AMD OpenNIC을 활용한 FPGA 기반 고정밀 RTT 측정 하드웨어 설계

강대희[1], 박태준[2]
[1]전남대학교 정보보안융합학과 석사과정
[2]전남대학교 인공지능학부 교수
meohee@jnu.ac.kr, taejune.park@jnu.ac.kr

High-Precision RTT Measurement Hardware Design Based on FPGA Using AMD OpenNIC

Daehee Kang[1], Taejune Park[2]
[1]Dept. of Information Security Convergence, Chonnam University
[2]School of Artificial Intelligence, Chonnam University

요 약

기존 소프트웨어 기반의 패킷 왕복 시간(RTT, Round Trip Time) 측정 방식은 추가적인 소프트웨어 계층의 오버헤드로 인해 정확성이 저하되는 한계점이 있었다. 그러나 FPGA 기반의 NIC을 사용하는 경우 아키텍처 수정이 가능하므로, 하드웨어 기반의 RTT 측정이 가능해진다. 본 논문에서는 FPGA 기반의 AMD OpenNIC 아키텍처에 RTT 측정 기록용 모듈을 추가하여 별도의 외부 장비 없이 고정밀 RTT를 직접 측정하는 방법을 보이고자 한다. 제안된 모듈은 250MHz의 주파수에서 동작하고 최대 4ns 단위로 RTT 측정이 가능하며, 20ns 내의 측정 오차만 보여 기존 소프트웨어 방식 대비 높은 정확성을 보인다. 본 기술은 향후 다양한 네트워크 환경에서의 성능 평가 및 최적화 연구에 활용이 가능할 것으로 기대된다.

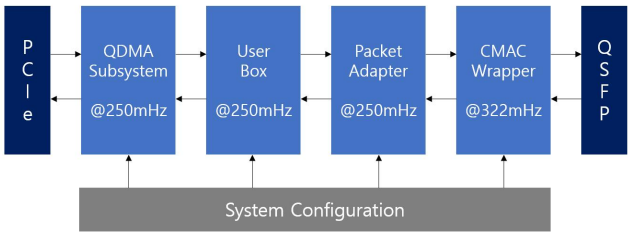
1. 서론

초고속 네트워크가 100Gbps를 넘어 400Gbps까지 상용화되면서, 데이터센터 내부 지연(latency)은 서비스 품질과 직결되는 핵심 지표가 되었다. 그리고 마이크로서비스, 분산 AI 학습과 같은 애플리케이션은 지연 시간이 수 ns만 증가해도 처리량이나 손익에 중대한 영향을 미치기에 성능 측정 역시 나노초 단위로 정밀 측정할 필요성이 증가하였다[1, 2].

그러나 현재 소프트웨어 측정 방식(Ping, Tcpdump, iPerf 등)으로는 RTT(Round Trip Time) 측정에 있어 커널을 경유하기 때문에 인터럽트, 컨텍스트 스위치, 버퍼링 지연 등이 누적되어 마이크로초 단위의 측정 오차를 보인다[1, 3]. 따라서 클라이언트가 송신한 패킷이 서버에서 처리되고 다시 수신되기까지의 RTT를 측정하면 클라이언트에서 발생하는 오버헤드로 인해 측정값에 오차가 생긴다. 반대로 원격 서버에서 소모되는 시간을 측정하기 위해 원격 서버에서만 지연 시간을 측정하게 되면 서버의 NIC(Network Interface Card) 등의 단말기에서 혼잡 제어 등 송수신 처리에 소모되는 지연 시간을 측정할 수 없게 된다. 이는 정밀한 RTO(Retransmission Time Out)를 설정하거나 초고속 네트워크 망을 구성하는 데에 있어 불리하다[3]. 또한, eBPF 등의 수단을 쓰더라도, 서버 NIC에서 발생하는 지연은 측정할 수 없다. 따라, 고정밀 RTT 측정은 하드웨어 기반으로 수행될 필요가 있다[1].

다만, 최근 특정 환경이나 상황에 맞게 NIC을 커스텀하거나 패킷 처리 회로를 재구성해야 하는 것에 수요가 증가하면서 FPGA 기반의 NIC[4, 5, 6, 7]을 채택하는 경우가 많아졌다. 이런 경우, 고정밀 RTT 측정이 필요할 때 아키텍처를 수정하여 레지스터 접근을 통해 측정값을 얻는 것이 가능하다. 즉, FPGA 기반의 NIC을 사용하면 하드웨어 아키텍처 수정을 통해 커널을 거치지 않는 RTT 측정이 가능해진다.

본 연구에서는 FPGA 기반의 OpenNIC 아키텍처[7, 8]를 간단히 수정하는 방식을 통해 나노초 단위의 고정밀 측정 단위 성능을 달성하였다. 또한, 소프트웨어 계층에서 발생하는 오버헤드를 제거하여 20ns 오차 내의 더 정밀한 측정값을 얻어내었다. 수정한 OpenNIC 아키텍처는 Packet Adapter 내 250MHz 클락의 패킷 어댑터를 수정한 것이며 4ns 단위의 고정밀 RTT 측정 단위 성능을 보인다.



(그림 1) OpenNIC의 하드웨어 구조

2. 구현

본 연구에서는 AMD OpenNIC Shell[6]이라는 100Gbps급 NIC 아키텍처를 사용하였다. 이는 Verilog HDL (Hardware Description Language)로 작성된 하드웨어 설계, 레지스터 맵이 오픈소스로, 사용자 환경에 맞는 자유로운 아키텍처 수정이 가능하게 공개되어 있다. 또한, 하드웨어 설계에 필요한 합성 과정과 배선 과정을 자동화한 파일을 제공하여 네트워크 관리자에게 쉽게 사용하게끔 사용 진입 장벽을 낮추어 주었다. 그러나 기본 배포판은 RTT 측정 기능을 포함하지 않는다. 따라서, 본 연구에서는 해당 아키텍처에 RTT 측정 모듈을 새로 설계하고 아키텍처를 수정하여 TX, RX의 송수신 시간을 기록하는 모듈을 삽입하였다.

새로 설계된 RTT 측정 모듈에는 클락 카운터가 사용되었으며, 간편한 계산을 위해 250MHz 클락을 카운팅하였다. 카운터에는 32비트 레지스터를 사용하였기 때문에, 4ns 단위로 1씩 증가하므로 최대 17.13초의 측정이 가능하다. 카운터가 32비트를 초과해 0으로 초기화될 경우도 고려하여 설계되었기 때문에 패킷의 RTT 시간이 17.13초만 넘지 않으면 된다. 초과했을 때 작동은 (수신된 시간 - 송신한 시간)에서 0xFFFFFFFF를 더하는 방식으로 하드웨어를 설계하였다.

하드웨어 수정은 가장 앞부분인 CMAC 모듈을 수정하지 않고 Packet Adapter 부분에서 이루어졌는데, 이는 CMAC 부분에서 AMD에서 블랙박스로 처리하여 설계를 온전히 제공하지 않았기 때문이다. 또한, Packet Adapter가 하드웨어 레벨에서 패킷이 파싱됨과 동시에 처리됨이 설계를 수정하는 데에 네트워크 관리자에게 더 직관적이고 수정이 쉽기 때문이다. 따라서, 사용자는 송신되는 패킷을 별도로 수정하지 않아도 되기 때문에 어느 애플리케이션으로 패킷을 송수신하든 측정할 수 있다. 다만, CMAC에서 발생하는 오버헤드 400ns 정도가 추가된다.

<표 1> Propagation RTT 측정 성능 비교 (단위 : ns)

	평균	최솟값	최댓값	최대-최소
FPGA	392.56	408	388	20
Tcpdump	78850.00	19000	180000	161000
Wireshark	78847.43	19322	180642	161320

<표 2> Bridge RTT 측정 성능 비교 (단위 : ns)

	평균	최솟값	최댓값	최대-최소
FPGA	78832.25	14136	111128	96992
Tcpdump	164530.0	55000	306000	251000
Wireshark	161273.0	55060	306001	250941

마지막으로 FPGA 내부에 구현된 레지스터에 접근하기 위하여, HDL 파일의 레지스터 맵에서 직접 카운터 레지스터에 접근 주소를 부여하였다. 그리고 사용자 공간에서 PCIe BAR 영역을 mmap()으로 매핑하였고, 레지스터값을 직접 읽고 쓰는 방식으로 RTT 측정값을 확인하였다. 그리고 기존의 복잡한 OpenNIC의 드라이버를 수정하거나 별도의 디바이스 드라이버를 작성하지 않고 메모리 맵 I/O 방식의 레지스터에 접근하기 위해 Pcimem[9]이라는 접근 유틸리티를 사용하였다.

3. 평가

평가는 AMD Alveo U250 FPGA[10]로 진행하였다. 표 1는 QSFP28 3m 케이블의 Propagation 지연 시간을 측정한 결과이다. 10000 개의 UDP 패킷을 송수신한 결과 최소 388ns, 최대 408ns, 평균값은 392.56ns으로 나타났다. 즉, 4ns 단위이기 때문에 총 5개의 값 분포로 나타난 것이며, 오차는 20ns 내로 발생함으로 보인다. 총 지연 시간에는 QSFP 라인의 약 5ns Propagation 지연 시간과 CMAC 모듈에서 처리하는 FEC 오류 정정 코딩 등에 소모되는 나머지 시간으로 소모된다.

반면 Tcpdump와 Wireshark로 RTT를 측정했을 경우 최댓값-최솟값의 오차가 크게 나타난다. 이는 커널, 패킷 큐 등 다양한 곳에서 발생하는 소프트웨어 계층 딜레이 타임이 모두 반영되었기 때문이다. 더불어 흔히 사용되는 Tcpdump는 나노초 단위 측정이 불가능하다. Wireshark는 나노초 단위 측정은 지원하였으나, Tcpdump와 큰 차이가 없는 오버헤드를 보였다. 따라서 소프트웨어 계층으로 인해 발생하는 오버헤드는 약 784us 정도의 큰 값를 보이며, 오차 역시 하드웨어를 직접 수정한 것과 160us 이상 차이를 보인다.

표 2는 A서버에서 출발한 패킷이 B서버에 소프트웨어로 구현된 브릿지를 통과하여 A서버로 돌아오는 RTT를 측정한 것이다. 표 1의 RTT가 매우 짧기에 교차검증을 진행한 것이지만, 각 애플리케이션과 FPGA의 차이가 마찬가지로 약 82000ns 정도 표 1에 버금갈 만큼 차이가 난다. 이 결과를 통해 다양한 환경에서의 RTT 측정에 있어서도 유리하다는 것을 알 수 있다. 따라서, 고정밀 RTT 측정이 필요할 경우, FPGA 기반의 NIC을 사용할 때 하드웨어 설계 수정을 통해 측정하는 것이 바람직하다.

4. 결론

본 논문에서는 AMD OpenNIC을 활용하여 FPGA 기반의 NIC을 사용하는 경우 약간의 아키텍처 수정을 통해 고정밀로 RTT를 직접 측정할 수 있음을 보였다. 그리고 FPGA 내부에서 RTT 측정 모듈과 카운터 모듈을 새로 설계함으로써 추가 장비 없이도 최대 4ns 단위의 정밀도로 RTT를 측정할 수 있음을 보였다. 또한, 실험을 통해 RTT 측정 오차 범위가 16ns로, 애플리케이션 기반 측정 방식과 본 연구와 비교하여 제안하는 하드웨어 방식이 더 정확하고 안정적임을 보여준다.

추후 연구로는 본 기술을 다양한 네트워크 환경 및 프로토콜에 적용하여 성능 측정 및 최적화 연구로 더욱 발전시키는 방향으로 진행하고자 한다. 또한 FPGA 기반의 하드웨어 가속 기술과 연계하여 실시간 성능 모니터링 및 최적화 기술 개발을 추가로 진행할 예정이다.

Acknowledgement

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 인공지능융합혁신인재양성사업(IITP-2023-RS-2023-00256629) 및 대학ICT연구센터사업(IITP-2024-RS-2024-00437718)의 연구결과로 수행되었음.

참고문헌

[1] Dreibholz, T. (2023). High-Precision Round-Trip Time Measurements in the Internet with HiPerConTracer. 2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 1-7.

[2] Arslan, S., Li, Y., Kumar, G., & Dukkipati, N. (2023). Bolt:{Sub-RTT} congestion control for {Ultra-Low} latency. In 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23) (pp. 219-236).

[3] Mittal, R., Lam, V.T., Dukkipati, N., Blem, E. R., Wassel, H.M., Ghobadi, M., Vahdat, A., Wang, Y., Wetherall, D., & Zats, D. (2015). TIMELY: RTT-based Congestion Control for the Datacenter. Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication.

[4] Forencich, A., Snoeren, A.C., Porter, G., & Papen, G.C. (2020). Corundum: An Open-Source 100-Gbps Nic. 2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), 38-46.

[5] Zhong, G., Kolekar, A., Amornpaisannon, B., Choi, I., Javaid, H., & Baldi, M. (2023). A Primer on RecoNIC: RDMA-enabled Compute Offloading on SmartNIC. ArXiv, abs/2312.06207.

[6] Zilberman, N., Audzevich, Y., Covington, G.A., & Moore, A.W. (2014). NetFPGA SUME: Toward Research Commodity 100Gb/s.

[7] AMD, AMD OpenNIC Project, https://github.com/Xilinx/open-nic, 2021.

[8] AMD, AMD OpenNIC Shell, https://github.com/Xilinx/open-nic-shell, 2021.

[9] Bill Farrow, Pcimem application, https://github.com/billfarrow/pcimem, 2011.

[10] AMD, AMD AlveoTM U250 Data Center Accelerator Card (Passive), https://www.amd.com/en/products/accelerators/alveo/u250/a-u250-p64g-pq-g.html, 2018.