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

(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.

---

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

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.
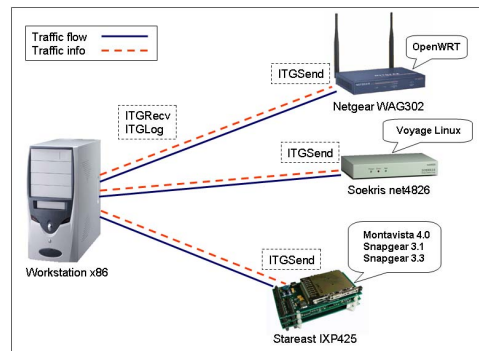

Fig. 1.   Experimental scenario.

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

than the open source solution. As a justification, Snapgear is a younger project with respect to Montavista and it has still a gap to bridge. Another interesting consideration is that, regardless of the transport protocol, the older version of Snapgear (i.e. the 3.1), based on a 2.4 kernel, performs better than the newer one (i.e. the 3.3), based on a 2.6 kernel.
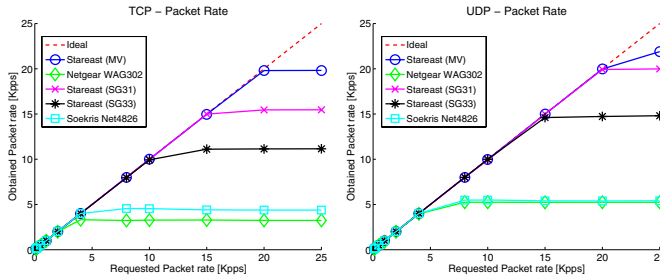

Fig. 2.   Packet rates comparison. TCP left and UDP right.

For better understanding the obtained results, we report in Table II and III the values related to Figure 2. These tables allow to precisely pinpoint the maximum rates obtained by the boards. Furthermore, for having a reference from higher-class processors, we also report the values obtained with a workstation using a P4@3.6GHz and 2GBytes RAM, and running Linux. As we can see from Table II, with TCP the Stareast is able to generate up to about 19.8 Kpps with Montavista, up to about 15.5 Kpps with Snapgear 3.1, and up to 11.1 Kpps with Snapgear 3.3. This table also shows that the Soekris board generates more packets per second than the Netgear. With UDP, instead, the Netgear achieves almost the same performance as the Soekris (5243 pps and 5410 pps respectively), as shown in Table III. An increase of the performance with UDP is also noticed for the Stareast which is able to generate up to about 21.9 Kpps with Montavista, 20.0 Kpps with Snapgear 3.1, and 14.8 Kpps with Snapgear 3.3.
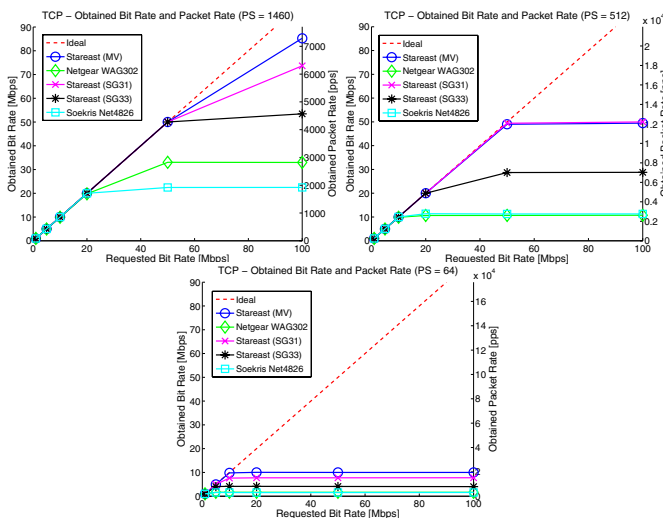

Fig. 3.   TCP bitrate comparison using three packet sizes.

2) Bitrate: In Figure 3 and 4 we report the bitrate obtained with the different boards and with TCP and UDP respectively. Such figures also show the packet rate corresponding to the obtained bitrate. The trend of the performance is the same of that revealed from the analysis of the packet rate. Indeed, also in this case, the best performance is obtained by the Snapgear with Montavista. However, with packet sizes of 1472 and 512 Bytes, the difference with Snapgear 3.1 is not as high as before. This is due to the fact that with these sizes, the requested packet rate is lower than the maximum one achievable by the board, and witnesses that the size of the packets has a small impact on the performance. With packets having a size of 64 Bytes we observe another time the same results of the packet rate analysis, with Montavista having much higher performance than the others.
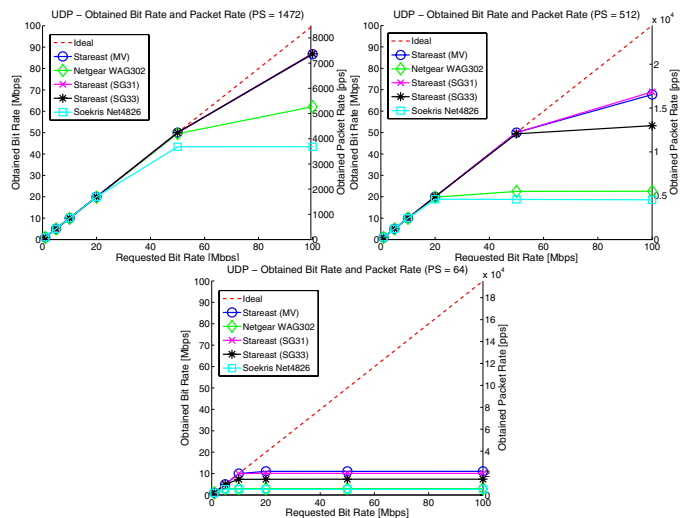

Fig. 4.   UDP bitrate comparison using three packet sizes.

Comparing Figure 3 and 4 we can observe the impact of the mechanisms implemented by TCP to provide its well known services such as reliability, ordered delivery, etc.. With UDP, the Stareast is able to nearly saturate the 100 Mbps link, with both Montavista and Snapgear 3.1, and packet size of 1472 Bytes. With TCP, instead, all but the Starteast with Montavista experiment a noticeable performance degradation. Moreover the performance of the Netgear are equal to or higher than those of the Soekris, especially with high packet size. This means that the architecture of the IXP processors is able to achieve a higher bitrate than traditional x86 processors with the same clock frequency, even if such processors have more RAM available. Moreover, the pipelining achieved thanks to the NPE of the IXP425 provides a performance gain which is higher for higher packet sizes.

3) Jitter: Figure 5 and 6 show the jitter measured with TCP and UDP respectively, as a function of the imposed bitrate. The trend of the curves is similar for both protocols. Moreover, the Stareast shows a singular behavior: with packet sizes of 1472 and 512 Bytes, the jitter observed with that board starts from a high value after which it first decreases and then increases. As

TABLE II
TCP PACKET RATES COMPARISON.

Imposed rate [pps]	100	200	500	1000	2000	4000	8000	10000	15000	20000	25000	80000
Stareast (MV)	100	200	500	1000	2000	4000	8000	10000	14969	19802	19822	19771
Netgear WAG302	99	198	495	990	1980	3314	3226	3276	3280	3240	3230	3190
Stareast (SG31)	100	200	500	1000	2000	4000	7995	9974	14988	15453	15465	15422
Stareast (SG33)	100	200	500	1000	2000	4000	7955	9934	11111	11142	11152	11087
Soekris Net4826	100	200	500	1000	2000	4000	4550	4542	4407	4383	4376	4331
Workstation x86	100	200	500	1000	2000	4000	8000	9998	14996	19995	24994	79290

TABLE III
UDP PACKET RATES COMPARISON.

Imposed rate [pps]	100	200	500	1000	2000	4000	8000	10000	15000	20000	25000	80000
Stareast (MV)	100	200	500	1000	2000	4000	8000	9991	14988	19975	21897	21843
Netgear WAG302	99	198	495	874	1977	3961	5249	5246	5241	5243	5243	5199
Stareast (SG31)	100	200	500	1000	2000	4000	8000	9897	14991	19922	19989	19943
Stareast (SG33)	100	200	500	1000	2000	4000	8000	9992	14611	14730	14812	14772
Soekris Net4826	100	200	500	1000	2000	4000	5480	5495	5402	5391	5411	5381
Workstation x86	100	200	500	1000	2000	4000	8000	9991	14985	19998	24925	79679

Fig. 5.   TCP jitter comparison using three packet sizes.

less than 0.03%. This means that, when the boards generate packets at their maximum bitrate, nearly all the packets are successfully received. A higher value of packet loss is noticed with lower packet size. This is due to the fact the packet rate is higher and, as a consequence, a small part of the packets is dropped by the internal buffers. However, with all the boards and the operating systems the packet loss is always less then 3%, even at very high speed, which means that such boards are suitable for real time network applications such as voice/video communications or networked games. This is an important consideration because the IXP4XX provide integrated functions for Digital Signal Processing that have been explicitly designed for voice and video processing.

C. Discussion

In this section we briefly discuss some lessons we learned in this work. We believe that these considerations can be useful to design, deploy, and evaluate novel networked embedded systems. The results presented in previous sections allow to evaluate the behavior of the applications when running over such architectures in terms of packet rate, bitrate, jitter, and packet loss. For example, when designing a NES that has to cope with real time traffic, the jitter values reported in Figures 5 and 6 have to be considered as the part of the total jitter due to the NES hardware and software components. Moreover, we have observed that the Stareast, which is based on the Intel IXP425, attains the highest performance. And, we have verified that the open-source operating systems we considered are still not enough mature to compete with commercial solution. This is probably one of the motivations which pushes the big vendors to opt for commercial operating systems such as Montavista Linux.

The results of this analysis have also triggered our choice for the access points of the Magnets backbone [23]. Such infrastructure has been indeed deployed using access points equipped with Intel IXP4XX network processors and proprietary operating systems. Furthermore, we believe that the availability of a traffic generation and measurement platform for Intel IXP4XX network processor, provided by this work,

a difference, with UDP and PS equal to 1472 Bytes the jitter starts from a lower value, reaches a peak, and then behaves like in the other cases. This behavior becomes more clear if we look at Table I, in which the packet rate corresponding to the analyzed bitrate is reported. Looking at this table we can observe that the packet rate for which the jitter is high is always around 1 Kpps. This means that the Stareast introduces some jitter when generating packets at such rate.

For all the boards, comparing the results obtained with UDP and TCP, we observe the higher jitter introduced by TCP. This effect is exacerbated in the case of packet size equal to 1472 Bytes because the network is working at the highest speed. In this case the jitter reaches values up to 250 μs with the Stareast and Montavista. However, except from this particular case, we can observe that all the boards do not introduce a large jitter neither when generating at full speed and can be therefore suitable for real time network applications.

4) Packet loss: Figure 7 shows the losses experimented in our tests with UDP. When using a packet size equal to 1472 Bytes, the losses are neglectable as the are always
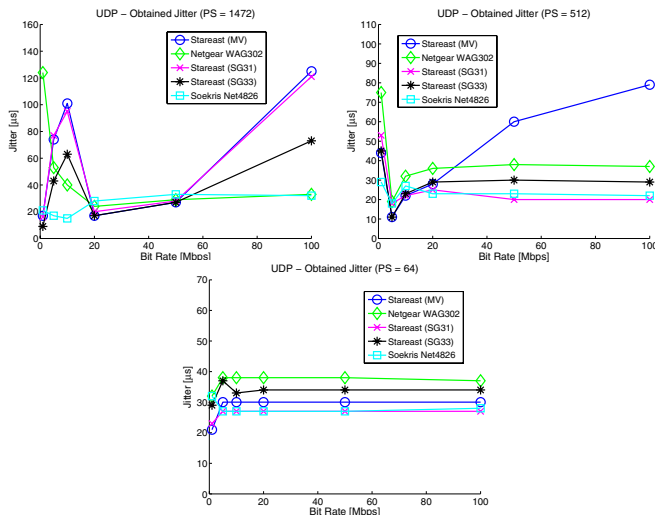
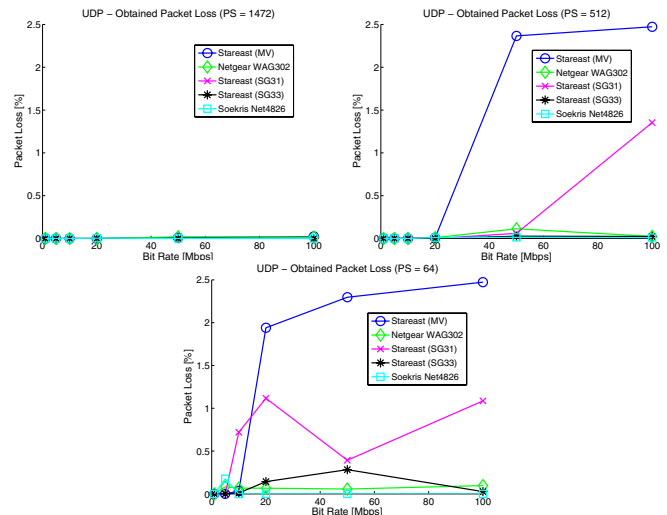Fig. 6.   UDP jitter comparison using three packet sizes.

Fig. 7.   UDP packet loss comparison using three packet sizes.

will stimulate further research in this field.

VI. Conclusion

In this paper we have analyzed different existing solutions for networked embedded systems. Beside the *classical* x86 processors, some boards based on Intel IXP4XX network processors have also been analyzed. Currently, such processors are used for several kinds of very spread NES, such as WiFi access points, ADSL modems, VPN routers and so forth. In this work we have shown the basic problems encountered in porting an application based on Berkeley Sockets on such architectures. The considered application is a traffic-generation and measurement tool, and its porting allowed us to experimentally evaluate the performance of such devices. We showed that the IXP425 achieves the highest packet rate and bitrate with respect to the other processors, especially when running a commercial operating system called Montavista Linux. We attributed this behavior to the fact the the open-source operating systems we considered are still not as mature as the commercial one. However, they are already able to achieve very high performance. We believe that this analysis can be useful to design, deploy, and test novel NES and that releasing the traffic generator that runs on IXP4XX will stimulate further research on this topic. Our ongoing work is mainly concerned with performing the same analysis with a broader range of devices as well as with the testing of novel applications (e.g video streaming) and novel protocols (e.g. DCCP and SCTP) on such devices.

References

[1] http://www.intel.com/design/network/products/
     npfamily/ixp4xx.htm
[2] http://www.mvista.com
[3] http://www.snapgear.org
[4] http://openwrt.org
[5] http://www.flexcomm.com.cn/products/
     StarEastMeshNetwork.htm
[6] http://www.soekris.com/net4826.htm
[7] http://www.grid.unina.it/Traffic/
[8] http://www.netgear.com/Products/
     APsWirelessControllers/AccessPoints/WG302.aspx
[9] http://www.dlink.co.in/Upload/UploadDownload/
     DRO-250i.pdf
[10] http://www.linksys.com/servlet/Satellite?
     childpagename=US%2FLayout&packedargs=c%3DL_
     Product_C2%26cid%3D1118334819312&pagename=
     Linksys%2FCommon%2FVisitorWrapper
[11] http://www.usr.com/products/networking/
     networking-product.asp?sku=USR8200
[12] Wen Xu; Peterson, L., "Support for software performance tuning on network processors", IEEE Network, vol.17(4), pp.40-45, July-Aug. 2003.
[13] Haas, R. et al., "Creating advanced functions on network processors: experience and perspectives", IEEE Network, vol.17(4), pp.46-54, July-Aug. 2003.
[14] Chakraborty, S.; Knzli, S.; Thiele, L.; Herkersdorf, A.; Sagmeister, P., "Performance evaluation of network processor architectures: combining simulation with analytical estimation", Comput. Networks Vol. 41(5), pp.641-665, Apr. 2003.
[15] Spalink, T.; Karlin, S.; Peterson, L.; Gottlieb, Y., "Building a robust software-based router using network processors", Symposium on Operating Systems Principles (SOSP), 2001.
[16] Srinivasan, D.; Wu-chang Feng., "Performance analysis of multi-dimensional packet classification on programmable network processors", International Conference on Local Computer Networks, pp. 360-367, 16-18 Nov. 2004.
[17] Giordano, S.; Procissi, G.; Rossi, F.; Vitucci, F., "Design of a Multi-Dimensional Packet Classifier for Network Processors", ICC '06, vol.2, pp.503-508, June 2006.
[18] Bhaskaran Raman; Kameswari Chebrolu, "Experiences in using WiFi for rural internet in India", IEEE Communications Magazine, vol.45, no.1, pp.104-110, Jan. 2007.
[19] Kyasanur, P.; Jungmin So; Chereddi, C.; Vaidya, N.H., "Multichannel mesh networks: challenges and protocols" IEEE Wireless Communications, vol.13(2), pp. 30-36, April 2006.
[20] Eriksson, J.; Faloutsos, M.; Krishnamurthy, Srikanth V., "Routing amid Colluding Attackers", IEEE International Conference Network Protocols 2007, pp.184-193, 16-19 Oct. 2007
[21] Botta A.; Pescapé A.,"How to set up and configure StarEast in Access Point Mode with Intel 2200 Wireless Network Interfaces", TR-DIS-NP-2-2007, Technical Report Dipartimento di Informatica e Sistemistica, Università di Napoli "Federico II".
[22] Botta, A.; Dainotti, A.; Pescapé, A., "Multi-protocol and multi-platform traffic generation and measurement", INFOCOM 2007 DEMO Session, May 2007, Anchorage (Alaska), USA.
[23] Karrer, R.; Matyasovszki, I.; Botta, A.; Pescapé, A., "MagNets - experiences from deploying a joint research-operational next-generation wireless access network testbed", TRIDENTCOM 2007, Orlando, Florida (USA), May 2007.