Proceedings of the 6th Győr Symposium and 3rd Hungarian-Polish
and 1st Hungarian-Romanian Joint Conference on Computational Intelligence

Packet Processing for Streaming Filtering on ATCA Platforms

D. Robu[1], C. Costache[1], A. Balica[1], T. Balan[1], F. Sandu[1]

[1]Transilvania" University, Electronics and Computers Department
BdEroilor nr 29A, 500036 Brasov, Romania
E-mail: robu.dan@unitbv.ro

Abstract: The paper presents the hardware and software integration of packet processing on an Advanced Telecom and Computing Architecture. The Continuous Computing PP50 sub-system is introduced, with its multi-processor architecture, various firmware and management solutions. With the purpose of flow discrimination by deep packet inspection, it is presented the development environment, software configuration and application management in a use-case for DSTP filtering.

Keywords: ATCA, packet processing, DPI, RMI-OS, Linux

1. Introduction

ATCA (Advanced Telecom and Computing Architecture [1]) represents a high-tech modern platform with a standard shelf and a back-plane ("mother-board") compliant with the regulations of PIGMG (PCI Industrial Computers Manufactures Group) [2]. The mother-board has a Switching Fabric behind (an "InfiniBand" in our platform), to be used together (with an intimate *common* "mesh" of direct links), by *computing, telecom* or *instrumentation* modules ("blades") taking benefit of the very large bandwidth (40G in our case) for local transfers [3], [4].

In the present paper, we present the structure and the integration of ATCA platform with great scalability and versatility for telecommunications. The focus is on the advanced Packet Processing capabilities, with details on the specific build environment, configuration, development flow and practical issues of DPI (Deep Packet Inspection) research.

Deep packet inspection in real time will play a vital role for the mobile operators and add value to the transmission medium. This will enable the services providers to offer more mobile applications that, in turn, will use this transmission medium. In addition, the traffic is expected to grow even more due to Machine to Machine (M2M) communication in the new Internet of Things (IoT), bringing Quality of Service (QoS) and Data Analytics (DA) on the front page.

2. ATCA Platform Integration

2.1. The workbench laboratory platform

The *apparatus* of our research is centered on the actual ATCA rack of our laboratory [5], manufactured by Radisys/Continuous Computing. It is of the type SH61 40G, having 6 front module slots (and 6 in the back, accessible via a RTM - Rear Transition Modules). There are 3 redundant power supplies.

The available blades (that can be reconfigured and rescaled at anytime, to meet various application requirements) are (see Figure 1): the FlexCore ATCA-FM40 – Ethernet switching boards, ATCA-XE80 computing boards, ATCA-PP50 – packet processing boards and ATCA-9100- TI Digital Signal Processing boards.

Figure 2. ATCA Platform Integration at "Transilvania" University – front and rear views with details: 1-ATCA system; 2-switch with management ("remote patch panel"); 3-Linksys router (LAN); 4-TP-LINK router (Mi-Fi – WAN, Wi-Fi, 3G) with antenna (5) & Huawei E220 UMTS modem (6); 7-WebCam; 8-Power contactor; 9-TeamViewer screen with WebCam snapshot; 10-Windows console; 11-port replicator; 12-Linux console

The platform was extended for *remote access* [5] – Fig.2. We have added full remote power-on/off capabilities via IP-controlled electronic relays engaging electromagnetic relays that drive big contactors.

A MS Windows "console" is accessible from the *Internet* via a TeamViewer tool (remote desktop display is also visually completed by a WebCam). This console (an extension to the Service Management Point presented in the next paragraph) has the ATCA in its *Intranet* (via a Linksys WRT54GL router).

Direct administrative access to the platform blades is available via serial ports of the console. Indirect administrative access is possible via Eth LAN, using the "Pigeon Point" IP Manager installed on the SH61 chassis.

The other console – a Linux machine – can be considered an extension of the Service Creation Environment presented in the next paragraph.

It runs the RMI/NetLogic/BroadCom SDK that supports the development for the PP50 blade (the focus of the present paper).

Recently, an external 2.5 OSI level switch (with management was added, as “remote patch-panel”) as well as a serial port replicator (for complete blade control via Hyperterminal) and a Mi-Fi multi-modal backup access (WAN, Wi-Fi, 3G) via a TP-LINK TL-MR3220 router.

2.2. ATCA integration in an Intelligent Network architecture

In the Intelligent Networks (IN) customers themselves (institutions or individuals) –and not only manufacturers, owners and / or operators of the systems – are those who are *enabled* to design, develop, install, manage, operate and close the *services*.

Originally, communications IN were extensions of traditional telephone networks PSTN (Public Switched Telephone Networks) which became SSP (Service Switching Point) by adding a service-code detector – which triggers the “serving routine” by invoking the SCP (Service Control Point) – that runs the FSL (FlexibleServiceLogic). The main architectural levels of communications IN (Fig.3) are:

• Service Switching Point (SSP) where there are resident the switching functions SSF (Service Switching Functionality), CCF (Call Control Functionality) etc.

SSP implements the BCSM (BasicCallStateModel), an algorithmic state machine (finite state machine) which implements all the transitions and states related to the call processing, from the resource allocation request (dependent not only on the destination, but also on the origin and time of request), hold and release of resources. BCSM events are considered “DetectionPoints”– DP (detection of asynchronous service requests).

• Service Control Point (SCP) that implements the SCF (Service Control Functionality) and the SDF (Service Data Functionality). It is a powerful platform or server (now even a Cloud manager), treating DP requests from SSP by running the Points in Call (PC) – the above-mentioned ((DP-“interrupt”) serving routines). Additional data requiredin the processing of SSD (Service Support Data) canbe obtained from SDF.

• Service Creation Environment (SCE) – for the development (or adaptation – “customization”) of services (usually via web clients, increasingly Thin Clients - tablets or smartphones). Although they can be used even general purpose programming languages (such as C or Java), SCE benefits from dedicated languages (e.g. JAIN-SLEE extensions).

• Service Management Point (SMP) which is an intermediate level for service administration and storage – for example, as XML files, particularly XPDL (XML for Process Description Language) obtained, for instance, with BPEL (Business Programming & Execution Language).

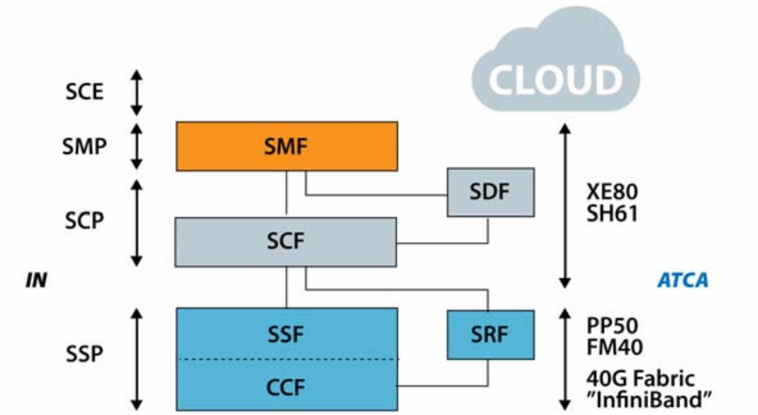

Figure 3. ATCA platform configured as IN architecture

Deep Packet Inspection (DPI) applications are expected to hit $1.5 B in 2014, and the forecast calls for growth [6]. DPI becomes critical for telecom operators in order to add value to the transmission medium used by over the top players (OTT) via services and applications from new mobile hand-held devices. Due to reduced revenues the mobile operators are interested in *monetizing* the transmission channel that is heavily used by OTT services providers (Over the Top players).

Our proposed industrial integration for the ATCA platform is suitable for monitoring and legal interception applications, for infotainment and packet processing of M2M communication.

2.3. The Packet Processing Sub-System

Our ATCA 40G platform has two PP50 blades, each with two packet processors RMI-Netlogic-Broadcom – type XLR732 – that have a *„super-scalar" architecture*. Each processor is multi-core MIPS64, each core being capable of running 4 threads (resulting a total of 32 virtual cores/processor).

Streaming „data flows" can be analyzed without being "unpacked" – a packet processor [7] can *inspect* in real time millions of simultaneous streaming threads, in *security* appliances. The packet processing enables application identification, flow *discrimination* and *traffic control* [8], [9].

In order to detect the *"digital fingerprint"* characteristic to particular flow classes, the computations take place in multiple *"virtual machines"* running the generic functions of fragmentation, directing, encapsulation, encryption (and so on) of the Packet Processing Language (PPL).

A non-blocking 10GbE switch (the "Core Switch" FM2112) interconnects the packet processors, the I/O and the backplane. Each XLR732 processor has two 10GbE ports to this switch, enabling a 10GbE duplex capacity. External I/O is both via redundant ports (1GbE standard PICMG 3.1.3 and 10GbE standard PICMG 3.1.9) to the black-plane (RTM and Fabric Interface) and with the front panel (via direct 1GbE and 10GbE ports).

3. Development of Packet Processing Solution

Our end purpose is to identify a M2M stream inside a general data capture. The capture would contain M2M traffic (e.g. DSTP) alongside normal HTTP or TCP protocols. We captured the stream in a file and analyse it or feed the stream directly to a packed processing application running on the XLR processors. For these purposes, but not only, we have devised a test environment in order to customize and debug the application [10]. The application is running on top of the RMIOS operating system loaded on one of the XLR cores (via userapp command).

Our developed application, PacketClassifier, has two main functions

1. config_parser – used to build the masks to verify the IP headers of the packets.

2. process_pkts – reads each IP header as it arrives and if it matches the classification is attempted. The program will take on an infinite loop (for) and read input messages from the messaging ring.

This sequence in Pseudocode would be:

```
while ( 1 ) {
    Issue MSGWAIT instruction with mask of Buckets to wait on
    // Thread goes to sleep, is awakened when a msg. is received
    Get the message
    Extract physical address from Packet Descriptor
    Convert physical address to virtual address
    Process the Rx packet // extract source and destination IP
if ( Tx packet ready to transmit ) {
        Initialize Packet Descriptor for the packet
        Send the Tx message
    }
}
```

As we are interested to process real data in our traffic analysis, DSTP traffic (DataSocket Transfer Protocol) represents a good candidate for M2M traffic. In order to capture the DSTP traffic there are two options:

Option 1 – Generate a dump pcap file that contains a DSTP stream. For this purpose we have used LabWindows CVI and we have identified two ways in providing data streams that can be later used by the application. A solution is to use a DSTP stream generator and dump the information into a file. For example, we have used already available deliveries from Labview CVI in the following way, see also Fig. 4

1. Initiate DSTP server (DataSocket Writer)

    1.a. Write data in file

    1.b. Write data in stream via localhost (127.0.0.1:3015)

2. Read DSTP data from stream

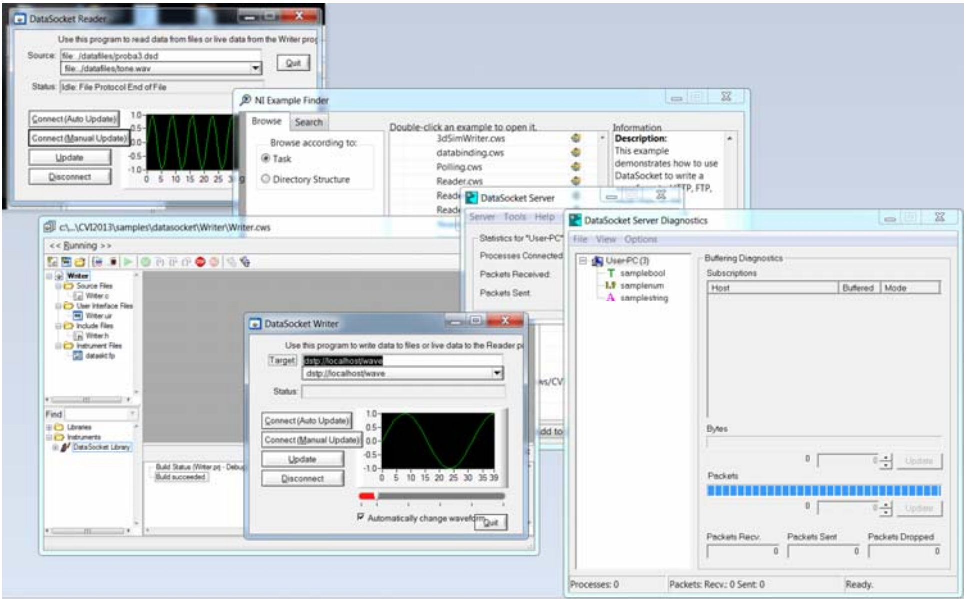

Fig. 4. LabWindows DSTP capture – Server/Writer as Stream/File and Read

Option 2 – Provide direct stream via a packet capture utility called *pretender*. provided with the RMIOS SDK for traffic supply to the gmac interfaces.

```
  Usage: ./pretender [-p port] -f <tcpdump-style filter string>
  atca@ubuntu $ cd /opt/rmi/fsim_2.0.0/install/
  atca@ubuntu  $  sudo  ./pretender  -p  6001  -i  eth0  -d  -f  "ether  dst
00:1c:25:a2:03:5f"
```

Above command will redirect all packets to gmac0, port 6001, from eth0, if they are addressed (destination) the specified MAC address. Using this command we can supply real time traffic to the packet processing application.

A valid way to send the stream from the dumpfile towards the application is to use the pretender as interceptor for data but generate the stream using an utility like dumpcap, tcprreplay or bittwist (some of them might require installation e.g. sudo apt-get install tcpreplay). We have successfully used bittwist to send the pcap file to pretender. The only constraint we have met is that we need to re-format the pcap file using tcprewrite like below:

```
  atca@ubuntu $ tcprewrite -i dumpfile.pcap -o test1.pcap --
dlt=enet --enet-smac=00:0C:29:42:F8:2A –enet
dmac=00:0C:29:42:F8:2B
```

Note: with this command we are adding test MAC addresses through conversion.

4. Packet Discrimination in Data Flow

Based on the criteria selected in the masks, TCP and UDP traffic can be separated with all the relevance that comes from that. Also, HTTP or, in our case DSTP for M2M services can be further studied.

Below is an example for simple IP filtering (the application execution is shown in Fig.5):

```
// extract the destination IP
dest_ip = (recv_pkt->data[30]<<24)|(recv_pkt->data[31]<<16)|(recv_pkt->data[32]<<8)|(recv_pkt->data[33]);
printk("Destination IP: %d.%d.%d.%d \n",
 (int) ((dest_ip>> 24) & 0xff), (int) ((dest_ip>> 16) & 0xff),
 (int) ((dest_ip>> 8) & 0xff), (int) ((dest_ip>> 0) & 0xff));
// extract the source IP
src_ip = (recv_pkt->data[26]<<24)|(recv_pkt->data[27]<<16)|(recv_pkt->data[28]<<8)|(recv_pkt->data[29]);
printk("Source IP: %d.%d.%d.%d \n",
 (int) ((src_ip>> 24) & 0xff), (int) ((src_ip>> 16) & 0xff),
 (int) ((src_ip>> 8) & 0xff), (int) ((src_ip>> 0) & 0xff));
```

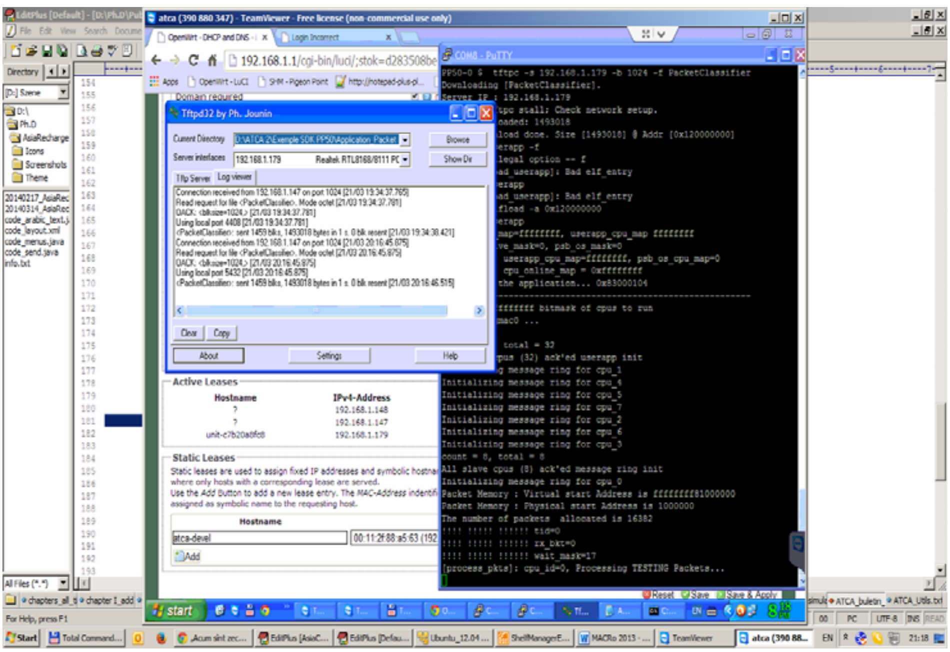

Fig. 5. Application execution on ATCA platform via remote access (Tftpd32)

5. Conclusions

As packet processing and flow discrimination is becoming a required capability for any telecom operator, ATCA integrations are to become more common. Our study and practical integration and development demonstrated the possibility of traffic discrimination, in an industrial scenario, involving DSTP. The presented technique can be successfully applied also to statistics, control, filtering, resource allocation or legal survey.

ATCA integration was modelled on IN, with emphasis on remote access. Our purpose is to implement a *service factory* with special capabilities for packet processing, especially real time DPI. In order to do that the server is enabled for instrumentation also, based on the principles of AXI (ATCA eXtensions for Instrumentation).

Further development we intend is oriented towards advanced M2M (Machine-to-Machine) communication. This type of communication has a nearly exponential growth, and in short time should overcome, in volume, the human-to-human (or human-to-machine) communication.

Last but not least, the real-time packet processing capabilities on ATCA would enable content-aware / application-aware SDN – software-defined networks to be implemented and to reach the inferior OSI layers, towards the protocol synthesis and embedded security.

Acknowledgement

This paper is supported by the Sectorial Operational Programme Human Resources Development (SOP HRD), ID134378 financed from the European Social Fund and by the Romanian Government.

References

[1] R. Kuhlmann - ATCA: Advanced Telecom Computing Architecture - Die Plattform der Zukunft für Telekommunikationssysteme - Franzis Verlag GmbH, 2007 – ISBN 9-7837-7234-1298

[2] PCI Industrial Computers Manufacturers Group - ATCA Short Form Specification – 2003

[3] Bergstrom E. (2003): Advanced TCA: A Force of One in Telecom & Datacom Applications. Crystal Cube Consulting

[4] Mellanox Technologies (2009): PICMG 3.2 Advanced Telecommunications and Computing Architecture

[5] Sandu, F., Costache, C., Balan, T. C., Balica, A. N: "Packet Processing on an ATCA 40G Platform" 2013, Proceedings of the 4th International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (MACRo2013), Scientia publishing house, 2013, ISSN: 2247 – 0948, pp. 227-238 / 239-250 / 251-258; 4-5 October, 2013, Tirgu Mures, Romania

[6] Radisys (2014). Partnering For DPI Deployment. FierceMarkets custom publishing. Available: http://go.radisys.com /rs/radisys/images/ebook-atca-partnering-for-dpi-deployment.pdf [last accessed: April 2014]

[7] Verlag Dr. Mueller e.K., 2007 – ISBN 9-7838-3642-9764Freescale Semiconductor, Inc - DPI Solutions for Telecom Networks – 2008

[8] H. Holma, A. Toskala, K. Ranta-aho, J. Pirskanen - High-Speed Packet Access Evolution in 3GPP Release 7 - IEEE Communications Magazine, Volume:45, Issue 12, pp.29-35, 2007 – ISSN 0163-6804

[9] Ipoque GmbH - Smart Traffic Management Policy Control and Charging in Converging IP Networks, 2012

[10] Alexandru Balica, Cosmin Costache, Florin Sandu, Dan Robu - Deep Packet Inspection for M2M Flow Discrimination, Integration on an ATCA Platform – Bulletin of the Air-Force Academy „Henri Coandă" Brasov, Vol XII, No 2 (26) 2014, pag.85-90, ISSN-L: 1842-9238