# LRO 기능을 활용한 리눅스 수신 제로카피 성능 분석

황재훈[1], 김기수[1], 황재현[2]
[1]성균관대학교 소프트웨어학과 학부생
[2] 성균관대학교 반도체시스템공학과 교수
hanktank0704@gmail.com, kimkisu2502@g.skku.edu, jh.hwang@skku.edu

# Performance Analysis of Linux Receiver Side with LRO

Jae-Hun Hwang[1], Kisu Kim[1], Jae-Hyun Hwang[2]
[1]Dept. of Computer Science, Sungkyunkwan University
[2]Dept. of Semiconductor Engineering, Sungkyunkwan University

## 요 약

기존 커널에서의 zerocopy 는 SmartNIC 의 header/data split 기능과 페이지 정렬된 데이터를 요구하여 일반적인 환경에서는 제로카피 구현이 어려웠다. 본 연구는 이러한 한계를 극복하고자, 오늘날 데이터센터에서 널리 사용되는 NIC 의 LRO 기능을 활용하여 추가 하드웨어 없이 제로카피를 구현하였다. 이를 위해 비정렬 데이터도 직접 메모리에 매핑될 수 있도록 커널 로직을 개선하고, 사용된 메모리 페이지가 효율적으로 재사용되도록 메모리 회수 정책을 최적화하였다.

## 1. 서론

최근 데이터센터 및 클라우드 서비스 환경에서 네트워크 속도의 급격한 증가로 인해 CPU 와 메모리 대역폭이 성능의 병목으로 작용하고 있다. 특히 NIC 에서 커널 버퍼로, 다시 유저 버퍼로 데이터를 복사하는 과정이 해당 환경에서 가장 큰 성능 저하 원인으로 지적되고 있다. 이를 해결하기 위해 복사 과정을 생략하는 제로카피 기술이 연구되고 있으며, 대표적으로 mmap 방식이 있다. 본 연구의 기초가 되는 TCP_MMAP 은 NIC 에서 수신한 데이터를 커널 공간에 저장한 뒤, 이를 유저 공간에 직접 매핑(mmap)하여 커널 공간과 유저 공간 사이의 데이터 복사를 제거하는 방식으로 작동한다. 그러나 이 방식은 header split 기능과 함께 데이터가 메모리 페이지 단위로 정렬되어 있어야 효율적으로 동작할 수 있으며, 일반 NIC 에서는 이러한 페이지 정렬이 보장되지 않아 적용에 한계가 있었다. 기존 연구들은 이 문제를 해결하기 위해 스마트 NIC 이나 FPGA 와 같은 특수 하드웨어의 기능을 활용했지만, 이는 시스템 비용과 복잡성을 높이는 단점을 갖는다.

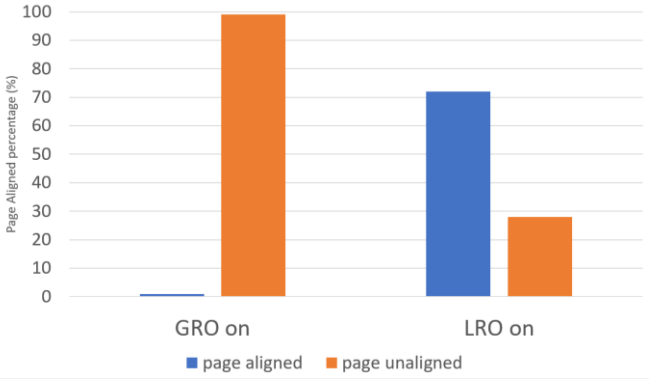
본 연구는 추가적인 하드웨어를 사용하지 않고도 제로카피 수신을 구현하기 위해 오늘날 데이터센터에서 사용되는 NIC 이 제공하는 LRO(Large Receive Offload) 기능에 주목하였다. LRO 는 수신된 여러 작은 패킷을 하나의 큰 버퍼로 통합하는 하드웨어 기능으로 자연스럽게 데이터가 페이지 단위로 정렬되는 현상을 관찰했다. 이를 활용하여 본 연구에서는 LRO 를 활용하여 일반 NIC 환경에서도 TCP_MMAP 방식보다 더 범용적이고 효율적인 제로카피 수신을 구현하고자 하였으며, Linux 커널을 수정하여 구현하고 성능을 분석하였다.

## 2. 본론

### 2.1. 실험 환경

본 연구는 Dell PowerEdge R750 서버(syslab03, syslab04)를 사용하여 진행되었다. 두 서버 모두 Intel Xeon Gold 6354(3.0GHz, 2 소켓, 18 코어) CPU 와 256GB DRAM 메모리를 탑재하고 있으며, Mellanox ConnectX-6 VPI(200Gbps) NIC 과 Intel X710(10GbE) NIC 을 보유하고 있다. 저장장치는 1.6TB NVMe SSD(Samsung PM1735)와 1.6TB SAS SSD 로 구성되어 있다. 실험 커널은 Linux Kernel 6.6.9 를 기반으로 수정되었으며, Mellanox NIC 의 LRO(Large Receive Offload) 기능을 활성화하고 MTU 는 9000 으로 설정한 상태로 패킷 수신 시 페이지 정렬 상태를 분석하였다.

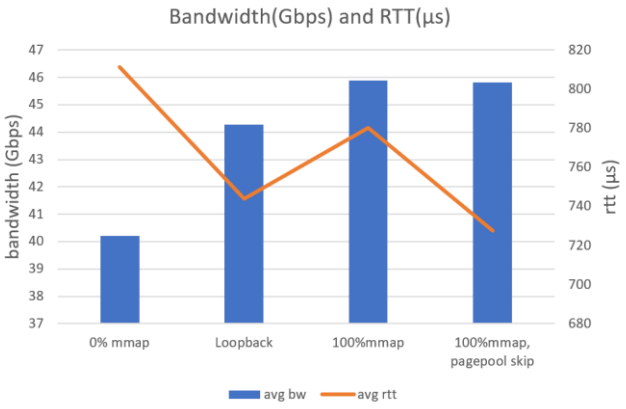
### 2.2. 페이지 정렬




(그림 1) 페이지 정렬 데이터 비율.

그림 1 은 tcpdump 를 이용해 수신된 패킷을 수집하고, 이후 Wireshark 를 통해 분석을 한 결과이다. 실험 결과, LRO 가 성공한 패킷의 개수는 전체 패킷의 약 6.5%로 소수에 불과했으나, 이들 패킷의 크기가 커 데이터 양 기준으로는 전체의 약 70%를 차지하는 것으로 나타났다. 결론적으로 전체 수집 데이터 2.5GB 중 약 1.8GB 가 LRO 처리를 통해 수신된 데이터였다.

이를 통해, LRO 만으로도 TCP_MMAP 에서 zero copy 를 적용할 수 있는 충분한 양의 패킷이 존재한다고 판단했다.




(그림 2) 성능 비교 차트

본 연구에서는 TCP_MMAP 기반의 zero copy 기능을 활용하여 데이터 카피 오버헤드를 최소화하고자 했다. 기존 커널에서는 데이터가 완벽히 페이지 단위로 정렬된 경우에만 mmap 을 통해 유저 공간으로 전달할 수 있었다. 그러나 실제 환경에서 수신된 SKB 에 포함된 frag 의 시작이나 마지막 page 는 종종 페이지 크기(4096 바이트)보다 작은 비정렬(unsaturated) 데이터를 포함하며, 이 부분은 mmap 을 활용하지 못하고 read() 시스템 호출을 통해 복사해야 하는 문제가 존재했다.

이러한 문제를 해결하기 위해, tcp_zerocopy_receive() 함수의 내부 로직을 개선하였다. 우선 페이지 크기보다 작은 크기의 마지막 비정렬 데이터(fragments)도 일단 mmap 으로 유저 영역에 전달되도록 수정하였다. 이후, 유저 영역에 전달되는 tcp_zerocopy_receive 구조체에 변수를 추가적으로 설정해서 mmap 이 된 비정렬 데이터가 시작되는 부분을 설정했다. 이를 통해 유저는 맵핑이 성공한 데이터의 영역에서 시작하는 부분을 알 수 있어 정상적으로 데이터에 접근할 수 있게 된다.

이러한 커널 코드 개선을 통해 mmap 을 통한 데이터 전달 비율을 100%에 가깝게 증가시킬 수 있었다. 그러나 성능 평가 과정에서 기존의 코드에서는 관찰되지 않았던, clear_page_erms() 라는 함수가 많은 양의 cpu 를 차지하는 현상이 관찰되었다. 위 함수는 mmap 이후 사용된 페이지들이 충분히 page_pool 로 회수되지 않아, page_pool 이 고갈될 때마다 추가적인 페이지 할당 작업을 수행한다. Mmap 의 비율이 크게 개선되면서 맵핑에 사용되는 page 의 개수도 그만큼 늘어나 발생하는 page 부족현상으로 판단했다.

추가적으로 커널 코드에서 clear_page_erms() 함수를 호출하는 page_pool_refill_alloc_cache() 함수를 분석했다. 여기서 page_pool 에서 페이지가 고갈이 될 경우, 사용이 끝난 페이지를 초기화하여 page_pool 이 사용할 수 있게 채워주는 역할을 수행한다. 하지만 page 를 확인하는 과정에서 동일한 NUMA 노드에 소속된 페이지만 허용하고 나머지 페이지는 재활용이 되지 않음을 확인하였다. 이는 페이지가 부족한 환경에서 오히려 성능 저하를 초래하는 원인이 되었다고 판단했다. 이러한 비효율을 해소하기 위해 NUMA 노드에 관계없이 초기화된 페이지를 재사용할 수 있도록 관련 커널 코드를 수정하였다.

그림 2 는 성능 평가 결과이다. 기존 loopback 기반 전송에서는 평균 대역폭이 약 44.27Gbps, 평균 RTT 가 744µs 로 측정되었다. 그러나 TCP_MMAP 기반의 제로카피 수신을 적용한 경우 평균 대역폭은 45.8Gbps 로 증가하였으나, 평균 RTT 는 오히려 780µs 로 다소 상승하는 현상이 관찰되었다. 이는 mmap 적용으로 데이터 복사 오버헤드는 줄였지만, page pool 관리 측면에서의 추가적인 부하가 발생했기 때문으로 해석된다. 이에 대응하여 page pool 재활용 최적화를 추가한 결과, 평균 대역폭은 45.8Gbps 로 유지되면서도 평균 RTT 는 727µs 로 개선되었다. 최종적으로, page_pool 최적화를 통해 대역폭 이점을 유지하면서도 기존 loopback 대비 더 낮은 RTT 를 달성하여 latency 까지 효과적으로 줄일 수 있었음을 확인할 수 있었다.

## 3. 결론

본 연구에서는 일반적인 데이터센터 NIC 의 LRO 기능을 활용하여 추가적인 하드웨어 없이도 일반적인 NIC 환경에서 효과적으로 제로카피 수신을 구현할 수 있음을 보였다. TCP_MMAP 기반의 zero copy 코드를 개선하여, 페이지 정렬이 되지 않은 데이터까지 mmap 으로 매핑할 수 있도록 커널 코드를 수정하였고, 이를 통해 데이터 복사 오버헤드를 제거하며 mmap 성공률을 100%에 가깝게 향상시켰다. 추가적으로, page_pool 관리 방식을 최적화하여 NUMA 노드에 관계없이 페이지를 유연하게 재사용하도록 개선함으로써, bandwidth 의 손실 없이 rtt 를 개선했다. 실험 결과, 기존 대비 평균 대역폭은 약 14% 증가하고, RTT 는 약 10% 감소하는 성과를 얻었다. 본 연구는 고속 네트워크 환경에서도 범용 하드웨어만으로 제로카피 기술을 실용화할 수 있는 가능성을 제시했으며, 향후 고대역폭의 네트워크 환경에 적합한 메모리 관리 기법 연구로도 이어질 수 있을 것으로 기대된다.

## 참고문헌

[1] Q. Cai, M. Vuppalapati, J. Hwang, C. Kozyrakis, and R. Agarwal, "Towards µs tail latency and Terabit Ethernet: Disaggregating the host network stack," *ACM SIGCOMM Conference*, Amsterdam, Netherlands, August 2022.

[2] Q. Cai, S. Chaudhary, M. Vuppalapati, J. Hwang, and R. Agarwal, "Understanding host network stack overheads," *ACM SIGCOMM Conference*, Virtual Event, USA, August 2021.