Resource Virtualization with Programmable Radio Processing Platform

(Invited Paper)

Zoran Miljanić and Predrag Spasojević
WINLAB, Rutgers University
671 Route 1 South, North Brunswick, NJ 08902, USA
{miljanic, spasojev}@winlab.rutgers.edu

ABSTRACT

As wireless communication moves to all IP and packet based processing, the virtualization techniques will need to be introduced to provide common interface to higher layers which will hide the necessary details for resource reservations and sharing. In the circuit switched world the communication resources are guarantied with the allocation of channels that satisfy certain bandwidth requirements, where the allocated bandwidth is taken regardless of the actual utilization. In the IP packet based world the reservations have a soft form with statistical guaranties allowing dynamic sharing of bandwidth, while the bandwidth is requested by setting the Service Level Agreement (SLA) parameters. Setting up SLA parameters and monitoring their compliance can take complex forms, so it is advantageous to abstract the SLA related communication from the higher networking layers and shield them from the changes in the environment and system load.

The virtualization is even more important for the reconfigurable and programmable devices suitable for cognitive radio applications. The changes in frequency band, modulation scheme, or other baseband and MAC protocols characteristics need to be monitored by the virtualization layer which is responsible for mapping original SLA of each flow to the new settings.

We present the novel programmable radio processing platform architecture framework, Virtual Flow Pipelining – FVP, which has underlying mechanisms for per flow performance guaranties. The SLA guaranties are enforced per flow using the scheduling mechanisms that will allocate proportional share of each hardware resource to the flow. The allocations of the resource share to the flow create the *Virtual Flow* consisting of the sequence of processing steps on the required processing modules. The wireless protocol processing often imposes end to end latency requirements for the whole processing flow, so the resource scheduler need to take the full flow latency into account as well.

General Terms

WICON'08, November 17-19, 2008, Maui, Hawaii, USA.

Design, Economics, Experimentation

Keywords

Virtualization, SDR – Software Defined Radio, Cognitive Radio, Multimodal Devices, Flexibility, Performance, OFDM, IP

1. INTRODUCTION

We have witnessed the explosive growth of wireless devices in the last decade. Every month more than 1.6 million mobile phones are being sold worldwide [2]. This explosive growth will coincide with next generation wireless infrastructure that will consist of multiple radio network technologies, the data rates from tens of megabits per second to a gigabit per second, and require efficient use of available spectrum to accommodate the growth. The expected spectrum liberalization will require either continuous or frequent spectrum sensing in addition to (per-packet) agility in adaptation to the interference conditions, automatic frequency band selection, and power control. The solutions for efficient spectrum usage in heterogeneous wireless environments drive the need for collaboration and intelligent adaptation of wireless devices, which will recognize the radio access and network infrastructure of the environment and dynamically change to satisfy connectivity needs [3], [4], [5], [6]. Simultaneous support for diverse traffic streams and dynamic adaption of devices drive the need for the virtualization support that will protect higher layer protocols from the changes at the radio access features and device loading with the simultaneous traffic flows.

Furthermore, the flexibility can not compromise the speed and Quality of Service, which are the fundamental requirements of the emerging mobile multimedia applications [5], [7], [8]. We propose a new programmable radio processing platform that will satisfy throughput requirements of the emerging 4G protocols, and provide deterministic performance guaranties to the multiple flows simultaneously [9].

The ongoing WLAN deployments and emerging 4G wireless broadband standards will impose heterogeneous wireless communication environments, where the infrastructure will be built with devices using different radio access technologies, and operating at different spectrum bandwidth. The inherent heterogeneity is already recognized by the ongoing 4G

Digital Object Identifier: 10.4108/ICST.WICON2008.4887
http://dx.doi.org/10.4108/ICST.WICON2008.4887

standardization efforts where the interoperability between devices is the requirement, while the infrastructure and terminal devices are allowed and expected to operate using variety of radio and higher layer communication protocols. The only common characteristics of the emerging wireless communication protocols is IP packet based communication. The flexibility and agility for per packet protocol selection is required not only for the interoperability, but even more for the support of new protocols that are in development and will emerge with the completely new applications domains (vehicular telemetry, M2M). Achieving the flexibility at high data rates on a per packet basis is very challenging because of the need to support at-speed processing for short (40 bytes) packets [10]. The problem is significantly more challenging if the protocol selection is required per packet basis, and even more so if the flexibility is required for the computationally intensive wireless physical layer processing. Satisfying these conflicting requirements of low latency processing time, fast context switching per packet basis, and high computation complexity is not feasible with the current stored program architecture processors optimized for static applications rather than dynamic and I/O intensive applications in the network centric environment.

The two extreme strategies considered so far for supporting the interoperability are multimodal hardware and full software implementation [5], [7] of digital domain processing. The multimodal hardware approach does not solve the requirement for a dynamic future evolution which will affect all communication layers, since it does not have enough or any flexibility to process the communication protocols different from the ones that the hardware is designed for. Also, as the number of protocols and their complexity increases it becomes an impractical solution for interoperability requirements.

The fully software programmable solution of high speed wireless protocol remains elusive with current state of the technology. We can not rely on Moore's low to catch up with the performance requirements since the gap between processing complexity of high speed wireless protocols and processing power of SoC devices is only increasing: The processing complexity of wireless protocols experiences the CAGR (Compound Annual Growth Rate) of 78% [1], while SoC performance is increasing at CAGR of 22%. This growing gap between the future demand and the capabilities of existing solutions calls for the research in breakthrough network centric architecture solutions that will take into account the new processing paradigms.

The traditional CPU architectures are compute centric with relatively static data sets and processing tasks orders of magnitude longer than the time it takes to establish the working data set. The traditional computer architecture techniques for the performance improvement through the exploitation of instruction level parallelism and inherent concurrency in applications are based on the characteristics of the application that do not hold in the network centric world. The communication protocol processing is a short sequence of data manipulation operations, sequential nature with high data dependency, and no or very little inherent concurrency to exploit at fine (instruction level) or course (program or task level) granularity of parallelism.

The new paradigms of computing in the pervasive wireless world require new techniques for performance improvement and scalable architectural solutions that will support flexible processing, rapid reconfiguration and efficient cross layer communication in collaborative environment.

2. VIRTUALIZATION REQUIREMENTS FOR WIRELESS COMMUNICATION

As wireless communication moves to all IP and packet based processing, the virtualization techniques will need to be introduced to provide common interface to higher layers which will hide the necessary details for resource reservations and sharing. In the circuit switched world the communication resources are guarantied with the allocation of channels that satisfy certain bandwidth requirements, where the allocated bandwidth is taken regardless of the actual utilization. In the IP packet based world the reservations have a soft form with statistical guaranties allowing dynamic sharing of bandwidth, while the bandwidth is requested by setting the Service Level Agreement (SLA) parameters. Setting up SLA parameters and monitoring their compliance can take complex forms, so it is advantageous to abstract the SLA related communication from the higher networking layers and shield them from the changes in the environment and system load. The following figure illustrates layering and use of virtualization for the interface towards the higher layers.

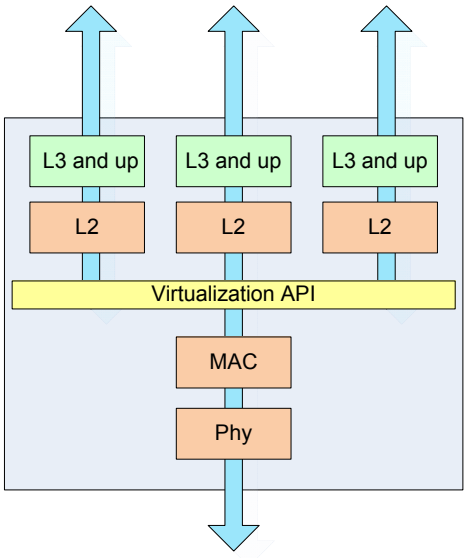

Figure 1: Virtualization Architecture

The virtualization layer handles the hardware resources responsible for managing communication bandwidth. These processing resources as well as communication bandwidth are shared among the sessions, so the virtualization is responsible for SLA enforcement for each session as well as protection between the sessions so that each gets the required share of the bandwidth. Each session treats its share of the physical bandwidth as the separate (virtual) channel, unaware of the other sessions in the system. Thus the higher protocol layers stay the same regardless of the number of sessions and physical bandwidth allocation.

Digital Object Identifier: 10.4108/ICST.WICON2008.4887
http://dx.doi.org/10.4108/ICST.WICON2008.4887

The virtualization is even more important for the reconfigurable and programmable devices since the physical characteristics change as well.

3. VIRTUAL FLOW PIPELINING

The virtualization layer monitors the setting of physical resources. The changes in frequency band, modulation scheme, or MAC protocols characteristics are monitored by the virtualization layer which is responsible for mapping original SLA of each flow to the new settings.

The SLA guaranties are enforced per flow, so the virtualization scheme requires the underplaying hardware resource scheduling mechanisms that will allocate proportional share of each hardware resource to the flow. The allocations of the resource share to the flow create the *Virtual Flow* consisting of the sequence of processing steps on the required processing modules. The wireless protocol processing often imposes end to end latency requirements for the whole processing flow, so the resource scheduler need to take the full flow latency into account as well.

Providing the latency and processing bandwidth share guaranties is difficulty with the software controlled processing platform. The software control assumes that the scheduler is implemented in software and requires real time operating system for sharing CPU processing power as well as the hardware processing engines used by the software as accelerators. The time scales of real-time operating system slices and CPU context switching in the software controlled environment are order of magnitude larger than the ones required by the wireless protocol processing (tens of μs vs. μs). Thus Software Defined Radio platforms are not adequate for the multi-flow communication support since time slicing of processing resources will be inefficient and non-precise. We introduce Virtual Flow Pipelining (VFP as the novel architectural approach for programmable radio protocol processing which supports controlled coexistence of communication flows and delivers high performance through the high utilization of hardware and software processing resources. The efficient techniques for resource sharing and low overhead context switching used by VFP, yield high speedup with parallelization or CPU and hardware processing engines.

4. VIRTUAL FLOW PIPELINING

VFP concept is illustrated in Figure 2.

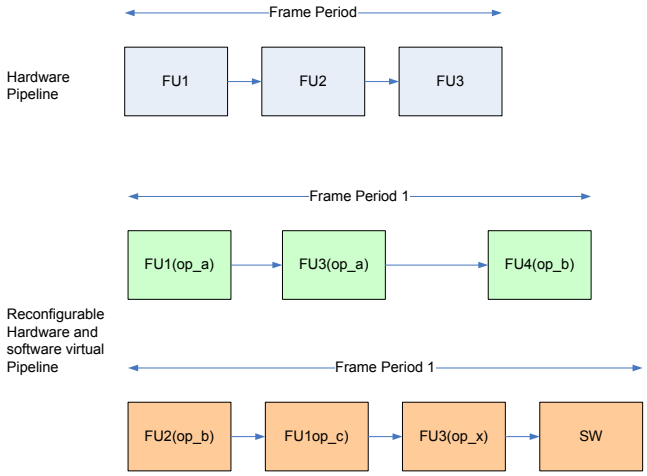

Figure 2: Virtual Flow Pipelining (VFP) processing

The traditional hardware pipeline based approach has a fixed sequence of operations, a fixed operation at each stage of the selected operating mode, and a fixed timing of operations, i.e., end to end processing latency. Furthermore there is no provision for multiplexing functional units among the flows provisioned in the system which is the basic requirement for enabling the processing platform virtualization. VFP approach adds the flexibility with respect to each design dimension described above and, in addition, allows software defined functions to be incorporated into the VFP based program control framework.

Virtual flow consists of a set of functions and their scheduling requirements associated with a higher protocol entity (application, session, IP, or MAC address). In a VFP scheme, the sequence of operations is specified by a flow control data structure which defines operations for each step in the flow, follow up steps and condition for the selection of the next steps in the case that processing flow may take alternative paths depending on ht run time results. The potential sequence space is defined during the flow provisioning time, but the actual operation sequence is determined at run time. Furthermore, the synchronization mechanisms of a VFP scheme are provided to resolve data dependencies and race conditions between processing units. The operations performed by functional units have variable size parameters for data and control information, allowing for the additional flexibility

The timing of the operations is also provisioned per flow, but dynamically selected based on the run time results. The scheduling function of the VFP controller multiplexes each functional unit (hardware or programmable processor) either based on a deterministic scheduling policy (time reservation) or statistical policy, depending on the flow setup. In order to support synchronous framing type of protocols (e.g., time division multiplexing), the flow scheduling information for the time reservation based scheme also specifies the repetition time. The scheduler is in charge of ensuring both the deterministic and the statistical performance guaranties.

In addition, the functional units can be of the central processing engine type, allowing the software programmable functions. These functions are incorporated in the processing flow using the

Digital Object Identifier: 10.4108/ICST.WICON2008.4887
http://dx.doi.org/10.4108/ICST.WICON2008.4887

same mechanisms as the ones for the functions performed by hardware modules, retaining all the system flow properties, and also relieving the software from engaging in the costly collaboration and communication operations. The hardware scheduler associated with the central processing units acts in lieu of an operating system.

The same VFP scheme also incorporates hardware functional units and software programmable processors for performing the higher layer protocols processing, including MAC, L2, and L3 layers. The units are shared across the layers based on either deterministic or statistical guaranties.

5. WIRELESS PROTOCOL PROCESSING VIRTUALIZATION FRAMEWORK

Supporting multiple concurrent flows and dynamic bandwidth allocation to the flows in wireless communication networks requires virtualization. Each flow is a virtual entity described with the following properties:

- Flow identifier – used by the higher layer protocols for association for data traffic mapping
- Deterministically or statistically guarantied allocation of processing resources
- Flow specification program
- Communication between functional elements involved in the protocol processing

The flows share the resources within the architectural framework which needs to insure that per flow properties are maintained throughout flow lifetime. In addition, the flows are protected form each other, i.e. that traffic fluctuations in one flow will not disrupt guarantied properties of the other flows.

Flow Identification

The flow identifier is unique number with the local domain significance with the communication nodes. It is used to identify flow control information specifying other flow properties, as well as data buffers storing the data for the functions within the protocol layer as well as for the interface to the higher layer that is communicating using the virtual flow.

Resource sharing between the flows

Controlling the processing and communication capacity of functional units is performed by the scheduler that combines deterministic and statistical allocation polices. The deterministic policy makes allocation per repetitive flow frame intervals, reserving the corresponding functional unit for the required processing time at the specific interval within the periodic frame. The remaining capacity of the functional units that is not reserved by the deterministic policy is allocated by the statistical scheduling policy using the fixed priority or Weighted Round Robin (WRR) scheme.

Figure 3 illustrates the VFP scheduling policy:. The processing elements are performing their functions as tasks which are initiated by the scheduler. The scheduler deterministic allocations are handling synchronous tasks which are scheduled for execution at the allocated time slot within the repetitive frame. As figure 2 illustrates, each virtual flow has associated time frame. The figure 3 shows the allocation of reserved bandwidth time slots to the synchronous tasks, and shared bandwidth to the asynchronous tasks. The time slots can possibly belong to the different virtual

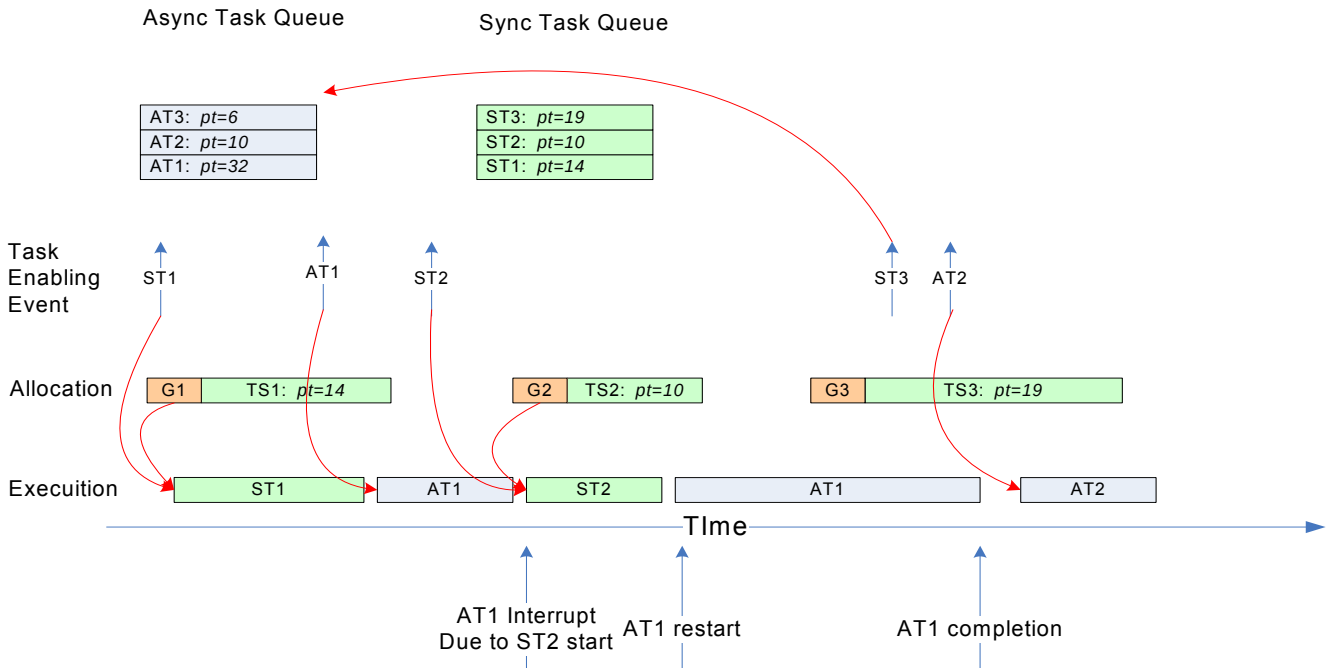

Figure 3: VFP scheduler policy

Digital Object Identifier: 10.4108/ICST.WICON2008.4887
http://dx.doi.org/10.4108/ICST.WICON2008.4887

flows, so resources allocation policy ran during the flow provisioning time ensures that processing time is reserved according to the flow bandwidth and timing requirements. The synchronous tasks are not interruptible in order to ensure their deterministic behavior in the presence of other tasks.

The statistical scheduling policy is handling asynchronous tasks which are allocated in the shared processing bandwidth left over after the allocation of the reserved capacity by the deterministic scheduler. The asynchronous tasks are scheduled by fixed or Weighted Round Robin scheduling discipline. The asynchronous tasks are interrupted at the start time of the synchronous task time slot, and resumed at the point of interruption after the completion of the synchronous task. The interruption is also necessary in order to guaranty the processing bandwidth for the deterministic processing of synchronous tasks.

The functions that have the strict time latency requirements, like physical layer processing sequence that has to fit within the protocol frame (like FFT, demodulation, decoding) are handled by synchronous tasks. The other functions, like MAC filtering, and L2 forwarding are handled by asynchronous tasks.

Flow programs

The VFP functions are executed as tasks, where task can run on potentially multifunctional hardware engines or software programmable CPU-s. The sequencing and selection of tasks is the top level programming paradigm of VFP framework. The VFP system controller is responsible for selecting for each step the task function and its associated parameters on each processing engines. The sequencing is enforced by ordering, function (or thread on CPU) selection and synchronization between the processing units. The following notation describes Virtual Flow Processing sequence specification:

$FU_i[op_a(ipar_1, ipar_2, opar_1) \rightarrow FU_j[op_k(ipar_1, opar1_1, oparm_2) \rightarrow FU_k[op_d(ipar_1, opar_1)$

The specification says that op_a at functional unit FU_i is followed by op_k at functional unit FU_j etc. The input and output parameters are specified for each operation. The operation execution constitutes that *task*. The underlying architecture support mechanism ensures synchronization between the tasks in the flow. The task can be performed by the multifunctional hardware unit, or by the programmable CPU (thread). In both cases tasks scheduling and sequencing is controlled by VFP system controller which maintains VFP performance and virtualization properties.

Communication

The VFP system controller also performs communication between functional units in the system, transferring data from the outputs of one functional unit to the inputs of the following units, as per input and output parameter references (address bases). The transfer is performed by the built in DMA engine which relives hardware processing engine or CPU-s of data transfer burden. Separating data transfer from processing roles also improves determinism in processor bandwidth allocation.

6. CASE STUDY

We will illustrate the VFP architecture with the example support for physical and MAC layer processing of simultaneous 802.11a and 802.11n flows. Figure 4 describes both the transmitter downlink processing flows and the functional units necessary to support these flows. The processing tasks necessary to complete a 802.11n flow are a superset of the ones needed to execute a 802.11a flow.

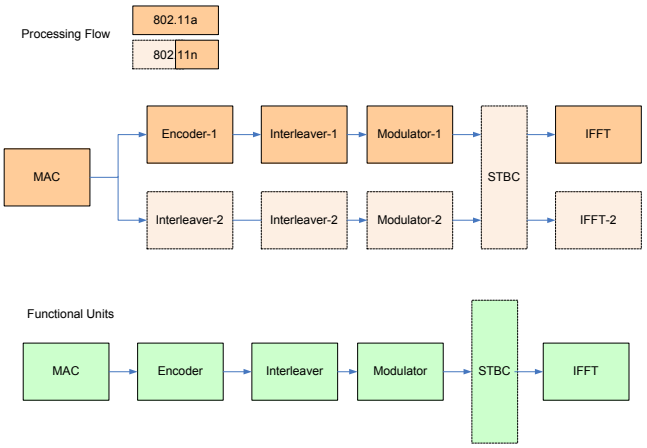

Figure 4: 802.11a/802.11n

The dotted line blocks correspond to those processing tasks where the two flows differ. The necessary functional units have the same functionalities between the two standards except for the space time block coding block (STBC) which is not needed by 802.11a. All functional units except the multiple access control (MAC) unit are synchronous task processing units which should operate in the described order and can be pipelined between different flows. In the case of 802.11n the two different

Digital Object Identifier: 10.4108/ICST.WICON2008.4887
http://dx.doi.org/10.4108/ICST.WICON2008.4887

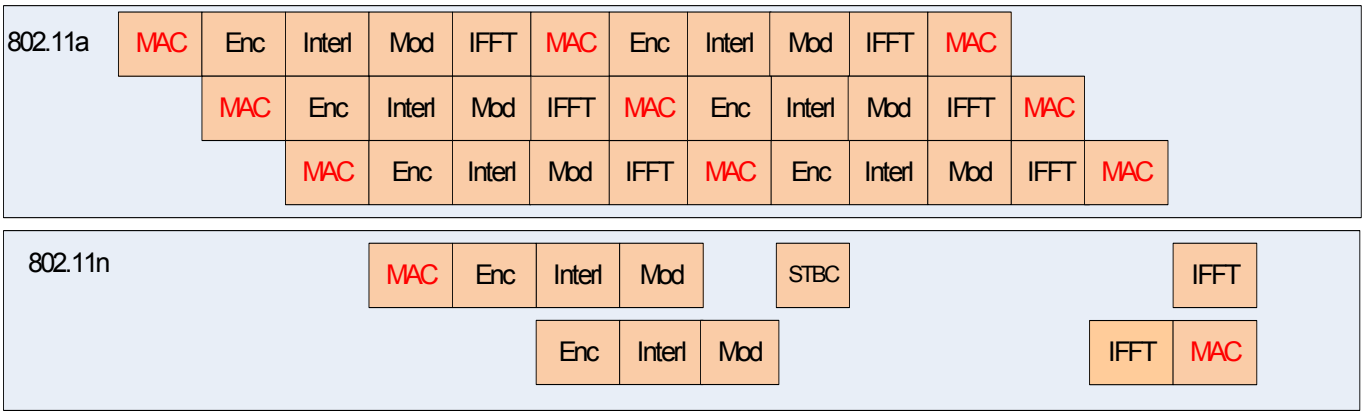

Figure 5: Transmitter Schedule with 3 SLA requests for a 802.11a and an 802.11n SLA downlink request transmitter processing flows and functional units

streams can be pipelined whereas, as is natural, all processing tasks have to wait for their input processing to be completed (e.g., STBC) before being scheduled. The MAC works with the scheduler to enable the resource sharing and ensure satisfying the SLA requests. The requests are satisfied by capturing the link and functional unit resource capabilities, based on the assigned coding rates, the modulation indices, and overall schedule timing requirements.

Figure 5 depicts the downlink transmission task schedule where the upper layers request transmission on three 802.11a flows, and one 802.11n flows. The 4 flows have potentially different SLA requirements. The MAC and the scheduler satisfy the requests based on SLA parameters, while ensuring that no functional unit is operating simultaneously for a given flow. For illustration simplicity, in this graph each task has the same duration. In this example the SLA requests are satisfied whereas two MAC frames are processed for each of the three 802.11a flows while only a single MAC 802.11n frame has been completed.

7. CONCLUSION

The convergence of wireless communicates to the IP based with soft QoS guaranties raises the importance of virtualization as the mechanism manage sharing the physical resources among the traffic flows and to abstract the details and changes of physical channel characteristics. The proliferation of different radio access network (RAN) standards and need for the simultaneous support for multiples standards at infrastructure and terminal devices further increases the need for virtualization. The introduction of programmable devices as the means to support multiple RAN standards further drive the need for the virtualization in order to abstract the adaptation of the communication channel and network access schemes, and maintain higher payer protocols resilient and transparent to the changes.

We have presented the novel Virtual Flow Pipelining (VFP) programming model and processing platform architecture as the means to support virtualization of traffic flows within the flexible and programmable framework. The VFP based processing platform enables the design of protocol processing application with the performance guaranties in a programmable manner and to support multiple traffic flows through the virtualization mechanisms.

We have studied in the simulation environment the characteristics of concurrent flow support for 802.11a and 802.11n streams and found out the VFP scheme gives adequate performance for concurrent flow support as long as the functional modules have enough aggregate performance and VFP control overhead is kept less than 100 clock cycles per function activation (task). This control overhead is well within the reach of practical ASIC implementation.

In our further study we will investigate the multi flow processing on the real VFP platform implemented with Xilinx Vertex 5 SX95T devices and compare, and investigate the effects of the real control overhead on multiple flow support and resources sharing within VFP framework. Development of different wireless protocols and their summations run time behavior analysis is planned for the next phase

Digital Object Identifier: 10.4108/ICST.WICON2008.4887
http://dx.doi.org/10.4108/ICST.WICON2008.4887

8. REFERENCES

REFERENCES

[1] Hardware Technology Exploration: Impact of Technology Evolution on E2R, End-to-End Reconfigurability (E^2R) White Paper, December 2005

[2] The Road to 4G: Will LTE, UMB and WiMAX Just Be Stops Along the Way?, In-Stat research report, August 2007

[3] J. Mitola III, "Cognitive Radio: An Integrated Agent Architecture for Software Radio," *PhD thesis*, Royal Institute of Technology (KTH), Sweden, May 2000.

[4] D. Raychaudhuri, "Adaptive Wireless Networks Using Cognitive Radios as a Building Block," *MobiCom Keynote Speech*, Sept. 2004.

[5] Vanu software-defined radio, http://www.vanu.com

[6] KU Agile radio, https://agileradio.ittc.ku.edu

[7] GNU Radio Project, http://www.gnu.org/software/gnuradio

[8] Universal Software Radio Peripheral (USRP), http://www.ettus.com/downloads/usrp_1.pdf, see also http://comsec.com/wiki?UniversalSoftwareRadioPeripheral

[9] Z. Miljanic et al., "The WINLAB Network Centric Cognitive Radio Platform - WiNC2R," Proc. CrownComm, Orlando, Florida: 2007.

[10] Yavatkar, R., and H. Vin (eds.). IEEE Network Magazine. Special issue on Network Processors: Architecture, Tools, and Applications 17, 4 (July 2003).

Digital Object Identifier: 10.4108/ICST.WICON2008.4887
http://dx.doi.org/10.4108/ICST.WICON2008.4887