

6G Wireless Access Protocol Development Using Intel DPDK for 50Gbps Throughput

Jungpil Choi[†], Yongseouk Choi[†], Heesang Chung[†]
Telecommunications and Media Research Laboratory[†]

Mobile Communication Research Division Electronics and Telecommunications Research Institute (ETRI)
{jpchoi, choiys, hschung}@etri.re.kr

Abstract—Next-generation 6G mobile communication systems will deliver high data rates and ultra-low latency services through sub-terahertz or terahertz bands. This makes developing wireless access protocols for 6G mobile communication systems a very challenging task. In this paper, we adopt Intel DPDK as a wireless access protocol development platform to implement a wireless access protocol with a processing speed of up to 50Gbps and present its test results. This work demonstrates the feasibility of implementing Tbps-class L2 protocols required for 6G networks using Intel Xeon CPUs and DPDK.

Index Terms—6G Mobile, Terahertz Wireless, Multi-Band Multi-Connectivity, Wireless Access Protocols

I. INTRODUCTION

6G networks aim to achieve significantly wider bandwidths than existing 5G bands by utilizing sub-terahertz and terahertz frequencies. This will enable data rates at least 10 times higher than 5G, paving the way for services like holographic communication, high-resolution digital twins, and XR streaming. Furthermore, with latency reduced by more than 10 times compared to 5G, 6G targets real-time interactive services such as instantaneous communication between autonomous vehicles and real-time remote surgery.

A major challenge in implementing 6G networks is achieving low latency and high throughput while transmitting massive amounts of packets. Traditional network stacks face limitations in scalability and efficiency due to excessive overhead. DPDK (Data Plane Development Kit) addresses this by providing a user-space packet processing framework that bypasses the kernel, allowing direct access to the Network Interface Card (NIC). It also offers DSA (Data Streaming Accelerator) as a hardware accelerator for large-volume data processing, enabling user-space applications to achieve low latency and high throughput. Therefore, the Intel Xeon general-purpose processor is well-suited as a development platform for 6G networks.

We've developed a UE(User Equipmet) wireless access protocol hardware platform and DPDK-based software, targeting a 6G Proof-of-Concept (PoC) mobile communication system. This system aims to deliver 200Gbps data processing at the physical layer and 50Gbps at the upper-layer service level. This paper describes the development methodology and performance validation for a UE wireless access protocol designed to handle 50Gbps service traffic. [1]–[3].

II. WIRELESS ACCESS PROTOCOL

A. DPDK(Data Plane Development Kit)

DPDK is a high-performance packet processing software development kit that enables high-speed packet I/O by allowing user applications to directly process packets from network cards on general-purpose computers, bypassing the kernel. First released by Intel in 2010, DPDK provides open-source libraries and drivers. DPDK libraries offer features such as processor affinity for independent CPU(Central Processing Unit) core usage, huge pages to mitigate TLB(Translation Lookaside Buffer) misses, UIO(User-space I/O) for direct hardware control, polling-mode packet processing, queue handling via lockless synchronization, batch packet handling for concurrent processing of multiple packets, and support for SSE(Streaming SIMD Extensions) and NUMA(Non-Uniform Memory Access) awareness [4].

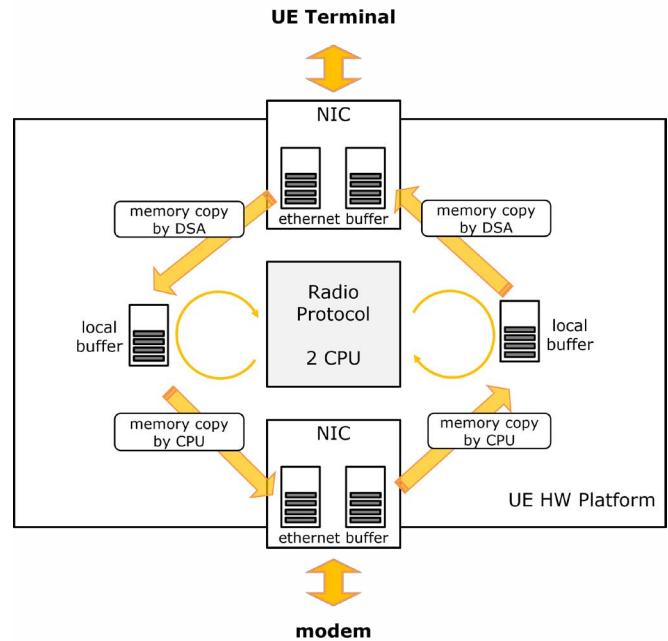


Fig. 1. UE L2/L3 software platform.

B. Wireless Access Protocol Software Platform

As shown in the figure 1, the L2/L3 upper-layer software platform architecture on the Xeon CPU is implemented to

enable memory copying between Ethernet buffers and local buffers by selectively using DSA(Data Streaming Accelerator) and the CPU for high-volume traffic processing. The wireless access protocol is implemented across two CPUs for 50Gbps processing, with each CPU having a processing capability of 25Gbps and consisting of 32 multi-cores. Individual L-cores within the CPU can have wireless access protocol function blocks and I/O interface functions independently allocated, enabling high-speed processing.

III. UE L2 PROTOCOL SOFTWARE

A. UE downlink L-Core flow

The figure 2 illustrates the Downlink L-Core flow of the User Equipment(UE) interacting with the modem emulator. The downlink L2 protocol was implemented using 7 out of 32 total L-Cores within a single CPU. When a Transport Block(TB) received from the modem emulator arrives via the Port 0 Ethernet interface, the received TB data is first processed by the I/O-Rx Core's I/O API(Application Programming Interface). It's then converted into DPDK Mbuf format and passed to the FAPI-Rx Core via a DPDK Ring Queue. The FAPI-Rx Core moves this data into a Protocol DPDK Pool before transferring it to the MAC Core via a DPDK Ring Queue. The MAC(Medium Access Control) Core processes the protocol PDU(Protocol Data Unit) and then forwards the processed SDU(Service Data Unit) to the RLC Core via a DPDK Ring Queue. The RLC Core processes the received RLC(Radio Link Control) PDU and delivers the processed RLC SDU to the PDCP(Packet Data Convergence Protocol) Core via a DPDK Ring Queue. The PDCP Core processes the received PDCP PDU and then passes the processed PDCP SDU to the SDAP(Service Data Adaptation Protocol) Core via a DPDK Ring Queue. Finally, the SDAP Core processes the received PDU and transmits the processed User Data Packet to the IO-Tx Core via a DPDK Ring Queue. The IO-Tx Core then uses Intel DSA(Data Streaming Accelerator) to move the UE User Data into a DPDK Pool for the Port 1 Ethernet interface, after which it forwards the corresponding Mbuf to the Port 1 Ethernet Queue.

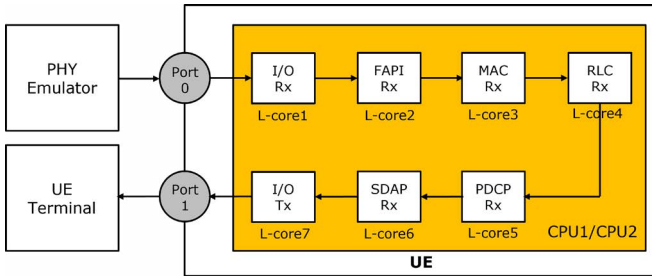


Fig. 2. UE Downlink L-core flow.

B. UE uplink L-Core flow

The figure 3 illustrates the Uplink L-Core flow of the UE interacting with the modem emulator. The uplink L2 protocol was implemented using 10 out of 32 total L-Cores within a

single CPU. When a UE User Data Packet is received by the Port 1 Ethernet interface, the received packet is stored in the Ethernet Queue for Port 1. The IO-Rx Core monitors this queue and receives the packet in DPDK Mbuf format. The received User Data Packet Mbuf is then moved to the protocol DPDK Pool using Intel DSA, and this DPDK Mbuf is transferred to the SDAP Core via a DPDK Ring Queue. The SDAP Core processes the received SDU and forwards the processed PDU to the PDCP Core via a DPDK Ring Queue. The PDCP Core processes the received SDU and delivers the processed PDU to the RLC Core via a DPDK Ring Queue. The RLC Core processes the received SDU and then passes the processed PDU to the MAC SCHED Core via a DPDK Ring Queue. The MAC SCHED Core considers the order of received packets and assigns them to match the TB (Transport Block) Size to be generated by FA0 and FA1. It then transmits the assigned TB-SDU mapping information to the MAC FA0 Core and MAC FA1 Core using a DPDK Ring Queue. The MAC FA0/1 Cores then form the TB based on the received information, perform protocol processing, and transfer it to the FAPI Core using a DPDK Ring Queue. The FAPI Core moves the received TB into a single DPDK Mbuf in the DPDK Pool for the Port 0 Ethernet interface. It then sends this DPDK Mbuf, containing the TB, to the I/O Tx Core via a DPDK Ring Queue. Finally, the I/O Tx Core processes the received DPDK Mbuf with the I/O API and then forwards it to the Ethernet Queue of the Port 0 Ethernet interface for transmission to the modem emulator.

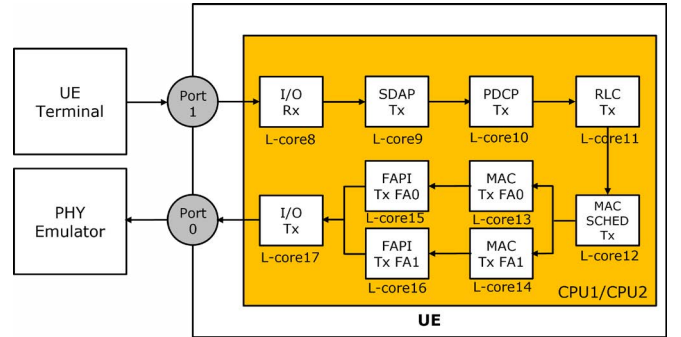


Fig. 3. UE Uplink L-core flow.

IV. UE L2 LOOPBACK TEST

A. Testing Environment

- Topology: Packet Gen - UE Uplink(UL) - PHY Emulator - UE Downlink(DL) - Packet Gen.
- Objective: To measure the traffic processing speed of the User Plane in UE.
- Measured Packet Sizes: 64, 256, 512, 1024 bytes, etc.
- Measurement Data :
 - Maximum Throughput: Determined by incrementally increasing the packet generator's transmission rate for each specific packet size.
 - Receive Buffer Delay: Quantified by progressively increasing the packet generator's rate for each packet

size and measuring the resulting latency within the receive buffer.

B. Test Procedure

For hardware platform and software performance validation, as shown in the figure 4, a Packet Generator is connected to a 100G Ethernet port of the hardware platform, and a Modem Emulator(PHY Emulator) is connected to another Ethernet port. The modem emulator outputs received signals without modification and provides TTI(Transmission Time Interval) signals to the CPU where the wireless access protocol is installed. On CPU1 of the hardware platform, after processing IP packets received from the packet generator with the wireless access protocol, MAC TBs(Transport Blocks) of a fixed size are created. These are then fragmented into Ethernet frame sizes and transmitted in synchronization with the TTI signal period received from the modem emulator. On CPU2 of the hardware platform, Ethernet frames received from the modem emulator (i.e., looped back from CPU1) are reassembled to restore the MAC TBs. These are then processed by the wireless access protocol and transmitted back to the packet generator as IP packets. The packet generator checks for errors in the received packets and counts only successfully received packets to measure the receive speed.

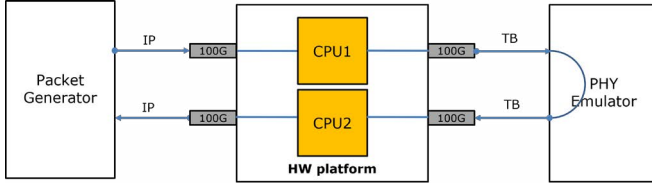


Fig. 4. L2 loopback test.

C. Test Results

As shown in the figure 5, the UE's data processing speed increases as the maximum packet size generated by the packet generator progressively grows from 128 bytes to 1024 bytes. The UE's maximum speeds are approximately 2.15 Gbps for 128-byte packets, 5.7 Gbps for 256-byte packets, 12.1 Gbps for 512-byte packets, and 24.24 Gbps for 1024-byte packets. The maximum speed increases with larger packet sizes due to factors like reduced packet overhead and decreased CPU load, leading to improved system processing efficiency. Therefore, to achieve maximum speed, IP packets should be at their maximum size.

TABLE I
UE L2 PROTOCOL MAXIMUM THROUGHPUT.

Packet Size(byte)	128	256	512	1024
Packet Generator(Gbps)	1.6	3.6	7.2	14
Max. Throughput(Gbps)	2.15	5.7	12.1	24.24
Queuing delay(us)	424	344	514	742

The table 1 shows the maximum processing speeds for each packet size. For example, if the packet generator produces

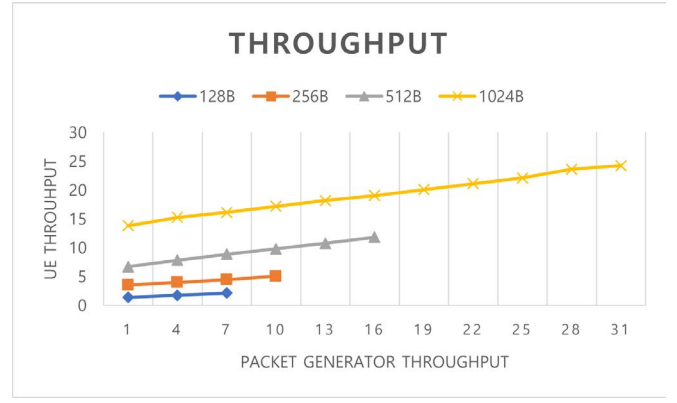


Fig. 5. UE L2 protocol throughput by packet size.

1024-byte raw data at a speed of 14 Gbps, then with various overheads added, packet data is generated and output at approximately 24.24 Gbps. Queuing delay refers to the latency from when a packet enters the UE until its first output.

V. CONCLUSION

In this paper, we presented the implementation method and test results for a UE L2 protocol achieving approximately 25 Gbps per CPU, totaling about 50 Gbps data processing speed using two CPUs. This demonstrates that Tbps-class L2 protocols required for 6G networks can be implemented using Intel Xeon CPUs and DPDK.

ACKNOWLEDGMENT

This research is supported by an Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00746, Development of Tbps wireless communication technology)

REFERENCES

- [1] M. M. Kiasaraei, K. Nikitopoulos, and R. Tafazolli, "Towards ultra-power-efficient, tbps wireless systems via analogue processing: Existing approaches, challenges and way forward," *IEEE Communications Surveys & Tutorials*, 2023.
- [2] Y. Liu, Y. Deng, A. Nallanathan, and J. Yuan, "Machine learning for 6g enhanced ultra-reliable and low-latency services," *IEEE Wireless Communications*, vol. 30, no. 2, pp. 48–54, 2023.
- [3] H. M. F. Noman, E. Hanafi, K. A. Noordin, K. Dimiyati, M. N. Hindia, A. Abdrabou, and F. Qamar, "Machine learning empowered emerging wireless networks in 6g: Recent advancements, challenges & future trends," *IEEE Access*, 2023.
- [4] DPDK Project, "About DPDK," <http://www.dpdk.org/about/>, 2025.