

Analisis Performansi Modifikasi Algoritma Caching Adaptif dan Probabilistik Pada Desain Sistem *Green Named Data Network*

1st Ghozi Rafi Janitra

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

ghozirafjanitra@student.telkomuniversity.ac.id

2nd Leanna Vidya Yovita

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

leanna@telkomuniversity.ac.id

3rd Tody Ariefianto Wibowo

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

ariefianto@telkomuniversity.ac.id

Abstrak — Negara indonesia melalui presiden Indonesia menggagas adanya konsep baru dalam tatanan pembangunan ekonomi yakni adanya "Green Energy" yang menekankan pada Pembangunan ekonomi yang ramah lingkungan, penggunaan energi yang efisien, dan mengurangi gas emisi karbon. Hal ini juga berlaku pada semua bidang khususnya pada *Networking* yang sering digunakan masyarakat indonesia untuk memperoleh media informasi melalui jaringan internet. Sehingga, diperlukan arsitektur jaringan yang cepat, efisien dan ramah lingkungan. *Named Data Network* hadir menawarkan arsitektur jaringan dengan berbasis konten itu sendiri yang dapat diakses dari *node* terdekat bukan dari produsen lagi. Namun NDN masih dalam perkembangan untuk mencari strategi yang paling cepat, efisien dan selaras dengan gagasan Green tersebut. Untuk menjawab tantangan tersebut penelitian ini menggagas sistem desain Green NDN menggunakan Algoritma Caching Adaptif dan Probabilistik. Metode yang digunakan untuk mendapatkan Performansi yang lebih baik dengan melibatkan software NDN-TOOLS. Hasil dari penelitian kami menunjukkan bahwa Algoritma Caching adaptif dan Probabilistik direkomendasikan untuk mentransmisikan data berukuran besar.

Kata kunci— Green Networking, Cache, *Named Data Network*, NDN, Cache Replacement

I. PENDAHULUAN

Saat ini banyak negara berlomba untuk menjadi negara yang paling sedikit menyumbang kerusakan pada lingkungan. Hal ini telah menjadi topik hangat dalam dunia internasional. Isu ini juga berdampak pada banyak sektor. Khususnya pada bidang jaringan. Pasalnya pusat data yang terdiri dari banyak server disana mencakup teknologi dan perangkat instance yang memiliki kinerja tinggi. Dari kebutuhan tersebut Pusat data memerlukan sumber daya yang cukup besar untuk beroperasi. Dalam rentang tahun belakangan ini, banyak upaya yang dipersembahkan untuk meminimalisir pemborosan energi yang tidak diperlukan. Upaya tersebut kerap sekali dijuluki sebagai Green networking [1]. Daya tarik ini menimbulkan kebutuhan akan arsitektur jaringan yang menawarkan keunggulan kecepatan

untuk mentransmisikan data besar dan memiliki performa efisien yang cukup baik.

Desain arsitektur Jaringan *Named Data Network* hadir untuk memenuhi permasalahan ini. NDN merupakan hasil penelitian untuk mengembangkan arsitektur jaringan yang sebelumnya masih berfokus pada *host-centric* dan kini telah beralih menjadi berfokus pada *data-centric*[2]. NDN menawarkan desain arsitektur jaringan dimana router memiliki kemampuan untuk menyimpan data konten (cache).

Secara umum, NDN bertujuan untuk mengurangi beban pada lalu lintas jaringan dengan menyediakan data sedekat mungkin kepada user. Konten yang semakin dekat akan menurunkan probabilitas terjadinya pengiriman ulang kembali.

II. KAJIAN TEORI

A. Named Data Network

Named Data Network merupakan proses evaluasi dari arsitektur sebelumnya dimana paket tidak hanya sekedar mampu menentukan alamat user, melainkan juga dapat secara instan menamai objek atau konten yang dimaksudkan oleh user [3]. Dalam NDN, user akan mentransmisikan paket interest dimana berisi nama dari konten yang ingin dia peroleh. Paket tersebut akan diteruskan ke router berdasarkan prefix nama yang telah didaftarkan sebelumnya hingga mencapai produsen data. Ketika paket interest tadi sudah sampai pada produsen, produsen akan menanggapi dengan mentransmisikan konten yang diminta. Konten akan mengambil jalur kembali yang persis seperti saat paket interest dikirimkan oleh user. Namun dalam pengiriman tersebut Konten akan disimpan juga didalam caching pada router. Konten yang lebih dekat dengan User akan meminimalisir kerja sistem dan meningkatkan kecepatan sehingga sistem akan lebih efisien daripada arsitektur sebelumnya.

B. Least Recently Used (LRU)

Pada *Named Data network*, Algoritma ini termasuk dalam kategori strategi cache replacement yang paling umum digunakan. Algoritma ini berkaitan dengan sistem memori

sementara atau Cache. Least Recently Used digunakan sebagai kebijakan pengantian cache di Content Store (CS). Secara konsep LRU memastikan konten senantiasa dalam keadaan segar dengan cara menghapus konten yang paling lama tidak diakses saat CS penuh[4]. Hal ini akan memperingan kerja sistem karena sistem tidak akan mempertahankan konten yang jarang diakses dan akan selalu diperbarui dengan konten yang sering diakses.

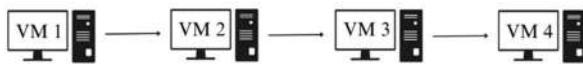
C. Caching Adaptif dan Probabilistik

Sama seperti dengan LRU, algoritma ini akan bekerja menggunakan strategi penghapusan konten yang jarang diakses berdasarkan umur dan besarnya ukuran konten. Algoritma ini berkonsep memastikan CS tidak penuh dan diisi dengan konten yang paling sering diakses. Sehingga sistem yang memiliki algoritma ini akan menghitung peluang tersimpannya sebuah konten berdasarkan nilai efektif yang paling tinggi.

III. METODE

A. Skenario penelitian

Pada penelitian kali ini, Penulis akan menguji kinerja algoritma *Energy-aware* yang bersifat Adaptif dan Probabilistik terhadap algoritma LRU dengan mentransmisikan beberapa file media berbagai macam ukuran. Media tersebut dalam format mp4 dan rar yang memiliki ukuran masing masing 83 MB, 244 MB dan 555,5 MB. Dikirimkan melalui VM 1 sebagai produsen (server), melalui VM 2 dan VM 3 sebagai node perantara serta berakhir di VM 4 sebagai user.



GAMBAR 1
Topologi simulasi jaringan NDN

Skenario ini dilakukan secara bertahap dan sistematis untuk memastikan bahwa sistem yang dirancang mampu memenuhi kinerja jaringan yang dibutuhkan serta memungkinkan efisiensi energi guna mengurangi dampak terhadap konsumsi energi.

B. NDN-TOOLS

NDN-TOOLS merupakan kumpulan alat yang digunakan penulis untuk memverifikasi dari batasan batasan parameter yang sudah ditentukan. Tools ini tersedia di github named-data.

C. Parameter

Penulis menggunakan beberapa parameter pengujian kinerja. Diharapkan dari parameter ini dapat diketahui sistem mana yang memiliki kinerja lebih efisien berikut beberapa parameter parameter yang digunakan penulis :

- *Time Elapsed*

Time Elapsed dapat didefinisikan sebagai selisih waktu dari pengiriman interest hingga data terakhir diterima oleh konsumen. *Time Elapsed* memiliki metode perhitungan sebagai berikut :

$$Time Elapsed (\Delta t) = t_2 - t_1 \quad (1)$$

di mana t_1 = waktu user mengirim interest, t_2 = waktu user menerima data. Dari pengukuran diatas batas yang diharapkan oleh penulis adalah waktu yang lebih rendah dari milik LRU yakni $\leq 25s$ [5].

- *Goodput*

Goodput adalah banyaknya *byte payload* (Data) yang benar-benar digunakan aplikasi per detik. *Goodput* memiliki persamaan sebagai berikut :

$$Goodput = \frac{8 \times Total\ byte\ diterima}{Time\ Elapsed} \quad (2)$$

Dari perhitungan diatas penulis mengharapkan sistem yang dirancang Memiliki *Goodput* yang lebih tinggi dibandingkan LRU, dengan target minimal $\geq 10,96$ Mbps [6].

- *Timeouts*

Secara umum *Timeouts* adalah kejadian ketika sistem tidak mendapatkan respon dari interest yang dikirimkan dalam jangka waktu tertentu. Sehingga sistem bisa menganggap data gagal dikirim lalu sistem akan meminta ulang dengan mengirim paket interest kembali. *Timeouts* dihitung berdasarkan jumlah interest yang tidak ditanggapi per unit waktu. Biasanya terdapat pada *Monitor log / NFD tracer*. Dari monitoring tersebut penulis mengharapkan sistem nya dapat memenuhi nilai yang lebih rendah dari $\leq 2,57\%$ dari total interest, dan lebih baik dibandingkan hasil dari baseline LRU [7].

- *Retransmisi*

Dimana ada *Timeouts* disitu ada yang namanya *Retransmisi*. *Retransmisi* tersebut adalah proses pengiriman ulang kembali paket interest dan paket data. Dimana sebelumnya peket interest sudah pernah dikirim akan tetapi tidak mendapatkan respons dalam waktu yang telah ditentukan. Berikut persamaan untuk menghitung berapa proporsi interest yang dikirim ulang :

$$\% \text{ retransmisi} = \frac{\text{Total retransmisi interest}}{\text{Total interest sent}} \times 100 \% \quad (3)$$

Retransmisi dilakukan untuk memberikan kesempatan kepada produsen untuk memberikan respon lagi terhadap interest yang diminta kembali. Penulis mengharapkan sistemnya dapat memenuhi nilai lebih kecil dibandingkan LRU dan harus berada di bawah batas $\leq 2,4\%$ [8].

- *RTT*

Secara umum seperti yang sudah diketahui bersama, RTT adalah waktu total yang dibutuhkan sistem dari user mengirimkan pekt interest hingga user menerima paket data. Pada NDN RTT memverifikasi apakah sistem tersebut mengambil paket data dari produsen atau mengambil dari cache terdekat. Hal ini disebabkan karena algoritma *caching* memastikan konten yang sering diakses tersimpan dekat dengan user untuk mengurangi waktu akses. Penulis mengharapkan dari penelitian ini nilai RTT pada algoritma

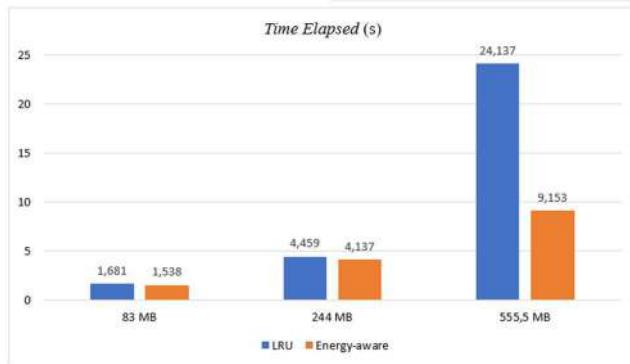
caching adaptif dan probabilistik memiliki nilai RTT yang lebih rendah dibandingkan LRU, dengan batas: min \leq 60 ms, avg \leq 74 ms, max \leq 180 ms [9].

IV. HASIL DAN PEMBAHASAN

Berikut ini adalah hasil dan analisa penelitian tentang perbandingan kinerja sistem yang menggunakan algoritma *Energy-aware* terhadap Algoritma LRU saat sistem mengirimkan informasi berbagai macam ukuran. Analisa ini akan menggunakan dan mencakup beberapa parameter yaitu : *Time Elapsed*, *Goodput*, *Timeouts*, Retransmisi dan RTT.

1. Time Elapsed

Berikut penulis sajikan gambar grafik dari *Time Elapsed*. Gambar tersebut merupakan rata-rata dari lima sesi berulang percobaan pada masing-masing file yang memiliki ukuran bervariasi menggunakan algoritma LRU dan algoritma *Energy-aware*.



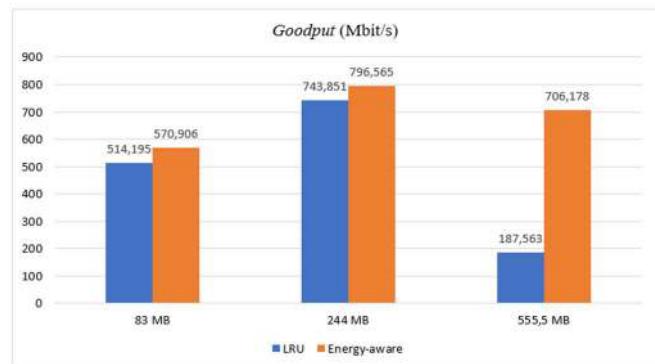
GAMBAR 2

Bagan Hasil Simulasi *Time Elapse* terhadap perubahan Ukuran Data

Berdasarkan gambar diatas dapat dilihat bahwa perubahan ukuran data berpengaruh terhadap *Time elapsed*. Semakin besar ukuran data yang ditransmisikan maka akan semakin lama waktu yang dibutuhkan untuk menyelesaikan proses transmisi. Temuan tersebut menghasilkan keberhasilan LRU dalam menyelesaikan transmisi data memiliki rata-rata waktu 10,09 detik, sedangkan *Energy-aware* memperoleh rata-rata waktu 4,94 detik. Dengan demikian, strategi algoritma *Energy-aware* mampu mempercepat waktu akses rata-rata sebesar 51,03% dibandingkan LRU. Temuan ini senada dengan penelitian Gameiro yang menyebutkan bahwa hasil *Time elapsed* yang baik yakni \leq 25s [5].

2. Goodput

Berikut penulis sajikan gambar grafik dari *Goodput*. Gambar tersebut merupakan rata-rata dari lima sesi berulang percobaan pada masing-masing file yang memiliki ukuran bervariasi menggunakan algoritma LRU dan algoritma *Energy-aware*.



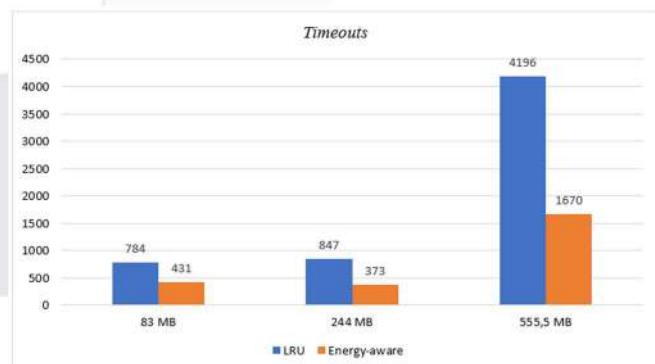
GAMBAR 3

Bagan Hasil Simulasi *Goodput* terhadap perubahan Ukuran Data

Berdasarkan gambar diatas dapat dilihat bahwa perubahan ukuran data berpengaruh terhadap *Goodput*. Akan tetapi pada data besar algoritma LRU mengalami penurunan. Hal ini disebabkan karena pada data yang memiliki ukuran besar akan menyebabkan *cache* penuh oleh data yang jarang diakses. Sehingga sistem akan mengirimkan kembali tanpa disimpan dalam *cache*. Proses mengirimkan kembali ini akan berdampak pada sistem, karena akan meningkatkan *overhead* dan retransmisi. Sebaliknya pada *Energy-aware* algoritma ini akan mengevaluasi data berdasarkan ukuran. Sehingga ketika data memiliki ukuran yang besar maka sistem akan lebih mampu menjaga efisiensi kapasitas *cache*. Hal ini juga dapat mengurangi *overhead* dan retransmisi karena data tidak senantiasa harus diambil dari *server*. Sehingga dari adaptifnya tersebut akan meningkatkan *Goodput* secara signifikan yakni hingga 276,50% pada ukuran data terbesar.

3. Timeouts

Berikut penulis sajikan gambar grafik dari *Timeouts*. Gambar tersebut merupakan rata-rata dari lima sesi berulang percobaan pada masing-masing file yang memiliki ukuran bervariasi menggunakan algoritma LRU dan algoritma *Energy-aware*.



GAMBAR 4

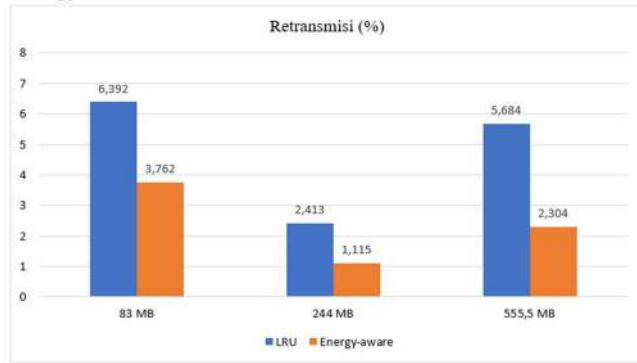
Bagan Hasil Simulasi *Timeouts* terhadap perubahan Ukuran Data

Berdasarkan gambar diatas dapat dilihat bahwa perubahan ukuran data berpengaruh terhadap jumlah *Timeouts*. Semakin besar ukuran data yang ditransmisikan maka akan semakin meningkatkan jumlah *Timeouts*. Perbedaan jumlah *Timeouts* pada masing-masing algoritma disebabkan oleh cara untuk dapat mempertahankan efisiensi *cache* saat menghadapi permintaan data dengan ukuran yang bervariasi. Temuan tersebut menghasilkan keberhasilan LRU

dalam mengurangi *timeouts* rata-rata sebesar 1942,47, sedangkan strategi Energy-aware mencatat rata-rata sebesar 824,64. Dengan demikian, strategi Energy-aware mampu menurunkan jumlah timeouts sebesar 57,55% dibandingkan dengan LRU.

4. Retransmisi

Berikut penulis sajikan gambar grafik dari Retransmisi. Gambar tersebut merupakan rata-rata dari lima sesi berulang percobaan pada masing-masing file yang memiliki ukuran bervariasi menggunakan algoritma LRU dan algoritma Energy-aware.

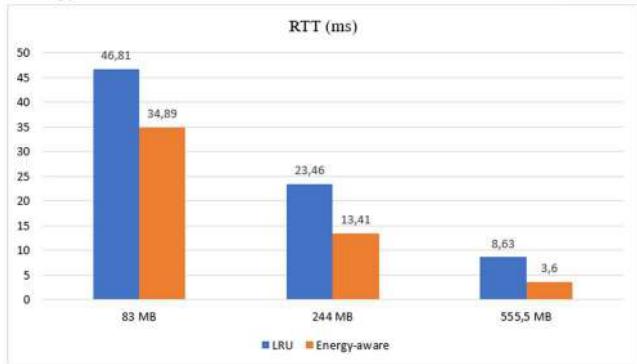


GAMBAR 5
Bagan Hasil Simulasi Retransmisi terhadap perubahan Ukuran Data

Berdasarkan gambar diatas dapat dilihat bahwa variasi ukuran data juga berpengaruh terhadap besar persentase retranmisinya. Temuan tersebut menghasilkan Keberhasilan algoritma LRU dalam pengujian ini menghasilkan rata-rata 4.83 retrasmisi, sedangkan strategi Energy-aware mencatat rata-rata 2.39 retrasmisi. Dengan demikian, strategi Energy-aware mampu mengurangi jumlah retrasmisi sebesar 50,52% dibandingkan dengan LRU. Temuan ini memenuhi harapan penulis yakni $\leq 2,4\%$ yang berdasarkan pada penelitian Pentikousis [8].

5. RTT

Berikut penulis sajikan gambar grafik dari RTT. Gambar tersebut merupakan rata-rata dari lima sesi berulang percobaan pada masing-masing file yang memiliki ukuran bervariasi menggunakan algoritma LRU dan algoritma Energy-aware.



GAMBAR 6
Bagan Hasil Simulasi RTT terhadap perubahan Ukuran Data

Berdasarkan gambar diatas dapat dilihat bahwa variasi ukuran data memengaruhi nilai RTT pada kedua strategi

caching yang diuji. Secara keseluruhan, strategi Energy-aware mampu menurunkan nilai RTT secara signifikan dibandingkan dengan LRU pada semua skema ukuran file. Rata-rata RTT pada strategi LRU adalah 26,30 ms, sedangkan Energy-aware rata-rata hanya 17,31 ms, menunjukkan pengurangan latensi sebesar 34,18%.

6. Kapasitas cache

Terlepas dari perbedaan beberapa parameter sebelumnya, terdapat perbedaan mendasar dalam cara kedua algoritma tersebut menangani kapasitas *cache*. Algoritma LRU ketika ukuran total data melebihi kapasitas *cache*, sistem tidak akan menyimpan segmen apapun dari data tersebut. Akibatnya, sistem akan meminta kepada produsen untuk mengirim kembali menuju user, hal ini menyebabkan meningkatnya waktu pengambilan serta dapat memungkinkan potensi terjadinya *timeout* dan retransmisi.

Sebaliknya pada algoritma Energy-aware meskipun jumlah segmen data yang diterima *cache* melebihi kapasitas maksimum *cache*, algoritma ini akan tetap mampu menyimpan sebagian besar segmen hingga kapasitas *cache* terpenuhi. Segmen yang tidak dapat disimpan dalam *cache* akan tetap diminta dari *server*, hal ini menunjukkan algoritma Energy-aware fleksible dalam hal penyimpanan data. Namun dengan adanya penyimpanan sebagian ini oleh sistem tetap menjadi perhatian karena mampu memberikan keuntungan dalam mempercepat waktu pengambilan data dan mengurangi beban protokol akibat retransmisi berulang. Temuan ini menunjukkan bahwa algoritma Energy-aware lebih efektif dalam memanfaatkan kapasitas dalam *cache* yang tersedia untuk meningkatkan efisiensi pada sistem secara menyeluruh.

V. KESIMPULAN

Dari penelitian ini penulis dapat membuktikan bahwa implementasi Algoritma LRU dan Algoritma Adaptif dan Probabilistik sukses dijalankan pada Virtual Machine, dengan menggunakan NDN-TOOLS yang telah disediakan parameter yang sudah ditentukan dapat diukur dan dimonitoring. Hasil penelitian memberikan bukti bahwa keunggulan kinerja Algoritma Adaptif dan Probabilistik diatas dari Algoritma LRU melalui efisiensi waktu dicerminkan dengan *Timed elapsed* yang makin cepat, *Goodput* mencerminkan efisiensi komunikasi dilihat dari transmisi data tanpa *overhead* yang berlebihan, sehingga sistem terhindar dari kerja yang berlebih akibat dari gagalnya transmisi saat komunikasi sedang berlangsung. Dimana dari hal tersebut dapat berdampak terhadap mengurangnya beban energi selama proses komunikasi. Sementara itu, *timeouts* dan retrasmisi mencerminkan kegagalan dalam proses komunikasi antar user dan *server* sehingga menyebabkan upaya pengiriman ulang, yang pada akhirnya akan menambah konsumsi beban energi. RTT juga memiliki kontribusi secara tidak langsung, karena semakin tinggi nilainya, semakin besar pula delay yang dapat memperlama aktivitas komunikasi dan meningkatkan penggunaan sumber daya. Sebagai penutup, kemampuan adaptif dari algoritma Energy-aware ketika kapasitas *cache* penuh juga merupakan suatu keunggulan yang dimilikinya dibanding dengan Algoritma LRU.

REFERENSI

- [1] A. P. Bianzino, C. Chaudet, D. Rossi and J. -L. Rougier, "A Survey of Green Networking Research," in IEEE Communications Surveys & Tutorials, vol. 14, no. 1, pp. 3-20, First Quarter 2012, doi: 10.1109/SURV.2011.113010.00106.
- [2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in ACM CoNEXT, 2009
- [3] Satria, Muhammad & Ilma, Farchah & Syambas, Nana. (2017). Performance comparison of named data networking and IP-based networking in palapa ringnetwork.43-48.doi:10.1109/ICWT.2017.8284136.
- [4] H. Dai, B. Liu, H. Yuan, P. Crowley and J. Lu, "Analysis of tandem PIT and CS with non-zero download delay," IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA, USA, 2017, pp. 1-9, doi: 10.1109/INFOCOM.2017.8057191.
- [5] L. Gameiro, C. Senna, and M. Luís, "Insights from the Experimentation of Named Data Networks in Mobile Wireless Environments," *Future Internet*, vol. 14, no. 7, p. 196, Jun. 2022. [Online]. Available: <https://doi.org/10.3390/fi14070196>.
- [6] Georgia Institute of Technology, "Performance of Application-Specific Buffering Schemes for Active Video Connections," Technical Report GIT-CC-98-17, 1998. [Online]. Available: <https://sites.cc.gatech.edu/projects/canes/papers/git-cc-98-17.pdf>.
- [7] T. Ma et al., "Interest Timeout Analysis in Vehicular NDN," in *Proc. Vehicular Networking Conference*, 2019.
- [8] K. Pentikousis, H. Badr, and A. Andrade, "A comparative study of aggregate TCP retransmission rates," VTT Technical Research Centre of Finland, 2007. [Online]. Available: <https://www.vtt.fi>.
- [9] S. Shakkottai, R. Srikant, N. Brownlee, A. Broido, and K. C. Claffy, "The RTT Distribution of TCP Flows in the Internet and its Impact on TCP based Flow Control," CAIDA, SDSC, 2002. [Online]. Available: <https://www.caida.org>.