Networked Embedded Systems: a Quantitative Performance Comparison

Alessio Botta,Walter de Donato, Antonio Pescapè, and Giorgio Ventre
University of Napoli "Federico II" (Italy)
{a.botta,walter.dedonato,pescape,giorgio}@unina.it

Abstract— **Networked embedded systems are gaining more and more attention and their use in current network scenarios is of indisputable importance. Research community and industry are proposing novel embedded solutions, often based on network processors, for network connectivity, data processing and service delivery. Despite this, quantitative performance comparisons of such systems seem to be very hard to find. In this paper, we describe an experimental analysis of different boards for networked embedded systems using both general-purpose and network processors, and running both commercial and open source operating systems. The results show that network-processor based boards are able to attain very high performance when compared to boards based on x86 processors, especially when running commercial operating systems. The analysis provides a reference for the design, development, and testing of novel networked embedded systems.**

I. INTRODUCTION

The growing complexity and heterogeneity of the Internet architecture is triggered by the constant deployment of novel applications and the provisioning of innovative services to the users. This increases the dynamic behavior of such infrastructure in which the embedded systems used for switching, routing, and connecting devices have to possess strong adaptation capabilities. Often high-speed Networked Embedded Systems (NES) are based on the well known Application-Specific Integrated Circuits (ASIC). Being tight to a specific application, NES based on such processors attain high performance at the cost of the flexibility. Thanks to their speed, the ASIC are typically used for line-speed packet processing applications such as packet inspection. But, when something changes e.g. packet headers, the systems can not be easily upgraded and sometimes have to be physically changed. In contrast, general purpose processors provide a great flexibility but they are not suitable to implement NES for such applications at current line speed. To bridge the gap between these technologies, few years ago many prominent vendors have started thinking of a new generation of processors for NES able to run at very high speed and to be easily programmed: the *network processors*. In a short time, several NES using network processors have been developed gaining the interest of both industry and research community.

In this paper we are not interested in ASIC-based NES because we believe they are not able to fullfill the requirements of current and future network scenarios. Therefore we study the behavior of some architectures for NES based on general-purpose processors and on network processors. In particular, we quantitatively evaluate the performance of two boards based on the Intel network processors that belong to the IXP4XX family [1]. They are cheap processors intended for use in small routers with advanced features (e.g. encryption, etc.), powerful IEEE 802.11 access points, etc.. We study the behavior of these processors with different operating systems both commercial (i.e. *Montavista Linux* [2]) and open source (i.e. *Snapgear* [3] and *OpenWRT* [4]). Moreover, we show their performance when operating on both an experimental board called *StarEast* [5] and in a real operational access point for IEEE 802.11 networks produced by *Netgear*. Furthermore, we compare the results with those achieved by a board for NES targeted to the same class of applications but based on a general purpose processor: the *Soekris Net4826* [6] based on the AMD Geode. For the analysis, we use different traffic patterns (obtained by opportunely combining Inter Departure Times and Packet Sizes) generated by using a well known traffic generator called D-ITG [7] which has been purposely ported on the Intel IXP4XX architecture. We describe the problems faced in the porting, underlining the peculiar characteristics of such architecture. Thanks to the use of different boards, operating systems and traffic profiles we provide a complete sketch of what a common application would experiment when running on architectures for networked embedded systems.

This paper is organized as follows. Section II describes the motivations at the base of our work, shortly illustrating the framework in which we place our research. Section III provides an overview of the considered Networked Embedded Systems (NES). In Section IV we describe the work we have done for using D-ITG over the IXP4XX-based NES, whereas in Section V results of this preliminary work are presented and discussed. Section VI ends the paper with some concluding remarks.

II. MOTIVATION AND RELATED WORK

In this paper we consider NES boards based on both the Intel IXP4XX network processors and the AMD Geode general purpose processor. The IXP4XX are widely utilized by several manufacturers for a wide range of commercial NES, such as IEEE 802.11 access points (e.g. Netgear WG302 [8]), access routers (e.g. D-Link DRO-250i [9]), network storage appliances (e.g. Linksys NSLU2 [10]), firewall/VPN devices

---

[0]This work has been done thanks to Intel Corporation that kindly donated two network processor boards equipped with Intel IXP425.
This work has been partially supported by PRIN 2007 RECIPE Project and by CONTENT EU NoE.

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2008 proceedings.

(e.g. USRobotics USR8200 [11]), and so forth. However, the high number of devices that have been sold has triggered the spreading of user communities aimed at experimenting operating systems other than those provided by the vendors, and applications also very far from the original design. At a first approximation, the works in literature concerning network processor performance can be divided in two main classes. The first class is characterized by the theoretical or simulation studies of the performance of such processors in their general abstraction [12], [13], [14]. The other class, instead, comprises works that are focused on specific applications such as packet forwarding or classification engines. These studies often present a real implementation and its performance [15], [16], [17].

The AMD Geode is targeted to the same class of applications and it is therefore used in a variety of NES which include also firewalls, Asterisk servers, and so forth. However, as a big difference from the IXP4XX, the AMD Geode belongs to the well known and studied x86 family. This means that a lot of existing applications need not to be changed in order to run on this architecture. And, it also implies that most of the work in literature is concerned with the implementation and evaluation of specific applications [18], [19], [20], rather then the study of the theoretical processor performance.

With the aim to provide a realistic experimental performance comparison, we ported an application based on the Berkely Sockets on the ARM processor present inside the IXP4XX (called Intel XScale), and we used the Network Interface Card drivers provided by the operating systems. We believe that even if this may not be the best way to obtain the highest performance from our application, it is surely the most appropriate to observe what the real applications experiment. In addition, using the standard Berkeley Sockets allows to perform a fair comparison with the NES based on general purpose processors. In this paper we provide the following contributions: (i) we study the behavior of NES boards based on both network and general-purpose processors with different operating systems and a Socket-based application for packet generation; (ii) we explain the problems encountered when porting such application on the Intel IXP4XX architecture; (iii) we sketch a reference for the performance of real applications on NES. It is also worth noting that we provide the application we used for the tests publicly available on the web [7].

III. CONSIDERED NETWORKED EMBEDDED SYSTEMS

This section provides an overview of the NES boards and the operating systems we used for our analysis, mainly with the purpose to guarantee the repeatability of our experimentation.

A. *Network processor equipped boards*

We considered two boards for NES based on a peculiar family of network processors, the Intel IXP4XX. Like other network processors, the IXP4XX contain a general-purpose central processor, dedicated to the control plane, and some satellite processors, dedicated to the data plane. As a general purpose-processors they use an ARM called XScale. Instead, as satellite processors they feature a varying number of RISC named Network Processor Engines (NPE). However, as a difference with higher class network processor, the satellite processors are not fully programmable. To exploit them, the Intel provides the developers with some precompiled micro-codes (which include also Ethernet interface handling), and specific APIs to interact with the micro-codes.

1) Stareast: StarEast is a stackable system with three kinds of modified PCI Mezzanine Cards. One is a baseboard, and the other two are adapter daughter cards to provide miniPCI and CardBus interfaces. The baseboard is equipped with a 533MHz Intel IXP425 network processor and provides two fast Ethernet ports, one UART, and two interfaces to connect the daughter cards. Moreover, the baseboard includes 133MHz, 256 MBytes of SDRAM, and 32MBytes of Intel StrataFlash memory. For our analysis we installed on it two different Linux distributions: Montavista Linux (version 4.0 based on kernel 2.6.10) and Snapgear (both version 3.1, based on kernel 2.4.24, and version 3.3, based on kernel 2.6.12). MontaVista is the undisputed leading provider of commercial Linux development platforms for intelligent devices and communication infrastructures. Its linux distribution supports a large variety of embedded systems and provides a commercial-grade development environment giving the ability to achieve rapid time to market. SnapGear Linux is an open-source distribution designed for deeply embedded microprocessors with or without memory management units. It supports more than 100 chip architectures and runs in more than 20 million devices globally. More details on the configuration of Stareast can be found at [21].

2) Netgear WAG302: This business-class Access Point supports both 5 GHz and 2.4 GHz 54 Mbps radio transmission with up to 108 Mbps in turbo mode. It is based on a 266 MHz Intel IXP422B processor, and equipped with 32 MBytes DRAM memory, 16 MBytes Flash memory, one 10/100 Mbps Ethernet port, and two miniPCI Atheros wireless interfaces. It comes with a customized Montavista Linux (version 3.0) distribution which limits its usage, but can be replaced exploiting *redboot* console during the bootstrap process. For our analysis we installed on it an OpenWRT Linux distribution (revision 10215) based on kernel 2.6.23.

B. *Board equipped with general-purpose processor*

We considered a wide spread board for NES based on the AMD Geode processor and produced by Soekris Engineering: the Soekris net4826-50. This compact, low-power, low-cost, advanced communication computer is based on a x86-compatible 266MHz AMD Geode processor. It has one 10/100 Mbps Ethernet port, 128 MBytes SDRAM main memory, and 256 MBytes CompactFlash memory for programs and data storage. This platform, as reported on many forums, is widely used to create fully customized routers and access points. For our analysis we installed on it a Debian derived distribution, called Voyage Linux, specialized to run on x86-based embedded platforms. We adopted the 0.4.1 stable version based on kernel 2.6.19.

IV. GENERATING TRAFFIC OVER NES

To study the behavior of considered NES we chose a widely used traffic generator able to both generate realistic traffic patterns and collect statistic of Quality of Service parameters like throughput, jitter, packet loss and delay.

D-ITG has been originally developed for x86 architectures. Therefore we could easily use it on the Soekris board. Instead, for the IXP4XX based boards, some issues had to be solved. The IXP4XX, indeed, contains an ARM processors which features different data alignment and *endianness*, as explained in the following.

The first issue we encountered is related to the *endianness*: computer architectures may adopt either a little-endian (e.g. the x86) or a big-endian scheme for storing numbers in memory, and some architectures (e.g. the ARM) can also use both. D-ITG components were unable to communicate correctly when running on architectures using a different endianness scheme[1]. And, a common format had to be used for representing the information exchanged between different architectures. To solve this problem, we modified the representation of the signaling messages exchanged among ITGSend, ITGRecv, and ITGLog components. We used the conversion functions of the Berkley Socket library, and adopted the network format for all the information exchanges.

The second issue is related to the alignment of the data structures in memory adopted by different architectures. In general, compilers force variables to be aligned in memory to multiples of a fixed number (e.g. 4) of bytes; this is necessary to be complaint to restrictions imposed by the underlying architecture. As a consequence, a structure may be stored in memory using some padding bytes to force this alignment. This can cause problems when such structure has to be read on an architecture and write on another one because its size may be altered. To solve this problem, we forced all the structures to be always aligned at 4 bytes by adding padding bytes into them.

As an outcome of this work, we made publicly available on the web the D-ITG version able to run on the IXP4XX and to interoperate with x86 architectures [7]. This provides the research community with the possibility to deploy a traffic generation and measurement platform on different embedded systems at the base of spread used devices (see Section II). We believe this is important because performing network measurements by using such devices (e.g. access points, ADSL modems, ...) as end points, allows to set up real heterogeneous measurement scenarios [22].

V. EXPERIMENTAL ANALYSIS

In this Section we present the network scenario we setup for the experimentation, some preliminary results we obtained, and a discussion of the lessons we learned.

A. Experimental Scenario

Our testbed is depicted in Figure 1. It is very simple because we are interested in observing the performance of the appliances, therefore avoiding other causes of uncertainty (network devices, hop counts, ...). The NES boards are connected back-to-back with the workstation which means that, in each experiment, only one board is active.

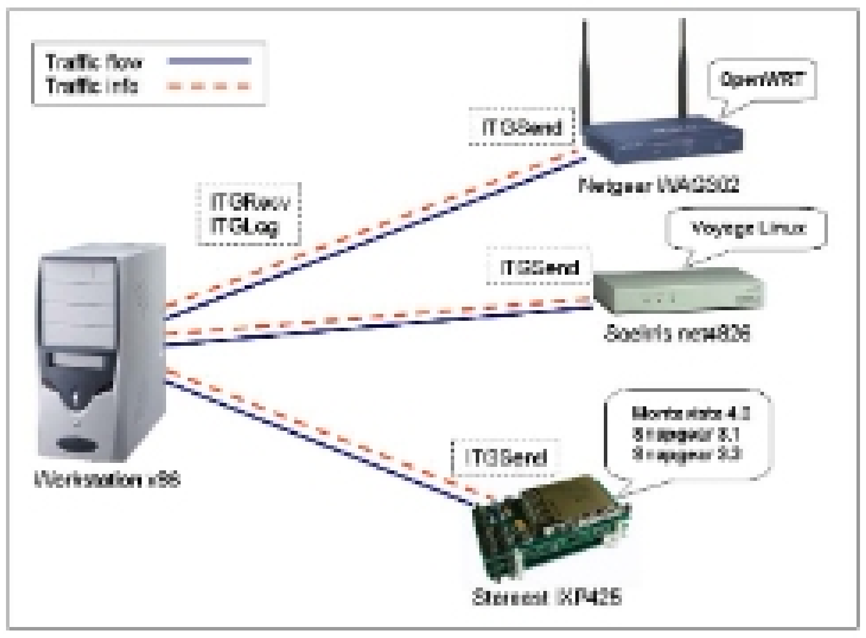

Fig. 1. Experimental scenario.

With TCP and UDP, we performed two different kinds of tests: one aimed to discover the maximum number of packets per second the devices are able to generate; another one aimed to measure the bitrate, the jitter, and the packet loss with different packet sizes and rates. In the first case we generated packets with the smallest size allowed by D-ITG (i.e. with only 16 Bytes of UDP/TCP payload). In the second type of experiments, we generated traffic with bitrate values ranging from 1 to 100 Mbps using 3 packet sizes. For each packet size and bitrate value, the corresponding packet rate is reported in Table I.

TABLE I

PACKET RATES AS A FUNCTION OF THE BIT RATE AND THE PACKET SIZE.

Bitrate [Mbps]	Packet rate [pps]			
	Only UDP PS = 1472 [Bytes]	Only TCP PS = 1460 [Bytes]	TCP & UDP PS = 512 [Bytes]	TCP & UDP PS = 64 [Bytes]
1	85	86	244	1953
5	425	428	1221	9766
10	849	856	2441	19531
20	1698	1712	4883	39062
50	4246	4281	12207	97656
100	8492	8562	24414	195312

B. Results

1) Packet rate: In Figure 2 we report the packet rate we obtained with the different boards and with both TCP and UDP. Two important things are immediately clear from these pictures. The first is that the Stareast board achieves the best performance, and the second is that, with such board, Montavista obtains a higher packet rate than Snapgear. Both are verified with TCP and UDP. However, while the first is somehow expected due to the fact that the Stareast has more computational power than the other boards, it is not equally straightforward that the commercial operating system is faster

---

[1] With the considered operating systems the XScale processor behaves like a *big endian* even if it could also operate as a *little endian*

978-1-4244-2324-8/08/$25.00 © 2008 IEEE.