CMPE 49F Project Description

I. Introduction

The drastically increasing multimedia content consumption by the Internet leads to high network traffic. This necessitates the improvement of system capacity and alleviation of latency. Simultaneously, the system energy compensation requirement arises. The content request volume for video content traffic emerges content-centric design of networks to resolve these technical issues [1]. The classical networking techniques are revised for improving the content dissemination in terms of the capacity, latency and energy. The Device-to-device (D2D) technique is a promising enabler for boosting the system capacity while with the short distance transmission technique pursues the energy efficiency [2]. The cognitive radio technique also has a potential for the capacity improvement in 5G networks [3]. Information-Centric Network (ICN) directly requests multimedia from the network instead of the classical IP-based approach [4]. Thus, the focal is put onto the multimedia content. For the substantial system capacity expansion and energy expenditure reduction motives, the layering concept for video contents can be utilized [5]. The content layering improves the scalability [6]. Some content layers are more lucrative in terms of magnifying the network gains. For instance, the base layer is essential since without it videos cannot be displayed [7], [8]. From the content-convergent network orientation aspect, the amalgamation of cognitive radio, D2D paradigm for the dissemination of layered multimedia contents is functional.

Caching is a fundamental capability for improving the system energy expenditure, reducing the latency and amplifying the capacity. Thereof, you will focus on the caching aspect and propose your own caching technique in D2D networks for layered multimedia content transmission. You will implement the Least Recently Used (LRU) and Least Frequently Used (LFU) techniques as baseline profiles. Then you will propose your own caching technique and compare with the baseline profiles.

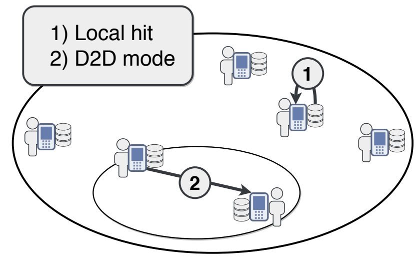

Fig. 1: Network Structure

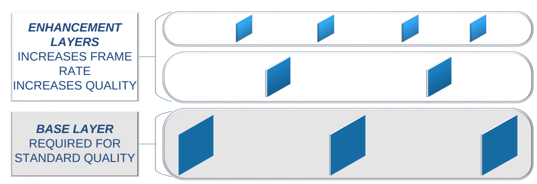

Fig. 2: Layered content model.

II. System Model

In this section, we give the model of our network system. In our system environment, users are dispersed in the spatial domain. For the user locality Poisson Point Process (PPP) is a commonly utilized distribution [9], [10]. In our system, users are distributed according to PPP with mean density $\lambda_{users} = 0.0015 \frac{user}{m^2}$ in a region with the radius $R = 300m$. Users hold devices with the repositories that are capable of storing contents. These devices can share video content with each other with the help of D2D paradigm. When a content is requested, first the requester will check its local cache. If not found, it will try to get the service in D2D operation mode. It will fetch the requested content from the closest device that stores the requested video content. The failed requests are not stored in a buffer for prospective retrials.

The content layering is used for the sake of temporal and spatial scalability [11]. The content layers are named as *i) base* and *ii) enhancement*. The base layer is the standard quality part. The enhancement layer is the extra bits requires for high quality viewability. Without the base layer, enhancement layer is of no use. i.e. You can not view the video content. In our system 100 distinct contents exist. The popularity of videos contents vary. Consider the Youtube videos. The trending videos are requestly widely whereas some videos are rarely viewed. For the condensation of the popularity characterization Zipf distribution is utilized [12], [13]. The video popularities follow the Zipf distribution with parameter $s = 0.8$. Say there are N=100 many different contents, then the popularity of the k^{th} content is $\frac{\frac{1}{k^s}}{\sum_{n=1}^{N}(\frac{1}{n^s})}$. Each content in our system consists of the aforementioned layers. The mean base content size is 25 Mbits while the mean enhancement content size is 5 Mbits [5]. Both layer content sizes follow exponential distribution with the given mean values. The cache capacities of user devices are 1 Gbits. Initially you fill the caches of all users with contents based on their popularities. You can propose your own cache reservation mechanisms for different content layers. This is your design choice that you need to explicitly explain in your project report.

Our devices operate in the terrestrial channel opportunistically (cognitive radio users/secondary users). i.e. The channel belongs to other users (primary users). If these users are active in some frequency our users cannot access the network at that frequency. You can think of the primary users as the house owner of a frequency and our cognitive (secondary) users as the renter of the frequency. Note that in the cognitive radio when a primary user (PU) starts transmission in a frequency the secondary user (SU) should preempt the frequency immediately. You can assume perfect sensing, so no collisions of PU and SUs occur. There exist 10 terrestrial frequencies each of bandwidth $B = 2$ MHz. The initial frequency f_1 is at 700 MHz. The second frequency f_2 is at 700+2 MHz. Continuing this way i^{th} frequency is calculated according to the formula $700 + (i-1) \cdot 2$ MHz. You will calculate the capacity of any frequency f_i under additive White Gaussian Noise with Shannon's capacity formula $C = B \cdot log_2(1 + \frac{P_{rec}}{B \cdot N_0})$. Here $B = 2$ MHz, $N_0 = 1.6e^{-19} \frac{Watt}{Hz}$ and $P_{rec}^{SU} = \frac{2.6e^{+13}}{(4 * \pi * f_i * d)^2}$, where d is the distance between the transmitter and the receiver of the D2D operation. For primary users you can take $P_{rec}^{PU} = \frac{4.7e^{+13}}{(4 * \pi * f_i * 150)^2}$. The user arrivals into the system are exponentially distributed with mean $\lambda_{PU} = 1 \frac{user}{sec}$ for primary users and $\lambda_{SU} = 1.5 \frac{user}{sec}$ for our secondary users. Parametrize user request rates and other given parameters (R, s, mean content sizes etc.) so that in need of a tune operation you will not lose time. Create a parameter list function and read all the system parameters from that function. An example channel history is given in the Fig. 3. Note that a frequency can be used by one service at a time. You can assume that PUs request only base layer contents. The rate of SUs that request both base and enhancement layer is a system parameter $p_{HQ} = 0.5$.

Say a PU request arrives for the content service environment at frequency f_i, $\ i \in \{1, 2, ..., 10\}$ with rate $\frac{\lambda_{PU}}{10}$.

- If that frequency f_i is idle (empty), just start the service.
- Else if frequency f_i is not idle but a PU exists, then PU is blocked.
- Else if that frequency f_i is not idle but a SU exists, cut the SU operation (base or enhancement) at that frequency and start the PU operation. For the ceased SU:
  - If at least one idle frequency exists at the time being, then continue the operation from some idle frequency (shown by light orange in Fig. 3).
  - If no idle frequency exists for the continuation of the SU operation the SU is dropped (shown by dark orange in Fig. 3).

This is how you will process the PU arrivals to the system. For the SUs the arrival processing mechanism is as follows:

- If a SU request arrives to the content service environment a random idle frequency will be selected for the service.
- If no idle frequency exists that request is blocked.
- Else if some idle frequency is found and a SU requests
  - base layer only then start service at that idle frequency
  - both base and enhancement layers, you can use two different techniques:
    - First serve the base layer and after successful completion of the base layer transmission start serving the enhancement layer immediately at the same frequency as shown for f_2 in Fig. 3.
    - Start the services for both base and enhancement components simultaneously at different frequencies shown in the first services for f_3 and f_4 in Fig. 3.

You need to keep track of channel status and log service, block and drop events of PUs (of type base) and SUs (of type base and enhancement). Think of your data structures for the channel status and events.

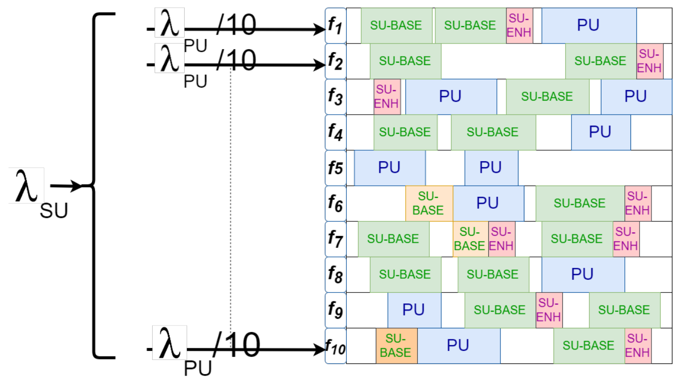

Fig. 3: Channel history example.

III. Caching Algorithms

In the literature there exists a pletora of caching algorithms. [14] proposed ProbCache algorithm in decentrized ICN environment. This algorithm caches contents in routers based on the paths and aims to reduce caching redundancy of in-network routers. In [15], they have devised a request score and chunk based caching (CC) technique for streaming videos. The request scoring rewards popular contents and they are more likely to be cached. For the chunking dimension they utilize the fact that if a chunk of a video is viewed, the prospective chunks will be requested as well. Suksomboon et. al. propose PopCache in Content-Centric Networks (CCNs) that stores popular contents close to the requesters and distribute contents along routers in a less redundant way [16]. [17] proposes a caching policy that augments the number of chunks to be cached in a content routers exponentially with increased access rate. In the literature there are also caching studies in D2D network [18], [19]. You will read these papers to understand the caching problem and how new techniques are evolved. In three weeks you will submit a brief summary of these caching papers that you have read (you can find other papers as well) and also propose your caching algorithm in cognitive D2D networks. Please do not copy paste one of these techniques. This will result in zero. Write down how your method differs from the techniques that you have read. Also investigate the time complexity of your proposed algorithm and mathematically extrapolate the results in big-O notation in your first report in three weeks (28^{th} of November) that is 30% of the project grade.

Next, you will implement your algorithm, LRU and LFU. You will compare your algorithm with the LRU, LFU caching techniques in cognitive D2D networks. You will investigate your proposed caching algorithm in terms of the performance metric *cache hit rate, average latency*. You will precisely define these metrics in your report and also list your notations in a Table. There will be a demo session where you will show your program code and your results. Please submit both your report and code in 19^{nd} of December before the demo session that will be decided soon. Your final submission and demo constitute the 70% of your project grade. You can implement your code either in Matlab or Phyton.

Let us briefly summarize what you will be doing in your project.

- Investigate current literature to learn about key concepts and the state-of-the-art for caching mainly the papers [14,15,16,17,18,19].
- Propose your own caching technique and tell how it differs from the papers [14,15,16,17,18,19].
- Do the time complexity analysis of your caching technique
- Submit the first report on the 28^{th} of November
- Identify data structures that you will use in your project for channel management
- Implement your caching algorithm, LRU and LFU
- Visualize cache hit rate and average latency values of your caching algorithm, LRU and LFU
- Submit the final report on the 19^{th} of December
- Attend the demo session and show your code and results

References

[1] G. Gür, "Energy-aware cache management at the wireless network edge for information-centric operation," *Journal of Network and Computer Applications*, vol. 57, pp. 33 – 42, 2015.

[2] S. Sinem Kafılоğlu, G. Gür, and F. Alagöz, "Analysis of content-oriented heterogeneous networks with D2D and cognitive communications," *arXiv*, 2018.

[3] C. Wang, F. Haider, X. Gao, X. You, Y. Yang, D. Yuan, H. M. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir, "Cellular architecture and key technologies for 5g wireless communication networks," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 122–130, February 2014.

[4] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagner, "Adaptive multimedia streaming in information-centric networks," *IEEE Network*, vol. 28, no. 6, pp. 91–96, Nov 2014.

[5] G. Gür and S. Kafiloğlu, "Layered content delivery over satellite integrated cognitive radio networks," *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 390–393, June 2017.

[6] J. . Ohm, "Advances in scalable video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 42–56, Jan 2005.

[7] S. Ullah, T. LeAnh, A. Ndikumana, M. G. R. Alam, and C. S. Hong, "Layered video communication in icn enabled cellular network with d2d communication," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Sept 2017, pp. 199–204.

[8] J. . Ohm, "Advances in scalable video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 42–56, Jan 2005.

[9] D. Malak and M. Al-Shalash, "Optimal caching for device-to-device content distribution in 5g networks," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 863–868.

[10] S. Kusaladharma and C. Tellambura, "Performance characterization of spatially random energy harvesting underlay d2d networks with primary user power control," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7.

[11] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial," *IEEE Communications Surveys Tutorials*, vol. 6, no. 3, pp. 58–78, Third 2004.

[12] D. Malak and M. Al-Shalash, "Optimal caching for device-to-device content distribution in 5g networks," in *2014 IEEE Globecom Workshops (GC Wkshps)*, Dec 2014, pp. 863–868.

[13] M. Emara, H. ElSawy, S. Sorour, S. Al-Ghadhban, M. Alouini, and T. Y. Al-Naffouri, "Optimal caching in multicast 5g networks with opportunistic spectrum access," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–7.

[14] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ser. ICN '12. New York, NY, USA: ACM, 2012, pp. 55–60. [Online]. Available: http://doi.acm.org/10.1145/2342488.2342501

[15] D. Hong, D. De Vleeschauwer, and F. Baccelli, "A chunk-based caching algorithm for streaming video," in *NET-COOP 2010 - 4th Workshop on Network Control and Optimization*, Gent, Belgium, Nov. 2010, session 05 : Streaming applications.

[16] K. Suksomboon, S. Tarnoi, Y. Ji, M. Koibuchi, K. Fukuda, S. Abe, N. Motonori, M. Aoki, S. Urushidani, and S. Yamada, "Popcache: Cache more or less based on content popularity for information-centric networking," in *38th Annual IEEE Conference on Local Computer Networks*, Oct 2013, pp. 236–243.

[17] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack, "Wave: Popularity-based and collaborative in-network caching for content-oriented networks," in *2012 Proceedings IEEE INFOCOM Workshops*, March 2012, pp. 316–321.

[18] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5g systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, February 2014.

[19] H. Zhu, Y. Cao, W. Wang, B. Liu, and T. Jiang, "Qoe-aware resource allocation for adaptive device-to-device video streaming," *IEEE Network*, vol. 29, no. 6, pp. 6–12, Nov 2015.