CMPE 49F Project Performance Evaluation

You will investigate the caching algorithms in terms of cache hit and latency. As our requests are comprised of different layers of videos we need a more elaborate calculation. You will be logging every event of all users and these will be utilized for the performance calculation. Here is an example log format and several example entries of event logs for different users and event types in Table. I.

You will not disperse the PUs and not simulate caching for PUs. They will be there only for services in the channel. The event types for PUs will be either SERVED or BLOCKED. Note that when a PU starts service, it will not be dropped as it has highest priority among the users. You will also ignore the local hit scenario as caching for PUs is not our main focus.

We will utilize only **SU logs**. Note that a SU content request can occur in two cases: *A) SU requests standard quality video (base layer only)*, *B) SU requests high quality video (both base layer and enhancement layer)*. The * marked cases will contribute to the latency. For base local hits you will take latency as 0.25 sec and for enhancement local hits 0.05 sec. You will calculate the latency for D2D services according to the service capacity between the receiver and the transmitter. You will utilize the Shannon's capacity formula $C = B \cdot log_2(1+\frac{P_{rec}}{B \cdot N_0})$ for getting the transmission rate in bits per second (bps). You will get your content's size according to exponential distribution with the 25 Mbits mean for base and 5 Mbits for enhancement layers. For each D2D transmission of any type (base/enhancement), by dividing the content size over the service capacity you will get the service duration in seconds ($t = \frac{size}{C}$). For calculating the average latency, you will accumulate each t of SUs for D2D services and local hits as T_{all}. Then you will count the number of SU services of type A that are served either locally (B_L) or via D2D (B_{D2D}) as $count_{base}$. You will also count the number of B-type SU services as $count_{enh}$.

- For a HQ request when the base is received successfully as of event type B_L or B_{D2D} but enhancement fails then increment $count_{enh}$ by one.
- For a HQ request when the base is received successfully as of event type B_L or B_{D2D} and enhancement (E_L or E_{D2D}) is also successful increment the $count_{enh}$ *only* once. Do not increment the counter once for the base transmission and once for the enhancement transmission.

Eq. 1 gives the average latency of the system.

$$latency = \frac{T_{all}}{count_{base} + count_{enh}} \tag{1}$$

For the *case A*, there are four possible event types:

1) Base Blocked (B_B)
2) Base Dropped (B_D)
3) Base Local Hit (B_L) *
4) Base Served by D2D Technique (B_{D2D}) *

For the *case B*, the event types are given as follows:

1) Base Blocked (B_B)
   a) Enh Blocked (E_B)
2) Base Dropped (B_D)
   a) Enh Dropped (E_D)
3) Base Local Hit (B_L) *
   a) Enh Blocked (E_B)
   b) Enh Dropped (E_D)
   c) Enh Local Hit (E_L) *
   d) Enh Served by D2D Technique (E_{D2D}) *
4) Base Served by D2D Technique (B_{D2D2}) *
   a) Enh Blocked (E_B)
   b) Enh Dropped (E_D)
   c) Enh Local Hit (E_L) *
   d) Enh Served by D2D Technique (E_{D2D}) *

Now, let us focus on the definition of the local hit rate for SUs. We will count the local hit rate for HQ and SQ requests of SUs separately. For the SQ local hit rate of SUs $p_{loc}^{SQ}(base)$, you will count all SQ SU events of types B_B, B_D, B_L and B_{D2D} as $count_{SQ}$. You will also count the SQ B_L type of SU events as $count_{SQ}^{loc}$. By $\frac{count_{SQ}^{loc}}{count_{SQ}}$, you get the $p_{loc}^{SQ}(base)$ value. For the HQ local hit rate of SUs, you calculate three different local hit scenarios separately:

- The base is a local hit $p_{loc}^{HQ}(base)$
- The enhancement is a local hit given that base is a local hit $p_{loc}^{HQ}(enh|base_{loc})$
- The enhancement is a local hit given that base is served via D2D technique $p_{loc}^{HQ}(enh|base_{D2D})$

First you will calculate $p_{loc}^{HQ}(base)$. You will calculate the number of base HQ SU events of all types B_B, B_D, B_L and B_{D2D} as $count_{HQ}^{base}$. Then you will get the number of HQ SU events of type B_L that are served locally as $count_{HQ}^{base(loc)}$. Eq. 2 gives the probability of base HQ SU requests being served locally.

$$p_{loc}^{HQ}(base) = \frac{count_{HQ}^{base(loc)}}{count_{HQ}^{base}} \tag{2}$$

Next, you will calculate the $p_{loc}^{HQ}(enh|base_{loc})$. You know the number of base SU events for HQ requests that are served locally as $count_{HQ}^{base(loc)}$. For these type of base HQ SU services you will count the number of corresponding enhancement layers that are served locally as $count_{HQ}(enh(loc)|base(loc))$. Eq. 3 gives the probability that the enhancement chunk of HQ SU requests will be served locally where the corresponding base chunks are also served locally.

$$p_{loc}^{HQ}(enh|base_{loc}) = \frac{count_{HQ}(enh(loc)|base(loc))}{count_{HQ}^{base(loc)}} \tag{3}$$

TABLE I: Event log format and example entries.

EV ID	PREV EV ID	EVENT TYPE	REQ TYPE	USE	LAYER	START TIME	END TIME	TX DEVICE	REC DEVICE	FREQUENCY
1	1	SERVED	SQ	PU	BASE	0.2 sec	2.57 sec	NON	NON	f_2
21	21	BLOCKED	SQ	PU	BASE	4 sec	4 sec	NON	NON	NON
52	52	DROPPED	SQ	SU	BASE	8.1 sec	8.9 sec	dev_2	dev_{50}	f_8
57	57	LOCAL HIT	SQ	SU	BASE	8.75 sec	9 sec	dev_{75}	dev_{75}	NON
95	95	LOCAL HIT	HQ	SU	BASE	14.62 sec	14.87 sec	dev_{135}	dev_{135}	NON
97	95	DROPPED	HQ	SU	ENH	14.87 sec	17.5 sec	dev_8	dev_{135}	f_1

Finally, you will calculate the $p_{loc}^{HQ}(enh|base_{D2D})$. You will calculate the number of base SU events for HQ requests that are served by the D2D paradigm as $count_{HQ}^{base(D2D)}$. For these type of base HQ SU services you will count the number of corresponding enhancement layers that are served locally as $count_{HQ}(enh(loc)|base(D2D))$. Eq. 4 gives the probability of enhancement chunks of HQ SU requests will be served locally where the corresponding base chunks are served via D2D.

$$p_{loc}^{HQ}(enh|base_{D2D}) = \frac{count_{HQ}(enh(loc)|base(D2D))}{count_{HQ}^{base(D2D)}} \tag{4}$$

You will calculate $latency$, $p_{loc}^{SQ}(base)$, $p_{loc}^{HQ}(base)$, $p_{loc}^{HQ}(enh|base_{loc})$ and $p_{loc}^{HQ}(enh|base_{D2D})$ values with at least $N_{SEED} = 10$ many simulation runs for LRU. Then you will take the averages of these N_{SEED} distinct runs for each metric. You will apply the same procedure for LFU and your caching algorithm and report their results. An example result for average latency of LRU, LFU and your caching algorithm is given in Fig. 1. Fig. 2 is the average $p_{loc}^{SQ}(base)$ result for 3 different caching techniques. In your final report there will be 5 such figures for $latency$, $p_{loc}^{SQ}(base)$, $p_{loc}^{HQ}(base)$, $p_{loc}^{HQ}(enh|base_{loc})$ and $p_{loc}^{HQ}(enh|base_{D2D})$ metrics.

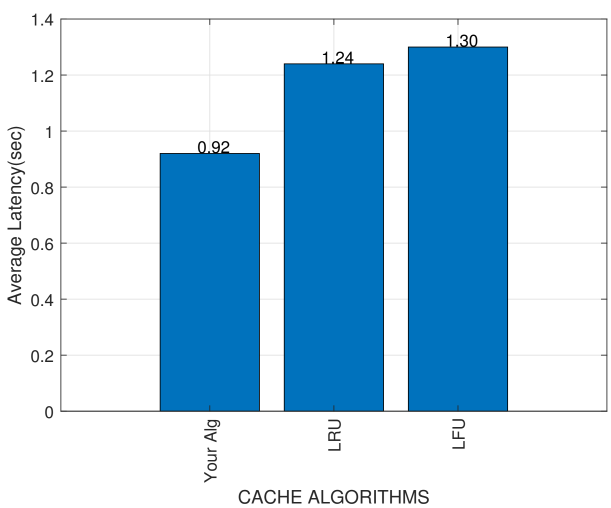

Fig. 1: Average Latency Results

Note that for the simulation duration you will take at least $T_{sim} = 1200sec$ at each seed of each caching algorithm. There will be $N_{dev} = 500$ devices. For each SU device, you can take the arrival rate as $\frac{\lambda_{SU}}{N_{dev}}$. Do not forget to save parameters in a separate function and read from there. Previously, we talked that for HQ SU requests base and enhancement layers can be served as first base then enhancement at the same

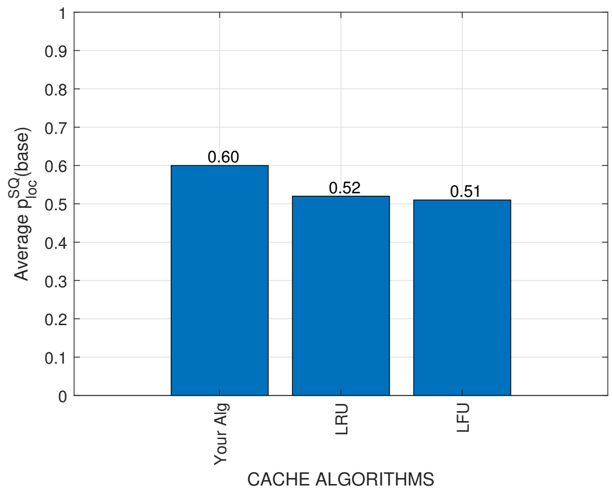

Fig. 2: Average Local Hit Rate of SQ Requests Results

frequency. Or they can be served simultaneously from different frequencies. Note than in the second scenario there will be Base Blocked and Enh Local Hit like cases not listed in case B event types above. Please use first base then enhancement at the same frequency approach for this project.