristics and evaluate hardware costs. They consider QoS mechanisms support in modern space network protocols, possibility of using them in embedded networks. Hardware costs are one of the main constraints for embedded networks. Therefore hardware costs of basic routers with these QoS mechanisms are compared.

Keywords: Embedded Network, Embedded System, GigaSpaceWire, Quality of Service (QoS), SpaceFibre, SpaceWire

INTRODUCTION

The technology progress addresses to integrate dozens to hundreds of Intellectual Property (IP) blocks, which are basic components on a chip, like, RAM, DSP, etc., within a single chip. Embedded systems are designed to perform dedicated specific tasks with real-time processing constraints. Such systems comprise complete devices ranging from portable systems to space on board systems. Also embedded systems are widely used in space industry. Satellites and different space systems consist of different sensors, computation modules, displays, control systems and etc. Systems are large and very complex (Balandina, 2014). Requirements for their data transmission speed, throughput spacecraft onboard constantly change. Two additional characteristics are very common in embedded systems: reactive and heavily constrained operation. Most embedded system interact directly with processes or the environment, making decisions on the fly, based on their inputs. It makes necessary for the system to be reactive, responding in real-time to process inputs to ensure proper operation. Besides, these systems operate in constrained environments where memory, computing power, and power supply are limited. Power consumption and area are very important characteristics for space industry too (Ganry, 2014).

DOI: 10.4018/IJERTCS.2014100104
Many applications require guaranteed delivery of transmitted data. Multitasking distributed nature of onboard applications require reliable control of their processing and communications priorities. Same demand one have in embedded network for mixed-critical systems. In real-time closed-loop control guaranteed control data delivery time is required. Typical for embedded systems mixture of different traffic types with high rate data sources and critical command traffic demands controlled network throughput distribution between different information flows. QoS requirements such as priorities, packet delivery and packet delivery time are important and critical for embedded networks. For example, command traffic is very sensitive to high delay, besides its delivery should be provided by space system (Sheynin, 2014).

Performance and operating characteristics of communication systems are crucial. Different methods of performance evaluation can be used as the structure of the system is designed. Systems modeling techniques or analytical methods for calculating the characteristics relate to it. In this article we consider main QoS characteristics, describe different proposals for architecture network layer implementation, analyze their hardware costs and compare QoS mechanisms of modern space onboard protocols from SpaceWire family protocols. SpaceWire protocols are used to drive the development of CCSDS Spacecraft Onboard Interface Services (SOIS) Area (CCSDS 850.0-G-2, 2013).

EMBEDDED SYSTEM AND EMBEDDED NETWORK

History of embedded system begins in 1960s. One of the earliest electronic computing devices credited with the term “embedded system” and closer to our present conception of such was the Apollo Guidance Computer (AGC), (Jiménez, 2014). AGC was part of the guidance and navigation system used by NASA in the Apollo program for various spaceships. Embedded system could be defined as a processor based system designed to perform a few dedicated functions, often in real-time, (Forrai, 2013). Embedded systems are used in complex devices such as mobile gadgets, automobiles, household appliances, satellites and etc. These systems consist of different component. For example, it could be Digital Signal Processor (DSP), Central Processing Unit (CPU), the system memory, a set of input-output ports, peripheral devices.

Regardless of the function performed by an embedded system, the broadest view of its structure reveals two major, tightly coupled sets of components: a set of hardware components and a series of software programs (Figure 1), (Jiménez, 2014). Hardware components include processing units, memories and etc. Software is typically firmware. Firmware is a computer program typically stored in a non-volatile memory embedded in a hardware device. It is tightly coupled to the hardware where it resides and though it can be upgradeable in some applications, it is not intended to be changed by users, (Jiménez, 2014). Software gives functionality to the hardware. System inputs/outputs are required to interact with user or other subsystems within the application. Keys and buttons, sensors, light emitting diodes (LEDs), liquid crystal displays (LCDs) are examples of user interfaces.

There are three categories that can be used to classify embedded systems in general, (Jiménez, 2014): Small, Distributed, and High-performance. The relationships among these classes are shown in Figure 2.

Small Embedded Systems is typically centered around a single microcontroller chip that commands the whole application. These systems are highly integrated, adding only a few analog components, sensors, actuators, and user-interface, as needed. These systems operate with minimal or no maintenance. Their cost is very low. They are produced in mass quantities. Software in these systems is typically single-tasked and rarely requires an RTOS. Examples of these systems
Adaptive Routing for an Ad Hoc Network Based on Reinforcement Learning
[www.igi-global.com/article/adaptive-routing-for-an-ad-hoc-network-based-on-reinforcement-learning/158965?camid=4v1a](www.igi-global.com/article/adaptive-routing-for-an-ad-hoc-network-based-on-reinforcement-learning/158965?camid=4v1a)

Centrality-Based Connected Dominating Sets for Complex Network Graphs
[www.igi-global.com/article/centrality-based-connected-dominating-sets-for-complex-network-graphs/124036?camid=4v1a](www.igi-global.com/article/centrality-based-connected-dominating-sets-for-complex-network-graphs/124036?camid=4v1a)