

## INTEROPERABILITAS NDN-DPDK DAN FORWARDING NFD DALAM VIDEO STREAMING BERBASIS NDN PADA VIRTUAL MACHINE

**Syafaat Deha Febriyan, Malldi Saesar Ariffudin, Syifa Aulia Sakira Nur Rahman, Anggita Yaumil Istiana, Leanna Vidya Yovita, Istikmal, Tody Ariefianto Wibowo**

Fakultas Teknik Elektro, Telkom University, Bandung, Indonesia

Email: syafaatdehafebriyan@student.telkomuniversity.ac.id<sup>1</sup>,  
malldisaesar@student.telkomuniversity.ac.id<sup>2</sup>,  
syifaauliasakiranr@student.telkomuniversity.ac.id<sup>3</sup>,  
anggityaumulistiana@student.telkomuniversity.ac.id<sup>4</sup>,  
leanna@telkomuniversity.ac.id<sup>5</sup>  
istikmal@telkomuniversity.ac.id<sup>6</sup>  
ariefianto@telkomuniversity.ac.id<sup>7</sup>

### Abstrak

Named Data Network (NDN) adalah konsep jaringan masa depan yang mengatasi beberapa masalah yang ada dalam arsitektur internet saat ini. Salah satu keuntungan utama NDN adalah penggunaan caching konten di setiap router, memungkinkan akses data lebih cepat dan efisien. Seiring perkembangan teknologi jaringan internet, permintaan konten video terus meningkat setiap tahun. Oleh karena itu, optimasi dan pengiriman video memerlukan kecepatan akses yang tinggi agar client dapat menjalankan konten dari server tanpa hambatan. Penelitian ini menggabungkan NDN-DPDK (Named Data Network - Data Plane Development Kit) yang dimana dirancang sebagai hardware acceleration atau meningkatkan performa dari sisi hardware, dan menggunakan forwarder default NDN yaitu Network Forwarding Daemon (NFD) yang memungkinkan berbagai macam eksperimen dengan arsitektur NDN. Implementasi dan analisis performa video streaming ini akan menggunakan parameter Quality of Service (QoS) seperti RTT, throughput, dan startup delay untuk mengukur kualitas streaming video disisi jaringan. Selain itu, dilakukan pengukuran CPU pada router yang paling dekat dengan client untuk mengevaluasi beban traffic yang diperlukan forwarder untuk mentransmisikan data. Hasil implementasi dan analisis pada penelitian ini mendapatkan kualitas streaming video pada pengakses kedua dengan waktu RTT 4 detik, throughput diatas 0,4 MBps, video startup delay 8 detik, dan penggunaan CPU 100%.

**Kata kunci:** Named Data Network, NFD, NDN-DPDK, Video Streaming.

<b>How to cite:</b>	Syafaat Deha, Malldi Saesar, Syifa Aulia, Anggita Yaumil, Leanna Vidya, Istikmal, Tody Ariefianto, (2023) Interoperabilitas NDN-DPDK dan Forwarding NFD dalam Video Streaming Berbasis NDN pada Virtual Machine, (8) 7, <a href="http://dx.doi.org/10.36418/syntax-literate.v6i6">http://dx.doi.org/10.36418/syntax-literate.v6i6</a>
<b>E-ISSN:</b>	2548-1398
<b>Published by:</b>	Ridwan Institute

### **Abstract**

*The Named Data Network (NDN) is a future network concept that addresses some of the issues present in the current internet architecture. One of the main advantages of NDN is the use of content caching at each router, enabling faster and more efficient data access. As internet network technology develops, the demand for video content continues to increase every year. Therefore, video optimization and delivery require high access speeds so that clients can run content from the server seamlessly. This research incorporates NDN-DPDK (Named Data Network-Data Plane Development Kit), which is designed as hardware acceleration, and uses NDN's default forwarder, Network Forwarding Daemon (NFD), which allows a wide variety of experiments with NDN architecture. The implementation and analysis of video streaming performance will use Quality of Service (QoS) parameters such as RTT, throughput, and startup delay to measure the quality of video streaming on the network side. In addition, CPU measurements are made on the router closest to the client to evaluate the traffic load required by the forwarder to transmit data. The results of implementation and analysis in this study obtained video streaming quality on the second accessor with an RTT time of 4 seconds, throughput above 0.4 MBps, a video startup delay of 8 seconds, and 100% CPU usage.*

**Keywords:** *Named Data Network, NFD, NDN-DPDK, Video Streaming.*

### **Pendahuluan**

Pada jaringan saat ini, permintaan akan konten video terus meningkat dan akan menjadi masalah pada jaringan masa depan (Fauzi, Harly, & Hanrais, 2012). Cisco memperkirakan pada tahun 2022 mendatang, hampir 79% dari trafik data yang dihasilkan oleh perangkat bergerak adalah trafik data video (Pepper, 2013). Jaringan internet saat ini memiliki beberapa kelemahan, terutama dalam pengoperasiannya (Rahman, Amarullah, & Hidayah, 2020). Salah satu contohnya adalah proses mentransfer paket data yang mengakibatkan data yang diakses oleh user menjadi cukup lama saat melakukan transfer paket data, bahkan tidak jarang terjadi request time out (RTO), Van Jacobson mengusulkan arsitektur jaringan internet masa depan yang terbaru yaitu Named Data Network (NDN) yang dapat mengatasi kelemahan jaringan internet saat ini (Zhang, 2014).

Named Data Network (NDN) merupakan arsitektur jaringan internet yang baru, arsitektur jaringan internet NDN memiliki beberapa tujuan yaitu salah satunya adalah untuk mengatasi kelemahan pada jaringan internet sebelumnya yaitu pada jaringan Internet Protocol (IP) (Wang, Moiseenko, & Zhang, 2015). Named Data Network mengirimkan paket ke alamat tujuan tertentu untuk mengambil data yang diidentifikasi dengan nama, yang mana user tidak memerlukan IP address untuk mengakses sebuah konten pada internet melainkan menggunakan nama konten (Janius, 2013). Komunikasi pada jaringan NDN dilakukan melalui pertukaran 2 jenis paket yaitu, interest paket dan data paket dimana kedua paket ini bisa mengidentifikasi nama yang ditransmisikan kedalam suatu paket data (Melati, Yovita, & Mayasari, 2021). Salah satu kelebihan yang dimiliki jaringan NDN ini adalah kemampuan caching konten pada setiap router NDN

(Satria & Rahardjo, 2016). Dengan kemampuan tersebut user akan lebih dekat dengan konten yang diharapkan mampu mengurangi latensi dan akan meningkatkan kecepatan pada permintaan konten (Rukmana et al., 2023).

Penelitian NDN yang sudah dilakukan di lab NDN-RG masih menggunakan forwarding default dari NDN yaitu NFD (Named Data Network) merupakan forwarding bawaan atau default dari Named Data Network (NDN). NFD hanya mendukung single core dan single threads saja. Fungsi utama NFD adalah untuk meneruskan paket dan data packets. Selain meneruskan paket, NFD juga supports beberapa strategi forwarding, dan management interface untuk mengkonfigurasi, mengontrol, dan memantau NFD (Ahdan & Rahardjo, 2016). Dalam penelitian ini akan mencoba menggabungkan forwarder NFD dengan forwarder yang memiliki fitur multi-core yaitu, NDN-DPDK.

Dimana NDN-DPDK mengalokasikan sebagian besar struktur datanya dalam 100 Gbps, berfungsi untuk menghilangkan latensi yang tidak dapat diprediksi pada saat pemrosesan paket. Selain itu, NDN DPDK mengurangi kegagalan alokasi memori pada saat sistem bekerja [6]. DPDK menggunakan akses memori lokal (memori yang terletak di socket NUMA), dengan menggunakan NUMA kecepatan transfer data lebih cepat up to 20%. Saat penerapan awal, NDN memiliki interest forwarding yang berbeda-beda. NDN-DPDK menggunakan forwarding strategy kesetiap entry FIB, ketika interest sampai dan tidak dipenuhi oleh Content Store (CS), forwarding plane melakukan pencarian FIB yang sesuai selain itu menentukan nexthop yang bisa mengatasi interest tersebut.

Penelitian sebelumnya, NDN-DPDK berhasil diintegrasikan dengan NFD dan diimplementasikan hanya dengan test NDNPING saja (Ramadha, Yovita, & Wibowo, 2022). Pada penelitian ini akan diimplementasikan dalam video streaming. Video streaming merupakan teknologi pengiriman data, video atau audio dalam bentuk yang telah dikompresi melalui jaringan internet yang ditampilkan oleh player secara realtime dimana dapat memutar file audio atau video secara langsung ataupun pre-recorded tanpa perlu mengunduh video terlebih dahulu.

Pada awal pengembngannya, video streaming memiliki beberapa protokol seperti Transmission Control Protocol/User Datagram Protocol (TCP/UDP) dan Realtime Transport Protocol/Realtime Transport Control Protocol (RTP/RTCP), TCP dan RTCP digunakan pada pengiriman data/control signalling sedangkan UDP dan RTP digunakan pada pengiriman media audio/video. Dalam penggunaannya UDP lebih banyak digunakan dan lebih cepat dalam pengiriman karena tidak memerlukan proses three-way handshake seperti TCP (Krishnan & Sitaraman, 2012).

Adapun kebutuhan yang harus dipenuhi dalam penelitian ini, membangun arsitektur NDN membutuhkan router yang mempunyai kemampuan untuk mendukung NDN. Selain itu, memiliki kemampuan caching content agar user bisa lebih dekat dengan content yang diharapkan mampu mengurangi latensi dan akan meningkatkan kecepatan pada permintaan konten (Yudha, Kom, & Kom, 2020). Aplikasi video streaming sangat sensitif terhadap performansi network di jaringan, oleh karena itu Pengujian arsitektur jaringan

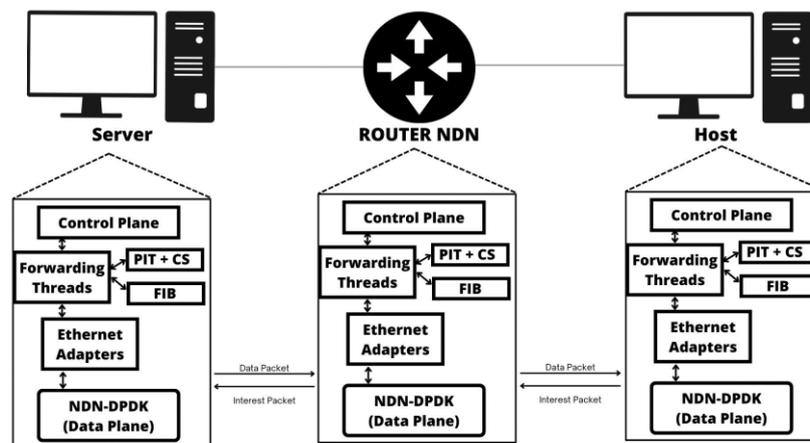
yang akan dibangun nantinya akan melibatkan pengukuran kinerja video streaming dengan menggunakan parameter Quality of Service (QoS) (Yudha et al., 2020).

Sistem video streaming dibangun dengan menggunakan server yang akan diuji melalui permintaan dari client untuk mengambil data dari server menggunakan *ndncat*, yang nantinya akan digunakan untuk memeriksa jaringan dan menentukan protocol yang digunakan.

## Metode Penelitian

### NDN-DPDK (*Named Data Network -Data Plane Development Kit*)

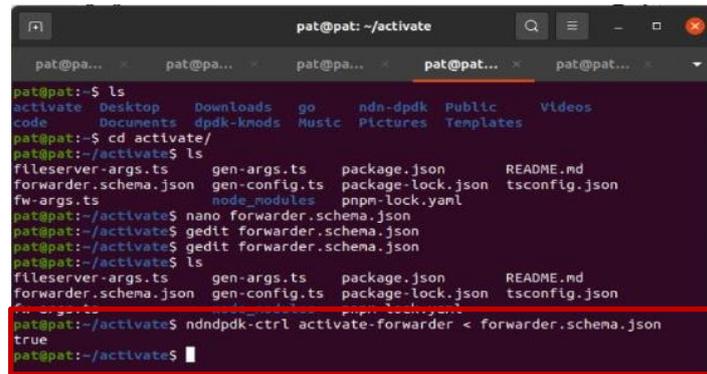
NDN-DPDK berfungsi sebagai *forwarder* yang mampu mencapai *throughput* lebih dari 100 Gbps saat berjalan pada *commodity hardware* (Shi, Pesavento, & Benmohamed, 2020).



Gambar 1 Cara Kerja NDN-DPDK

Berdasarkan Gambar 1 Cara kerja NDN-DPDK untuk melakukan pengiriman sebuah paket data memiliki *control plane*, *forwarding threads*, *ethernet adapters*. NDN-DPDK sebagai *data plane* yang berfungsi untuk mengontrol jalannya NDN-DPDK. Cara kerja NDN-DPDK memiliki perbedaan dengan *forwarder* lainnya bahwa NDN-DPDK memiliki penggabungan antara PIT (*pending interest table*) dan CS (*content store*) dengan penggabungan tersebut dapat menjadi lebih efisien saat melakukan pengiriman paket data. *Ethernet adapters* pada NDN-DPDK berfungsi sebagai jalannya komunikasi dengan menggunakan MAC Address. NDN-DPDK mendukung berbagai protokol jaringan, termasuk *Ethernet*, TCP dan UDP.

Pengimplementasian yang akan dilakukan NDN-DPDK yaitu akan diimplementasikan dengan NDN *Video-Streaming*, NDN-DPDK berfungsi sebagai *forwarder* yang akan melakukan pengiriman sebuah paket data. Mengaktifkan NDN-DPDK sebagai *forwarder* atau sebagai penerusan paket dapat dilihat pada Gambar 2. Client akan melakukan *request* paket menuju *server* berupa data *video streaming*.



```
pat@pat: ~ -/activate
pat@pa... pat@pa... pat@pa... pat@pat... pat@pat...
pat@pat:~$ ls
activate Desktop Downloads go ndn-dpdk Public Videos
code Documents dpdk-knods Music Pictures Templates
pat@pat:~$ cd activate/
pat@pat:~/activate$ ls
fileserver-args.ts gen-args.ts package.json README.md
forwarder.schema.json gen-config.ts package-lock.json tsconfig.json
fw-args.ts node_modules pnpm-lock.yaml
pat@pat:~/activate$ nano forwarder.schema.json
pat@pat:~/activate$ gedit forwarder.schema.json
pat@pat:~/activate$ gedit forwarder.schema.json
pat@pat:~/activate$ ls
fileserver-args.ts gen-args.ts package.json README.md
forwarder.schema.json gen-config.ts package-lock.json tsconfig.json
pat@pat:~/activate$ ndndpdk-ctrl activate-forwarder < forwarder.schema.json
true
pat@pat:~/activate$
```

Gambar 2 Mengaktifkan Forwarder NDN-DPDK

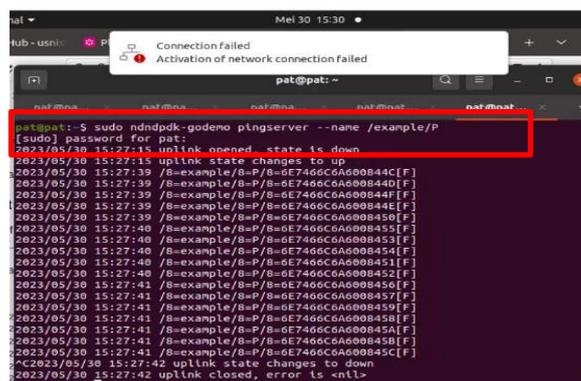
Pada Gambar 3. NDN-DPDK diaktifkan sebagai *fileserver* dengan diaktifkannya NDN-DPDK sebagai *file server* akan berfungsi untuk menyediakan paket data, paket data tersebut untuk melakukan pengiriman paket kepada *client* yang melakukan sebuah *request* paket kepada *server*.



```
pat@pat:~$ sudo ndndpdk-ctrl --gqlserver http://127.0.0.1:3031 system start
[sudo] password for pat:
pat@pat:~$ sudo ndndpdk-ctrl --gqlserver http://127.0.0.1:3031 activate-fileserver < activate/coba.json
true
pat@pat:~$
```

Gambar 3 Mengaktifkan File Server

Pada pengujian yang telah dilakukan dengan menggunakan NDN-DPDK adalah skenario NDN-PING antara *router A* dan *router B* seperti pada Gambar 4. *router B* sebagai *server* dan pada Gambar 5. *router A* sebagai *client* melakukan *request* terhadap *server*.



```
pat@pat:~$ sudo ndndpdk-godeno plngserver --name /example/P
2023/05/30 15:27:15 uplink opened, state is down
2023/05/30 15:27:15 uplink state changes to up
2023/05/30 15:27:39 /B=example/B=P/B=0E7466C6A00844C[F]
2023/05/30 15:27:39 /B=example/B=P/B=0E7466C6A00844D[F]
2023/05/30 15:27:39 /B=example/B=P/B=0E7466C6A00844E[F]
2023/05/30 15:27:39 /B=example/B=P/B=0E7466C6A00844F[F]
2023/05/30 15:27:40 /B=example/B=P/B=0E7466C6A008455[F]
2023/05/30 15:27:40 /B=example/B=P/B=0E7466C6A008453[F]
2023/05/30 15:27:40 /B=example/B=P/B=0E7466C6A008451[F]
2023/05/30 15:27:40 /B=example/B=P/B=0E7466C6A008452[F]
2023/05/30 15:27:41 /B=example/B=P/B=0E7466C6A008456[F]
2023/05/30 15:27:41 /B=example/B=P/B=0E7466C6A008457[F]
2023/05/30 15:27:41 /B=example/B=P/B=0E7466C6A008459[F]
2023/05/30 15:27:41 /B=example/B=P/B=0E7466C6A008458[F]
2023/05/30 15:27:41 /B=example/B=P/B=0E7466C6A00845A[F]
2023/05/30 15:27:41 /B=example/B=P/B=0E7466C6A00845B[F]
2023/05/30 15:27:41 /B=example/B=P/B=0E7466C6A00845C[F]
2023/05/30 15:27:42 uplink state changes to down
```

Gambar 4 PING Router B sebagai Server

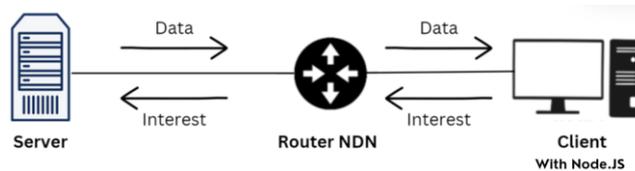
```

pat@pat: ~$ sudo ndmipdk-godeno pingclient --name /example/P
2023/05/30 15:27:38 uplink opened, state is down
2023/05/30 15:27:38 uplink state changes to up
2023/05/30 15:27:39 100.00% D 6E7466C6A608844D 826970us
2023/05/30 15:27:39 100.00% D 6E7466C6A608844E 727565us
2023/05/30 15:27:39 100.00% D 6E7466C6A608844F 627247us
2023/05/30 15:27:39 100.00% D 6E7466C6A6088450 20012us
2023/05/30 15:27:40 83.33% E 6E7466C6A6088451 Interest expired
2023/05/30 15:27:40 85.71% D 6E7466C6A6088452 615990us
2023/05/30 15:27:40 87.50% D 6E7466C6A6088453 715573us
2023/05/30 15:27:40 88.89% D 6E7466C6A6088454 815950us
2023/05/30 15:27:40 90.00% D 6E7466C6A6088455 910512us
2023/05/30 15:27:41 90.93% D 6E7466C6A6088456 408975us
2023/05/30 15:27:41 93.33% D 6E7466C6A6088457 809220us
2023/05/30 15:27:41 91.07% D 6E7466C6A6088458 345601us
2023/05/30 15:27:41 92.31% D 6E7466C6A6088459 380746us
2023/05/30 15:27:41 92.86% D 6E7466C6A608845A 281338us
2023/05/30 15:27:41 93.79% D 6E7466C6A608845B 101116us
2023/05/30 15:27:41 94.12% D 6E7466C6A608845C 12189us
2023/05/30 15:27:43 88.89% E 6E7466C6A608845D Interest expired
2023/05/30 15:27:43 84.21% E 6E7466C6A608845E Interest expired
2023/05/30 15:27:43 80.00% E 6E7466C6A608845F Interest expired
    
```

Gambar 5 PING Router A sebagai Client

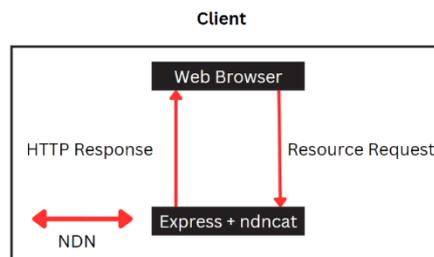
### Video Streaming

Pada bagian sub-sistem yang pertama adalah sistem *video streaming* berbasis NDN. *Video streaming* di dalam penelitian ini menjadi *server* yang akan diuji melalui beberapa *client request* ke *server*.



Gambar 6 Cara Kerja Video Streaming

Pada Gambar 6. adalah cara kerja *video streaming* yang dilakukan pada penelitian ini. *Server* akan menerima *resource request* NDN berbasis Node.JS dari *client*, Node.JS disini berfungsi untuk mengubah *resource request* menjadi *interest packet* NDN dengan bantuan *ndncat*. *Interest packet* diteruskan dari *client* hingga *server* yang di *request* mengirimkan *data packet* kepada *client*.

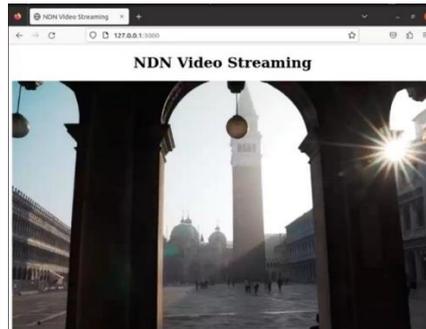


Gambar 7 Metode Client pada Video

Sesuai dengan kelebihan jaringan NDN, *data packet* bisa dikirimkan ataupun tidak dari *server*. Jika *node/router* NDN lebih dekat dengan *client* yang me-*request data*, maka *router* NDN akan langsung mengirimkan *data packet* kepada *client*. *Data packet* yang diterima oleh *client* akan diubah menjadi HTTP Response seperti pada Gambar 7. oleh *Node.js* dan dikirimkan kembali menuju *client*.

## Implementasi

Pengimplementasian *video streaming* berbasis NDN terdapat pada Gambar 8. yang diakses melalui *web browser* disisi *client*. Kinerja dari *video streaming* diukur menggunakan parameter dari *Quality of Service (QoS)* pada sisi jaringan yang terdiri dari *video startup delay*, *Round Trip Time (RTT)*, dan *throughput*. Pada sisi *router* yang dekat dengan *client* juga diterapkan pengukuran *CPU usage* untuk melihat presentase CPU yang digunakan dalam menjalankan *video streaming*.



Gambar 8 Client akses Video Streaming

Pengujian *video streaming* dilakukan dengan menjalankan di sisi *server* dan *client*. Untuk menjalankan di sisi *server* terdapat pada Gambar 9. sedangkan menjalankan disisi *client* terdapat pada Gambar 10.

```
ndn@ndn-VirtualBox:~/video-streaming$ cd 'segments-vs-chavoosh(6s)'/
ndn@ndn-VirtualBox:~/video-streaming/segments-vs-chavoosh(6s)$ bash run.sh
Run!
ndn@ndn-VirtualBox:~/video-streaming/segments-vs-chavoosh(6s)$
```

Gambar 9 Run disisi Server

```
ndn@ndn:~$ cd vidstreamndn/
ndn@ndn:~/vidstreamndn$ ls
index.html  node_modules  package-lock.json  temp  update.sh
index.js    package.json  README.md          topo
ndn@ndn:~/vidstreamndn$ sudo node index.js
```

Gambar 10 Run disisi Client

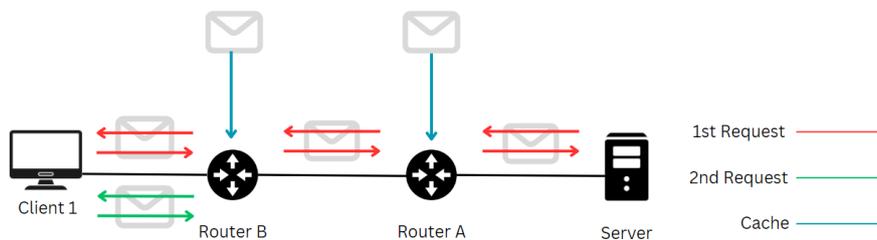
## Skema Pengujian Sistem

Pada skenario ini menggunakan topologi seperti pada Gambar 11. Topologi yang digunakan terdiri dari 1 *client*, 2 *router*, dan 1 *server*. Setiap pengujian memanfaatkan parameter yang sama pada Tabel 1. Kemudian semua perangkat yang akan diuji akan

dioperasikan menggunakan emulasi jaringan PNETLab (*Packet Network Emulator Tool Lab*).

**Tabel 1**  
**Parameter NDN-DPDK**

No	Parameter	Nilai Parameter
1	Durasi Video	1
2	Size Video	38,53 MB
3	Video Streaming sistem	NDN
4	Content Store	500 MB
5	Forwarding	NFD
6	Router	2
7	Client	1
8	CPU	8

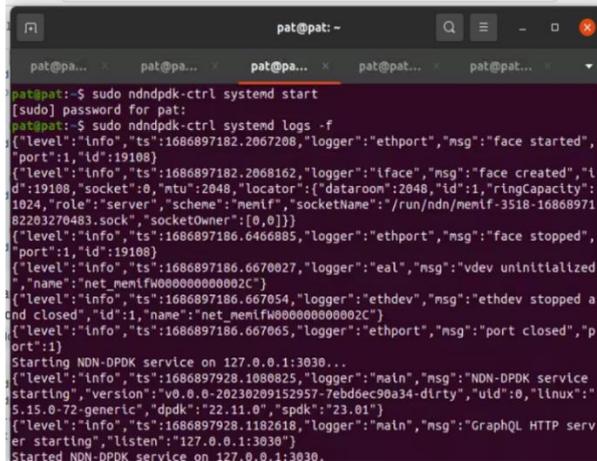


**Gambar 11** Topologi NDN-DPDK

### Proses Pengujian

#### Sebagai *Forwarder*

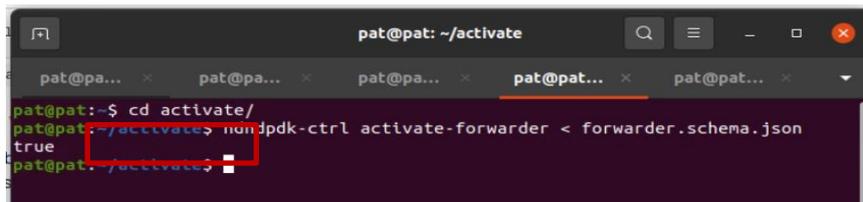
- Langkah pertama untuk melakukan pengujian NDN-DPDK adalah dengan Mengaktifkan NDN-DPDK *Service* dengan menggunakan perintah **sudo ndndpdk-ctrl systemd start**, selanjutnya adalah mengaktifkan perintah **sudo ndndpdk-ctrl systemd logs -f**, perintah tersebut bertujuan untuk melihat logs atau *service* apa saja yang terjadi saat melakukan konfigurasi pada NDN-DPDK. Perintah tersebut dapat dilihat pada Gambar 12.



```
pat@pat:~$ sudo ndndpdk-ctrl systemd start
[sudo] password for pat:
pat@pat:~$ sudo ndndpdk-ctrl systemd logs -f
{"level": "info", "ts": 1686897182.2067208, "logger": "ethport", "msg": "face started", "port": 1, "id": 19108}
{"level": "info", "ts": 1686897182.2068162, "logger": "iface", "msg": "face created", "id": 19108, "socket": 0, "mtu": 2048, "locator": {"dataroom": 2048, "id": 1, "ringCapacity": 1024, "role": "server", "scheme": "menif", "socketName": "/run/ndn/menif-3518-1686897182203270483.sock", "socketOwner": [0, 0]}}
{"level": "info", "ts": 1686897186.6466885, "logger": "ethport", "msg": "face stopped", "port": 1, "id": 19108}
{"level": "info", "ts": 1686897186.6670027, "logger": "eal", "msg": "vdev uninitialized", "name": "net_menifw000000000002c"}
{"level": "info", "ts": 1686897186.667054, "logger": "ethdev", "msg": "ethdev stopped and closed", "id": 1, "name": "net_menifw000000000002c"}
{"level": "info", "ts": 1686897186.667065, "logger": "ethport", "msg": "port closed", "port": 1}
Starting NDN-DPDK service on 127.0.0.1:3030...
{"level": "info", "ts": 1686897928.1080825, "logger": "main", "msg": "NDN-DPDK service starting", "version": "v0.0.0-20230209152957-7ebd5ec98a34-dirty", "uid": 0, "linux": "5.15.0-72-generic", "dpdk": "22.11.0", "spdk": "23.01"}
{"level": "info", "ts": 1686897928.1182618, "logger": "main", "msg": "GraphQL HTTP server starting", "listen": "127.0.0.1:3030"}
Started NDN-DPDK service on 127.0.0.1:3030.
```

Gambar 12 Mengaktifkan NDN-DPDK Service

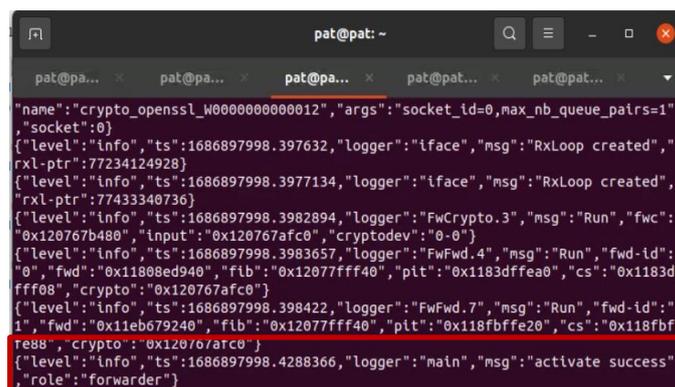
- 2 Selanjutnya, mengaktifkan NDN-DPDK sebagai *forwarder* yang berfungsi untuk mengirimkan sebuah paket antar *router* nya, dengan menggunakan perintah **ndndpdk-ctrl activate-forwarder < forwarder.schema.json** yang terlihat pada Gambar 13. Kemudian akan memiliki keluaran **true** jika berhasil.



```
pat@pat:~$ cd activate/
pat@pat:~/activate$ ndndpdk-ctrl activate-forwarder < forwarder.schema.json
true
pat@pat:~/activate$
```

Gambar 13 Mengaktifkan NDN-DPDK Sebagai Forwarder

- 3 Kemudian NDN-DPDK *Service* akan menghasilkan keluaran **activate success**, **role:forwarder** seperti pada Gambar 14 NDN-DPDK berhasil diaktifkan sebagai *forwarder*.

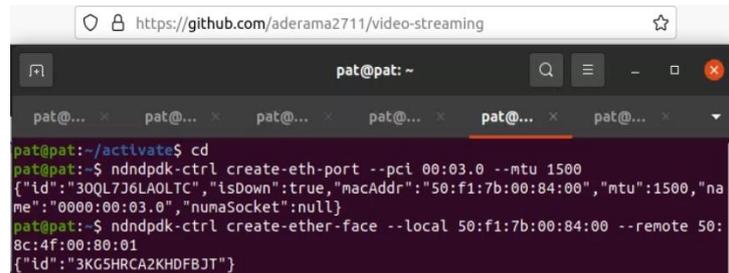


```
pat@pat:~$ ndndpdk-ctrl activate-forwarder < forwarder.schema.json
{"level": "info", "ts": 1686897998.397632, "logger": "iface", "msg": "RxLoop created", "rxl_ptr": 77234124928}
{"level": "info", "ts": 1686897998.3977134, "logger": "iface", "msg": "RxLoop created", "rxl_ptr": 77433340736}
{"level": "info", "ts": 1686897998.3982894, "logger": "FwCrypto.3", "msg": "Run", "fwd": "0x120767b480", "input": "0x120767afc0", "cryptodev": "0-0"}
{"level": "info", "ts": 1686897998.3983657, "logger": "FwFwd.4", "msg": "Run", "fwd-id": "0", "fwd": "0x11808ed940", "fib": "0x12077fff40", "pit": "0x1183dffea0", "cs": "0x1183dff08", "crypto": "0x120767afc0"}
{"level": "info", "ts": 1686897998.398422, "logger": "FwFwd.7", "msg": "Run", "fwd-id": "1", "fwd": "0x11eb679240", "fib": "0x12077fff40", "pit": "0x118fbffe20", "cs": "0x118fbfe88", "crypto": "0x120767afc0"}
{"level": "info", "ts": 1686897998.4288366, "logger": "main", "msg": "activate success", "role": "forwarder"}
```

Gambar 14 Berhasil Mengaktifkan Forwarder

- 4 Membuat *face* dan *port* pada masing masing *client* dan *server* yang digunakan, untuk membuat port dapat menggunakan perintah **ndndpdk-ctrl create-eth-port -client 00:04.0 -mtu 1500** dan untuk membuat *face* dapat menggunakan perintah **ndndpdk-ctrl create-ether-face -local 50:f1:7b:00:84:00 -remote 50:8c:4f:00:80:01**.

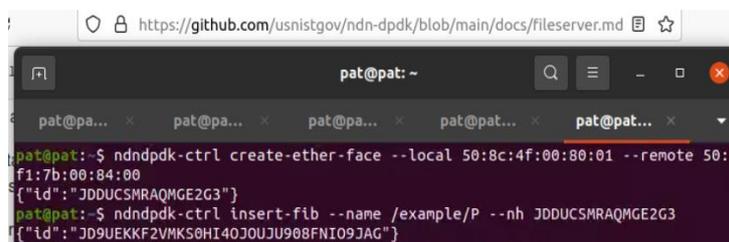
Perintah tersebut dapat dilihat pada Gambar 15. Konfigurasi tersebut dapat disesuaikan dengan perangkat masing-masing yang digunakan seperti, `macAddr`, `mtu`, dan `client`.



```
pat@pat: ~  
pat@pat:~/activate$ cd  
pat@pat:~$ ndndpdk-ctrl create-eth-port --pci 00:03.0 --mtu 1500  
{ "id": "30QL7J6LAOLTC", "isDown": true, "macAddr": "50:f1:7b:00:84:00", "mtu": 1500, "name": "0000:00:03.0", "numaSocket": null }  
pat@pat:~$ ndndpdk-ctrl create-ether-face --local 50:f1:7b:00:84:00 --remote 50:8c:4f:00:80:01  
{ "id": "3KGSHRCA2KHDFBJT" }
```

Gambar 15 Membuat Port dan Face

- 5 Tambahkan *Forwarding Information Base* (FIB) pada sisi *client*, *Forwarding Information Base* (FIB) berfungsi untuk menyimpan informasi mengenai bagaimana sebuah paket data mengirimkan paket tersebut di dalam jaringan. Ketika sebuah paket data dapat diterima oleh *router* atau *node* dalam jaringan NDN, *Forwarding Information Base* (FIB) akan mencocokkan nama konten dalam paket data tersebut dengan *entri* yang ada didalam tabel *routing* tersebut. Setelah pencocokan berhasil FIB akan menentukan rute pengiriman paket data berdasarkan nama konten sesuai permintaan. Untuk menambahkan *Forwarding Information Base* (FIB) dalam NDN-DPDK dapat menggunakan perintah **`ndndpdk-ctrl insert-fib --name /example/p --nh JDDUCSMRAQMGE2G3`**, untuk perintah `--nh` tersebut harus disesuaikan dengan hasil keluaran masing-masing. Perintah yang digunakan dapat dilihat pada Gambar 16.



```
pat@pat: ~  
pat@pat:~$ ndndpdk-ctrl create-ether-face --local 50:8c:4f:00:80:01 --remote 50:f1:7b:00:84:00  
{ "id": "JDDUCSMRAQMGE2G3" }  
pat@pat:~$ ndndpdk-ctrl insert-fib --name /example/P --nh JDDUCSMRAQMGE2G3  
{ "id": "JD9UEKKF2VMKS0HI40JOUJU908FNI09JAG" }
```

Gambar 16 menambahkan FIB pada NDN-DPDK

- 6 Lakukan pengujian antara *client* dan *server* dapat terhubung atau tidak, untuk melakukan uji coba ini pada *server* dapat menggunakan perintah **`sudo ndndpdk-godemo pingserver --name /example`**. Perintah tersebut dapat dilihat pada Gambar 17.



```
pat@pat:~$ FACEID=$(jq -n '{
  scheme: "memif",
  socketName: "/run/ndn/fileserver.sock",
  id: 0,
  role: "server",
  dataroom: 1500
}' | ndndpdctl --gqlserver http://127.0.0.1:3030 create-face | tee /dev/s
tderr | jq -r .id)
{"id": "27EUQIUM7SMETJG"}
pat@pat:~$ ndndpdctl --gqlserver http://127.0.0.1:3030 insert-fib \
> --name /fileserver --nh $FACEID
{"id": "27AUOQS2FTJQ9H9VB5CONA1QDSECID4C8K"}
```

Gambar 19 Proses FIB pada Node

- Langkah kedua, melakukan pengujian NDN-DPDK adalah dengan menghubungkan dengan *Video – Streaming* adalah dengan Mengaktifkan NDN-DPDK sebagai *file server*, Mengaktifkan NDN-DPDK sebagai *file-server* bertujuan untuk menjadikan sebuah *server*, *server* berfungsi untuk pengelolaan dan melakukan distribusi informasi antara pengirim dan penerima data. Untuk Mengaktifkan NDN-DPDK dapat menggunakan perintah awal **sudo ndndpdctl --gqlserver http://127.0.0.1:3031 systemd start**. Untuk perintah tersebut dapat dilihat pada Gambar 20.

```
pat@pat:~$ sudo ndndpdctl --gqlserver http://127.0.0.1:3031 systemd start
[sudo] password for pat:
pat@pat:~$ sudo ndndpdctl --gqlserver http://127.0.0.1:3031 activate-fileserv
er < activate/coba.json
true
pat@pat:~$
```

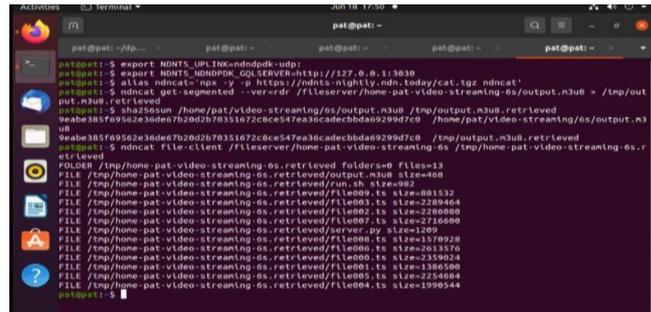
Gambar 20 NDN-DPDK sebagai File Server

- Langkah ketiga pada pengujian NDN-DPDK sebagai *Video-Streaming* adalah dengan menggunakan perintah **sudo ndndpdctl --gqlserver <http://127.0.0.1:3031> activate-fileserver < activate/coba.json**, dengan menggunakan file json yang telah dibuat dan akan memiliki keluaran **true**. Untuk perintah tersebut dapat dilihat pada Gambar 21.

```
pat@pat:~$ sudo ndndpdctl --gqlserver http://127.0.0.1:3031 systemd start
[sudo] password for pat:
pat@pat:~$ sudo ndndpdctl --gqlserver http://127.0.0.1:3031 activate-fileserv
er < activate/coba.json
true
pat@pat:~$
```

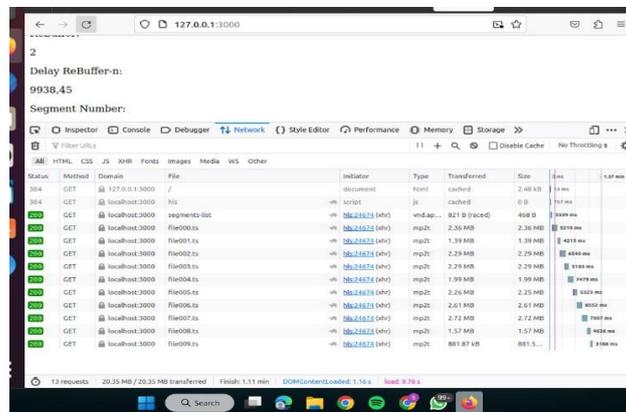
Gambar 21 Mengaktifkan NDN-DPDK sebagai File Server

- Langkah keempat adalah melakukan perintah *ndncat* yang berfungsi untuk pengguna mengirimkan dan menerima paket dalam jaringan NDN setelah dilakukan *insert fib* pada langkah sebelumnya. Untuk perintahnya dapat dilihat pada Gambar 22.



Gambar 22 NDN-CAT pada Client

5. Langkah kelima adalah dengan mencoba *video streaming* dengan menggunakan NDN-DPDK dan menganalisis hasil dari keluaran pada *video streaming*. Untuk hasil *segment list* dapat dilihat pada Gambar 23.



Gambar 23 Video Streaming dengan NDN-DPDK

### Hasil dan Pembahasan

Berdasarkan metode penelitian yang telah ditentukan di atas, dilakukanlah verifikasi pengujian menggunakan parameter QoS yang diuji sebagai berikut:

- *Round Trip Time (RTT)*  
RTT digunakan agar bisa memperkirakan jumlah waktu yang digunakan ketika *client* mentransmisikan *interest packet* atau *request packet* ke *server* hingga *response packet* dikirimkan kembali kepada *client*.

- *Throughput*  
*Throughput* adalah jumlah pengiriman data yang berhasil dikirimkan dari pengirim ke penerima dalam jangka waktu tertentu. Banyak jumlah paket yang sukses dikirimkan adalah parameter untuk suatu jaringan dikatakan baik atau buruk. *Throughput* dapat dirumuskan sebagai berikut:

$$\text{Throughput} = \frac{\text{Jumlah data dikirim}}{\text{Waktu pengiriman data}} \quad (3.1)$$

- *Video Startup Delay*

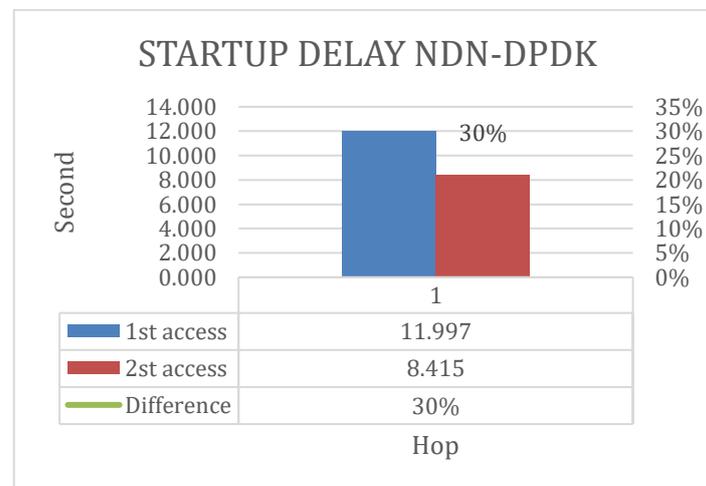
*Video Startup Delay* adalah waktu yang dibutuhkan bagi *client* untuk menunggu video diputar. Menurut (Krishnan, n.d.), penting untuk mengurangi *video startup delay* menjadi kurang dari dua detik karena peningkatan *video startup Delay* dapat mengurangi minat pengguna untuk menonton video.

- *CPU Usage*

CPU Usage adalah persentase penggunaan CPU yang digunakan dalam menjalankan beberapa aplikasi terkait penelitian *video streaming*, seperti *forwarder* dan aplikasi penyedia konten.

Kemudian didapatkan data hasil pengujian sebagai berikut.

**Startup Delay NDN-DPDK**

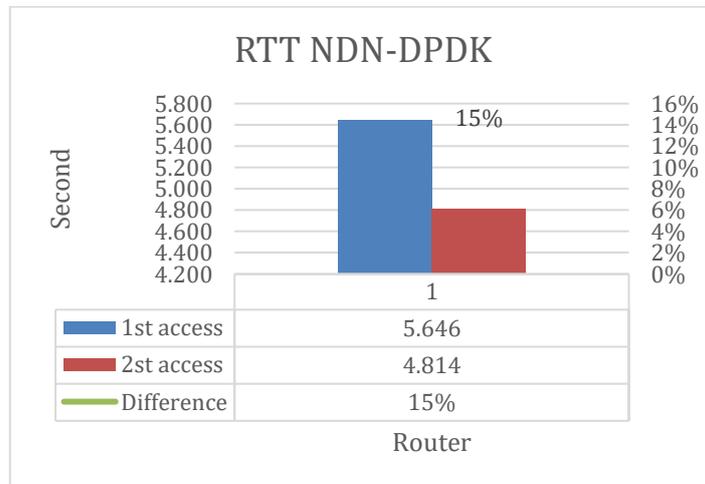


**Gambar 24** Grafik *Startup Delay* NDN-DPDK

Startup delay adalah jumlah waktu yang dibutuhkan klien untuk menunggu video diputar. Menurut [9], waktu tunda memulai video harus kurang dari 2 detik karena peningkatan waktu tunda memulai video dapat mengurangi minat klien untuk menonton video sebesar 5,8% setiap detiknya.

Berdasarkan pada hasil grafik *bar* berwarna biru menunjukkan hasil akses pertama yang mendapatkan hasil 11 detik, dan *bar* berwarna *orange* adalah hasil pengakses kedua mendapatkan hasil 8 detik. Terdapat peningkatan sebesar 30% dari pengakses pertama ke pengakses kedua. Sistem streaming video yang menggunakan basis NDN menunjukkan peningkatan kinerja yang cukup besar ketika diakses untuk kedua kalinya. Hal ini disebabkan oleh fungsi *caching router* dalam jaringan NDN. Ketika klien meminta segmen *video* untuk pertama kalinya, segmen video disimpan di penyimpanan konten router NDN terdekat. Akibatnya, jika klien kedua menginginkan segmen video yang sama, router NDN terdekat dapat mengirimkannya tanpa meneruskan interest packet ke server/produser. Hasilnya, klien kedua tidak mengalami penurunan kualitas.

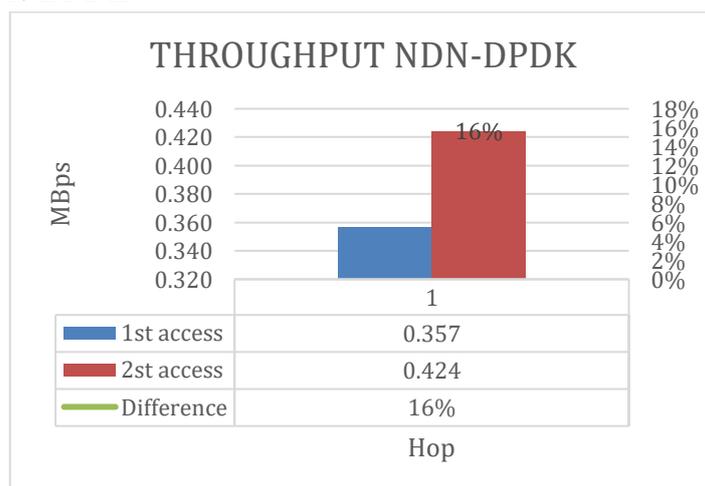
**Round Trip Time (RTT) NDN-DPDK**



**Gambar 25** Grafik RTT NDN-DPDK

RTT digunakan untuk menghitung jumlah waktu yang dibutuhkan dari saat klien mengirimkan paket interest atau paket request ke server hingga paket response dikirim kembali ke klien. Berdasarkan pada hasil grafik *bar* berwarna biru menunjukkan hasil akses pertama yang mendapatkan hasil 5 detik, dan *bar* berwarna *orange* adalah hasil pengakses kedua mendapatkan hasil 4 detik. Terdapat peningkatan sebesar 15% dari pengakses pertama ke pengakses kedua, hasil akses kedua menunjukkan adanya penurunan yang signifikan, yang disebabkan oleh mekanisme *caching* yang diterapkan dalam sistem untuk meningkatkan kinerja akses berulang terhadap *video streaming*.

**Throughput NDN-DPDK**



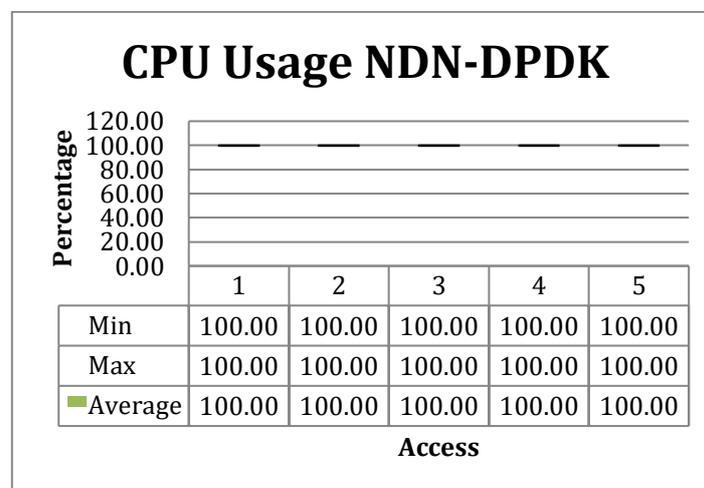
**Gambar 26** Grafik *Throughput* NDN-DPDK

Throughput adalah jumlah transmisi data yang berhasil dikirim dari sumber ke penerima dalam setiap periode waktu. Jumlah paket yang berhasil dikirim merupakan kriteria untuk menentukan apakah suatu jaringan baik atau buruk. Berdasarkan hasil

grafik yang terdapat pada Gambar 26 menampilkan hasil *throughput* yang didapatkan pada NDN-DPDK, pada bar berwarna biru itu merupakan akses pertama mendapatkan nilai *throughput* 0.357 MBps dan pada bar berwarna *orange* itu merupakan akses kedua mendapatkan nilai yang lebih besar dibandingkan dengan akses pertama.

Terdapat peningkatan sebesar 16% dari pengakses pertama 0,357 MBps ke pengakses kedua 0,424 MBps. Hal ini terjadi pada akses kedua, video yang telah diakses sebelumnya dapat disimpan di *cache* pada *router*. Sehingga, saat akses kedua dilakukan, data dapat diambil dari *cache*, yang lebih cepat daripada harus menunggu data dari *server/producer*. Hal ini meningkatkan *throughput* karena waktu yang diperlukan untuk mengambil data berkurang.

### CPU Usage NDN-DPDK



Gambar 27 Grafik CPU Usage NDN-DPDK

Persentase CPU yang digunakan dalam menggunakan berbagai aplikasi terkait, pada makalah ini menggunakan aplikasi NDN video streaming dan aplikasi penyedia konten lainnya. Akibatnya, Penggunaan CPU harus diukur untuk menentukan berapa banyak sumber daya yang dikonsumsi. Berdasarkan hasil CPU Usage NDN-DPDK yang terdapat pada Gambar 27. bahwa NDN-DPDK menggunakan CPU mencapai 100% dengan rata-rata 100%, membuktikan bahwa NDN-DPDK membutuhkan spesifikasi khusus dalam melakukan konfigurasinya. CPU Usage merupakan hasil dari penggunaan CPU yang dijalankan dengan NDN-DPDK dan dihubungkan dengan NDN *video-streaming*.

### Kesimpulan

Berdasarkan hasil penelitian dapat disimpulkan jika penggunaan NDN-DPDK jika diimplementasikan di video streaming dengan environment yang berbeda dengan penelitian sebelumnya [6] menghasilkan startup delay 11 detik pada pengakses pertama dan menghasilkan 8 detik pada pengakses kedua. Untuk RTT mendapatkan hasil 5 detik untuk pengakses pertama dan 4 detik pada pengakses kedua, dan untuk *throughput* mendapatkan hasil 0,357 MBps pada pengakses pertama dan 0,424 MBps pada pengakses

kedua. Dari nilai startup delay, RTT, dan throughput setiap akses kedua dilakukan menunjukkan bahwa penggunaan sistem video streaming berbasis NDN dengan fitur caching router dalam jaringan NDN dapat meningkatkan kinerja, mengurangi startup delay, RTT, dan meningkatkan throughput.

Nilai throughput yang didapat masih jauh dari penelitian sebelumnya, hal ini terjadi karena pada penelitian sebelumnya menggunakan resource NDN-DPDK yang sangat tinggi yaitu menggunakan berjalan server Supermicro 6039P-TXRT yang dilengkapi dengan dua Intel Xeon Gold 6240 CPU (18 core pada 2,60 GHz, dengan Hyper-Threading dinonaktifkan), 256 GB memori 2933 MHz dalam empat saluran ( $64 \times 1$  GB) hugepages telah dialokasikan untuk NDN-DPDK pada setiap NUMA socket. Sedangkan, pada penelitian ini NDN-DPDK menggunakan PNETLab berjalan pada server 8 core, 10 GB memori dalam 1 saluran (8x1 GB) hugepages telah dialokasikan untuk NDN-DPDK pada setiap NUMA socket. Hal ini mengakibatkan nilai dari startup delay, RTT, dan throughput yang didapat kurang baik dan mendapatkan hasil CPU usage 100% pada setiap aksesnya.

### BIBLIOGRAFI

- Ahdan, Syaiful, & Rahardjo, Ir Budi. (2016). Overview Keamanan pada Jaringan NDN (Named Data Networking). *Sekolah Teknik Elektro Dan Informatika, Institut Teknologi Bandung*.
- Fauzi, Fadhli, Harly, Gevin Sepria, & Hanrais, H. S. (2012). Analisis penerapan teknologi jaringan LTE 4G di Indonesia. *Majalah Ilmiah UNIKOM*, 10(2), 281–288.
- Janius, D. H. (2013). *Analisis Qos Video Streaming Pada Jaringan Wireless Menggunakan Metode HTB (Hierarchical Token Bucket) (Doctoral dissertation, Universitas Islam Negeri Sultan Syarief Kasim Riau)*.
- Krishnan, S. Shunmuga. (n.d.). *Video Stream Quality Impacts Viewer Behavior : Inferring Causality Using Quasi-Experimental Designs Categories and Subject Descriptors*.
- Krishnan, S. Shunmuga, & Sitaraman, Ramesh K. (2012). Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. *Proceedings of the 2012 Internet Measurement Conference*, 211–224.
- Melati, Seshariana Rahma, Yovita, Leanna Vidya, & Mayasari, Ratna. (2021). Caching Performance of Named Data Networking with NDNS. *2021 International Conference on Information Networking (ICOIN)*, 261–266. IEEE.
- Pepper, Robert. (2013). Cisco visual networking index (VNI) global mobile data traffic forecast update. *Mobile World Congress*.
- Rahman, Muhammad Abdi, Amarullah, Rustan, & Hidayah, Kemal. (2020). Evaluasi penerapan model pembelajaran e-learning pada pelatihan dasar calon pegawai negeri sipil. *Jurnal Borneo Administrator*, 16(1), 101–116.

- Ramadha, Ade Aditya, Yovita, Leanna Vidya, & Wibowo, Tody Ariefianto. (2022). Design and Implementation Named Data Networking-Based Video Streaming System. *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, 66–70. IEEE.
- Rukmana, Arief Yanto, Zebua, Rony Sandra Yofa, Aryanto, Didik, Nur'Aini, Indah, Ardiansyah, Welly, Adhicandra, Iwan, & Setiawan, Zunan. (2023). *DUNIA MULTIMEDIA: Pengenalan dan Penerapannya*. PT. Sonpedia Publishing Indonesia.
- Satria, Muhammad Najib Dwi, & Rahardjo, Ir Budi. (2016). *Bentuk Serangan DoS (Denial of Service) dan DDoS (Distributed Deial of Service) pada Jaringan NDN (Named Data Network)*.
- Shi, Junxiao, Pesavento, Davide, & Benmohamed, Lotfi. (2020). NDN-DPDK: NDN Forwarding at 100 Gbps on Commodity Hardware. *ICN 2020 - Proceedings of the 7th ACM Conference on Information-Centric Networking*, 30–40. <https://doi.org/10.1145/3405656.3418715>
- Wang, Lijing, Moiseenko, Ilya, & Zhang, Lixia. (2015). Ndnlive and ndntube: Live and prerecorded video streaming over ndn. *NDN, Technical Report NDN-0031*.
- Yudha, Fietyata, Kom, S., & Kom, M. (2020). *Pemanfaatan Google Cloud dan Teknik Load Balancing untuk Optimalisasi Performa Akses Halaman Web*.
- Zhang, L. (2014). *Named data networking ACM SIGCOMM Computer Communication Review*.

---

**Copyright holder:**

Syafaat Deha, Malldi Saesar, Syifa Aulia, Anggita Yaumil, Leanna Vidya, Istikmal, Tody Ariefianto (2023)

**First publication right:**

Syntax Literate: Jurnal Ilmiah Indonesia

**This article is licensed under:**

