Mind the Gap: Modelling Video Delivery Under Expected Periods of Disconnection

Argyrios G. Tasiopoulos, Ioannis Psaras, George Pavlou
Dept. of Electrical & Electronic Engineering, University College London
WC1E 7JE, Torrington Place, London, UK
{argyrios.tasiopoulos, i.psaras, g.pavlou}@ucl.ac.uk

ABSTRACT

In this work we model video delivery under expected periods of disconnection, such as the ones experienced in public transportation systems. Our main goal is to quantify the gains of users' collaboration in terms of *Quality of Experience* (*QoE*) in the context of intermittently available and bandwidth-limited WiFi connectivity. Under the assumption that Wi-Fi connectivity is available within underground stations, but absent between them, at first, we define a mathematical model which describes the content distribution under these conditions and we present the users' *QoE* function in terms of undisrupted video playback. Next, we expand this model to include the case of collaboration between users for content sharing in a *peer-to-peer* (P2P) way. Lastly, we evaluate our model based on real data from the London Underground network, where we investigate the feasibility of content distribution, only to find that collaboration between users increases significantly their *QoE*.

Categories and Subject Descriptors

C.2 [**COMPUTER - COMMUNICATION NETWORKS**]; C.2.4 [**Distributed Systems**]
Subjects: Distributed applications

1. INTRODUCTION

Public transport is the preferred means of travel by many commuters in big cities like London and New York [11]. Quite often, while traveling, commuters spend their time on smartphone applications where, among others, live streaming is an increasingly popular one [4]. However, in the underground case, video delivery is extremely challenging since commuters face consecutive intervals of connection, when the train is in the station, and disconnection, when the train is in the tunnel,[1] which has a serious impact on their *QoE*.

In the standard case, users follow a *pull* approach, by requesting and downloading content from a server. Mobile users access the Internet through Wi-Fi access points or via cellular networks. Here, we are making the assumption that Wi-Fi access points are installed at each station so we limit our study to Wi-Fi connectivity, which is the case for the London Underground network, where cellular connectivity is not available. The download speed of each user depends on the total number of users who try to access the Internet at the same time, given that the access point bandwidth is shared equally among them. Assuming that bandwidth is limited (when in the station), and disconnection time is longer than the connection time (*i.e.*, the train spends more time travelling between stations than staying in one station), collaborative download seems an attractive approach to foster uninterrupted video playback. We call the collaborative video download approach *PUll and SHare* (PUSH).

Mobile *Peer-to-Peer* (*p2p*) networks [6] [8] offer an alternative and efficient solution for co-operative file sharing, especially in environments where connectivity is intermittent. In particular, *p2p* video streaming for mobile devices has attracted a lot of attention recently [17][14][9][16]. These studies focused more on the implementation and evaluation of these systems than on the design of a theoretical framework, which we argue that is necessary for understanding the dynamics of content delivery in such environments.

Content sharing in the context of public transportation systems has also been investigated in the past [13]. One part of this research took into consideration only Bluetooth connectivity capabilities of mobile devices while addressing the problem of transmitting data as a background process by studying commuters' mobility patterns and meeting frequencies [13][12]. Thus, by default they did not include the case of live video streaming. Another category examined the installation of additional routers on public transportation vehicles, such as buses and trains [10][15], but again these routers would have an independent store-and-forward functionality rather than continuous Internet connectivity. These store-and-forward devices would effectively act as users who are capable of storing bigger amounts of data, which they share on demand.

Although mobile multimedia in intermittently connected, ad hoc and opportunistic networks has received wide attention [17][9][16][5] both from an implementation and from a modelling perspective, little has been done in the context

[1] In some cases cellular connectivity might be available while underground and in-between stations, but reception is still poor. We leave the modelling and evaluation of the cellular case for future work.

CHANTS'14, September 7, 2014, Maui, Hawaii, USA.

http://dx.doi.org/10.1145/2645672.2645680.

of modelling collaborative video download in urban transportation systems.

That said, the goals of our work are: *i)* to model the *pull* case under bandwidth-limited intermittent connectivity, *ii)* to model the same environment for the *pull and share* (PUSH) case, which follows a *p2p* approach, *iii)* to express the users' utility for both approaches and to quantify the potential gains of the *PUSH* case in terms of *QoE.* In more detail, we build a model to estimate the video playback disruptions over consecutive periods of connectivity gaps. We map these disruptions to a utility function, assuming that their maximum *QoE* is achieved when the video is watched without gaps. The resulting utility function is taking into account energy consumption issues for *p2p* sharing, as well as changing network conditions in terms of users entering and leaving the system.

The rest of the paper is organised as follows. In Section 2, we describe both the *pull* and the *PUSH* models and build their utility functions. In Section 2 we also analyse the behaviour of the resulting utility functions over time. Then, in Section 3, we evaluate the performance of both models in realistic settings, using real commuter traces from the London Underground network. Finally, we conclude in Section 4, where we also discuss directions for future work.

2. MODEL FRAMEWORK

We begin by modelling the basic *pull* case, where a user is *pulling* video chunks on demand from a server. We then proceed to model the *PUSH* case in which, via collaboration, users interested in the same content form groups, download and then exchange content in a *p2p* manner.

Our objective is to develop a utility function for each of the two video delivery approaches that captures all the important parameters which affect the users' QoE. We envision that the outcome of our models are part of a smartphone application which takes care of the related procedures, without introducing any additional delay or overhead.

2.1 Pull Video Delivery Model

Since we want to model an environment of consecutive time intervals of connection and disconnection (*i.e.*, train in or inbetween stations), we define as *Epochs*, noted as Ep^i for an epoch i of duration $|Ep^i|$, a time interval which consists of a *Connection Period*, C^i of duration $|C^i|$, and a *Disconnection Period*, $\widetilde{C}^i$ of duration $|\widetilde{C}^i|$, respectively.

An important assumption of our model is that users are connected to the Internet via Wi-Fi access points. This means that the total number of users, N, who demand bandwidth, share the available bandwidth, B_{total}, equally, assuming a fair underlying transport protocol. Thus, each user is allocated $\frac{B_{total}}{N}$ bandwidth. This is in contrast to the case of cellular connectivity, where each user utilises the bandwidth of his/her own channel.

In practice, the number of users can change arbitrarily over time with an impact on bandwidth reallocation. Therefore, without loss of generality, we assume that the bandwidth is reallocated every 1 second, which is arbitrarily chosen as the convergence time that the applied transport protocol needs to adjust to the new conditions.

Next, assuming that video is streamed in chunks, we calculate the rate at which chunks are downloaded over a given period of time. That said, a chunk's size, S, is equal to:

$$S = b \times y \tag{1}$$

where b is the chunk's bit-rate and y is the chunk's duration in terms of playback seconds. Hence, the chunks' rate per second at time t, $X(t)$, for each user is:

$$X(t) = \frac{B_{total}}{S \times N(t)} \tag{2}$$

where $N(t)$ is the number of users who at time t request for some video content.

Consequently, given that $|C^i|$ (the connectivity duration) is an integer multiple of seconds, the total number of chunks that a user can download over epoch i, X^i, is:

$$X^i = \sum_{t=1}^{|C^i|} X(t) \tag{3}$$

and throughout epochs f to i:

$$X_{f\to i} = \sum_{j=f}^{i} X^j \tag{4}$$

where as f we consider the first connectivity epoch of a user.

For simplicity we assume that users share chunks only after having downloaded them completely, although in practice incomplete chunks can also be shared. Consequently, during epochs f to i an active user has downloaded:

$$L_{f\to i} = \lfloor X_{f\to i} \rfloor \times y \tag{5}$$

worth of watching time for a content of his/her choice.

Note that there is significant difference between the amount of content (in terms of playback time) that someone has downloaded and buffered, and the content that he/she has actually watched. At first, let us consider the simple case of the first epoch f. During f a user can watch content equal to the downloaded duration, $L_{f\to f}$, or the epoch duration, $|Ep^i|$, whichever is shorter. Obviously, that applies only when the total duration of a content, Y, surpasses the duration of the epoch. All the above observations are accumulated into the following equation:

$$W^f = \min[\min(|Ep^f|, L_{f\to f}), Y] \tag{6}$$

Given the watching time over the first epoch, W^f, we are now able to calculate the *playback disruption time*, D^f, that a user experiences during epoch f. Then, D^f is the difference between the epoch duration and the buffered content playback time, as it is illustrated by the formula:

$$D^f = \begin{cases} 0 & \text{if } W^f = Y \\ |Ep^f| - W^f & \text{otherwise.} \end{cases} \tag{7}$$

Since our objective is to estimate the aggregated watching and playback disruption time at the epoch level, we express both of them in a form of recursive association to the previous epochs.

The total playback time of a user until an epoch i, is:

$$W^i = \min[W^{i-1} + \min(|Ep^i|, L_{f\to i} - W^{i-1}), Y] \tag{8}$$

where, "$L_{f\to i} - W^{i-1}$" is essentially the difference between the downloaded content so far (from epoch f to epoch i) and the buffered but still unwatched content from the previous epoch $(i-1)$. Therefore, the total playback disruption until epoch i becomes:

$$D^i = \begin{cases} D^{i-1} & \text{if } W^i = Y \\ \sum_{j=f}^{i} |Ep^j| - W^i & \text{otherwise} \end{cases} \qquad (9)$$

Finally, we define the user's utility in epoch i, U^i, as:

$$U^i = W^i - a_d \times D^i \qquad (10)$$

where a_d is the *delay sensitivity coefficient* and indicates the delay tolerance of this user in terms of disruptions counted in seconds. Equation 10 captures the QoE a user obtains in this environment since a playback of 1 second of the content increases the utility by 1, while a disruption of 1 second decreases it by a_d. Effectively, if a user is willing to tolerate 0.5 seconds of disruption after 1 second of playback, he/she will maintain his/her utility at the same level (*i.e.*, the utility will neither increase or decrease) with $a_d = 2$. Note that the utility could become negative since by watching a content with disruptions a user could have an experience even worse than not watching it at all. In practice, a user after a negative utility point would quit watching the content. Lastly, the utility function in Eq. 10 does not take into account the video bit-rate, although intuitively higher bit-rate could be associated with a higher QoE; we leave this study for future work.

2.2 Pull and Share Video Delivery Model

In mobile *p2p* networks users interested in the same content form groups in which they are considered as peers. In these groups they act as both content consumers, since they download part of the content, and suppliers, since they share their downloaded content with the rest of the peers of the group. In our setting, since the bandwidth is limited and allocated equally, we are interested in examining the benefit of collaborative content download to improve QoE in the context of interrupted video delivery. With this goal in mind, we extend our previously presented model in order to study the performance potential of *p2p* sharing for real-time, mobile video delivery.

2.2.1 Model and Assumptions

Assume that users interested in the same content form a group g of $N_g(t)$ members at time t, by using a hypothetical smartphone application which supports the *PUSH* approach. The main idea is that each member of the group will share his/her downloaded content with the rest of the group. Wi-Fi direct (based on Wi-Fi Alliance[2]) allows for transmission up to 656 feet away. Thus, we assume that a user can share his/her content with the rest of the group peers within the same platform or train. This leads us to our first assumption:

Assumption 1: All members of a group can share downloaded chunks with all others by broadcasting it once[3].

The maximum throughput of sharing is achieved when users transmit non-overlapping chunks, which again are assigned to each user by our hypothetical application. This does not necessarily mean that users have to request a content concurrently, but rather that they have to be synchronised for downloading their common remaining content. This synchronisation, of joining a group g at epoch i, could take t^i seconds due to various kinds of overheads and possible playback disruptions. In the following epochs this group will have already been formed but the number of the group participants could be different since commuters leave the group at arbitrary stations, which in our model is interpreted as the beginning of a new epoch. Then, the total number of chunks that this user can download over this epoch by following the *PUSH* approach, $\widetilde{X}^i$, is:

$$\widetilde{X}^i = \sum_{t=1}^{t^i} X(t) + \sum_{t=t^i+1}^{|C^i|} X(t) \times N_g(t) \qquad (11)$$

where at time $t > t^i$, $X(t) \times (N_g(t) - 1)$ chunks are downloaded from the group members and the remaining $X(t)$ from the Wi-Fi access point, as in the *pull* case. Please note that this does not mean that the user has to receive $X(t) \times (N_g(t) - 1)$ chunks at time t but rather that this is the amount of chunks that (s)he will receive until the end of the epoch. This is always the case when the aggregated speed of sharing is bigger than that of the aggregate download bandwidth B_{total}. We also assume that:

Assumption 2: Users can receive content simultaneously from two WiFi interfaces, one connected to the WiFi access point and the other to the WiFi direct interface peer.

The replacement of X^i by $\widetilde{X}^i$ is the first main modification of the *PUSH* against the *pull* approach. The second one concerns the utility function of a user, as we show next.

2.2.2 Users Utility Function in Pull and Share Model

In both mobile and fixed *p2p* networks, when users have to act as content suppliers they have to spend their personal resources. This fact creates a strong incentive for users to join a group and receive contents without contributing anything, also known as *free rider problem*. This problem has attracted the interest of the academic community in the past and a number of solutions have been proposed [3][7]. However, an in depth analysis on incentive-based collaboration is beyond the scope of this paper.

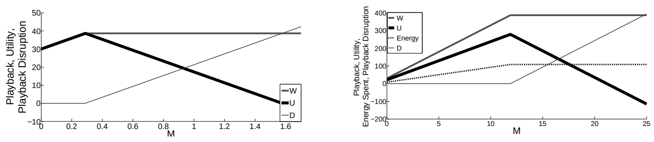
In our setting, the resource that a smartphone user has to spend is the transmission energy of the content that (s)he broadcasts. If the transmission/receiption speed in the *p2p* mode is much larger than the reception one from the WiFi access point, then the energy spent to broadcast is much smaller in the *PUSH* case, since content transfers last shorter periods of time. Our model implies *fairness* which under these conditions exists when all the members of a group contribute equally by spending the same amount of energy.

Energy consumption has to be depicted in the utility function of a user in the *PUSH* case. In order to estimate this loss in terms of QoE, we compare the energy that a user would spend to download some amount of content from the Wi-Fi access point with the energy that he/she spends in broadcasting the same amount of content to the rest of the group members. We capture this ratio in the *speed correlation coefficient*, $a_{speed}(t)$:

$$a_{speed}(t) = \frac{S/B_{trans}}{S/\bar{B}_{rcv}(t)} = \frac{\bar{B}_{rcv}(t)}{B_{trans}} \qquad (12)$$

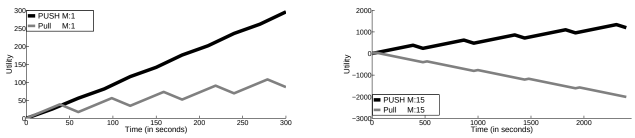
[2]Wi-Fi direct. `http://www.wi-fi.org`

[3]We base our assumption on pseudo-broadcast techniques, similarly to [9], where a connection is set up between two devices only, while the rest of the devices "eavesdrop" on the same channel.



(a) *Pull* Utility (b) *PUSH* Utility

Figure 1: *Pull* and *PUSH* Utility comparison



(a) M: 1 (b) M: 15

Figure 2: Utilities over epochs for *PUSH* and *Pull* approach

Setting Variable	Value
B_{total}: WiFi Bandwidth	54 Mbps
b: Video Bit Rate	419 Kbps
y: Chunk's Playback Duration	5 sec
Y: Video Size	4.5 min
$\|C\|$: Connection Duration	30 sec
a_d: Delay Sensitivity Coefficient	1
s: Zipf Exponent	0.8

Table 1: Evaluation setting

where B_{trans} is the bandwidth that a user can use in order to broadcast a chunk and $\bar{B}_{rcv}(t)$ is the average bandwidth that has been allocated to a user while he/she was active, under connectivity periods, until time t. Unfortunately, we have to omit its calculation due to space restriction which, however, is quite straightforward. Apparently, this coefficient represents how faster or slower a chunk can be transmitted within the group (in *p2p* mode) compared to the time needed to be downloaded from the Wi-Fi access point.

Next, by exploiting some information regarding Wi-Fi chipset power consumption, we can estimate the energy power correlation coefficient, a_{energy}, as:

$$a_{energy} = \frac{T_x}{R_x} \tag{13}$$

where T_x and R_x are the power consumption under transmission and reception over a certain time window, respectively. In fact, these values are estimated to 640 mW for T_x and 432 mW for R_x for the time an average web page needs to be transferred [1][2].

Thus, the utility loss of a user for sharing k chunks, $\widetilde{U}_L(k)$, is estimated in terms of watching time by:

$$\widetilde{U}_L(k) = a_{energy} \times y \times k \times \sum_{t=t_1}^{t_k} a_{speed}(t) \tag{14}$$

where $t_1, t_2, ..., t_k$ are the time instants where a user broadcasts. Therefore, the final utility in the *PUSH* case is:

$$\widetilde{U}^i = \widetilde{W}^i - a_d \times \widetilde{D}^i - \widetilde{U}_L(k) \tag{15}$$

where $\widetilde{W}^i$ and $\widetilde{D}^i$ are the watching time and playback disruption, respectively for the *PUSH* case.

2.3 Utility Function Behaviour

In this section, we capture the behaviour of the utility function for both the *pull* and the *PUSH* cases. We make the following assumptions: *i)* all epochs have the same overall duration, *ii)* the disconnection to connection duration ratio, M, is constant for all epochs: $|\widetilde{C}| = M \times |C|$, and *iii)* the number of users remain stable over all epochs.

We assume that $N = 100$ commuters participate, the values of $|C|$, B_{total}, b, y, and a_d are set according to Table 1 and we capture the utility functions as the disconnection to connection duration ratio M increases.

In Fig. 1a we depict the behaviour of the utility function for the *pull* case. Initially and for as long as the disruption time is zero, the utility increases linearly together with the actual watching time. This is because, as mentioned earlier, the utility function increases by one for every playback second. In the *pull* case, this increase of the utility function reaches up to 39 seconds (*i.e.*, $M = 0.3$). Uninterrupted playback here is due to the fact that the connection duration at the beginning of the first epoch is sufficient for the user to download enough content for another 9 seconds while disconnected. The watching time, on the other hand, remains stable after the tipping point of $M = 0.3$ in Fig. 1a, as the fixed connection period $|C|$ allows for downloading fixed amount of data. That said, and given that playback disruptions start rising the overall utility declines.

On the other hand, in Fig. 1b, we capture the utility function of the *PUSH* case for a user who belongs to a group of 10 peers, while the rest of the commuters are downloading content individually. Hence, bandwidth allocation is the same as in the *pull* case. We observe that the utility function does not increase together with the watching time, due to the extra energy spent by the group members to broadcast their individually downloaded chunks. In absolute numbers however, the utility function is much bigger than the one in the *pull* case. Furthermore, we see that due to collaborative downloading, the system is more tolerant to disruptions which start only after the disconnection duration is 12 times the connection duration, *i.e.*, $M = 12$.

Please note that in order to capture the worst case scenario, we have assumed that the speed correlation co-efficient $a_{speed}(t)$ (Eq. 12), namely the parameter that depicts the ratio of the amount of data that a user would have downloaded from the WiFi point to the data he/she would have broadcast within the time needed to broadcast one chunk of data to its peers, is equal to 2. In other words, we assume that downloading from the WiFi is twice as fast as downloading from a nearby device. Given that the WiFi route would involve network congestion and round-trip delay to some server, this setting is rather unrealistic and in reality the utility in case of the *PUSH* approach is higher.

Finally, in Fig. 2, we plot the utility functions over consecutive epochs when the disconnection to connection ratio, M, is 1 (Fig. 2a) and 15 (Fig. 2b). We see that in this theoretic setting the utility of the *PUSH* case can tolerate disconnection times of up to 15 times the connection intervals.

3. EVALUATION

In this section we evaluate both approaches based on real commuter traces from the London Underground network.

3.1 Content Assignment - Group Formation

We begin by developing a content assignment algorithm in order to guide group formation within commuter trains

Algorithm 1 Content Assignment Algorithm for Epoch i

```
    Input: N^i, p_new, s
    Output: Content and group for each user in epoch i
 1: procedure MAIN:
 2:     contentPopularity:={}
 3:     contentPopularity[empty content]:=0
 4:     contentPerUser:={}
 5:     for user in N^i do
 6:         p_user := random_number(0, 1)
 7:         if p_user ≤ p_new then
 8:             user_content:=new_content
 9:             contentPopularity[user_content]:=0
10:         else
11:             user_content:=zipf'sLawCDF(contentPopularity,s)
12:         contentPopularity[user_content]+:=1
13:         contentPopularity:=sort(contentPopularity)
14:         contentPerUser[user]:=user_content
15:     return contentPerUser
```

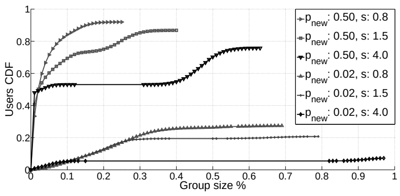


Figure 3: Algorithm assignments

in the *PUSH* case. The algorithm's objective is to create a dynamic popularity for the requested contents, given that requests follow Zipf distribution (lines 11) in Algorithm 1.

We build on a hypothetical application which provides commuters with information about the current groups (and the related contents being downloaded), along with an estimation of the disruption that they will likely experience. Based on that, commuters can choose to join an existing group or create a new group, which will initially consist of one user only.

Our algorithm captures the users' choice by parameterising the probability of demanding a new content, denoted as p_{new} (line 7). Note that we also consider the *empty content* as a content that someone could join (line 3) and in that way we cover the user's choice of staying idle during one epoch.

We note that the content demand is estimated at the epoch level. We refer to new users who joined the system, have finished watching a content, or during the previous epoch chose the empty content, as available users N^i. We assume that these users express their request for each epoch only at the beginning of a connection period. Algorithm 1 returns the content assigned to each user and the groups are formed among them according to content similarity.

In Fig. 3 we capture how 100 users are allocated to groups of relevant magnitude, for various p_{new} and s parameters, after 50,000 iterations. We notice that p_{new} assigns a unique content to evenly distributed groups of users. For example, for $p_{new} = 0.5$, half of the users' population belongs to a group of size of 1% of the total population. As regards the Zipf exponent s, the higher its value the less the chances of a user joining a content which is not ranked in the head of the distribution (*i.e.*, among the first). Thus, in the case of a low p_{new} value, the empty content is usually ranked first. Nevertheless, in the rare case that upon the first few content assignments a non-empty content will be created the group that will be formed will be huge as we can see for s equal to 4. Although Zipf exponent $s = 4$ is an unrealistically high value, we use it here for the purpose of capturing the behaviour of the proposed group formation algorithm.

3.2 Tube Setting

For our realistic setting we use the publicly available traces from Transport for London (TfL),[4] which include the incoming and outgoing rates of commuters per quarter of an hour for each station. We isolate the commuters of one line, the Victoria Line, which is the line with the fewest interchanges. We then map commuters to trains (given the train frequencies) according to their entry and exit stations. Thus in the end, for each train route we have a group of commuters, each of them travelling for a fixed number of stations.

We then apply the setting of Table 1 and we obtain the results of Fig. 4. From Fig. 4b and 4a we can compare between the *pull* and *PUSH* approaches as the probability of a user requesting a new content (p_{new}) increases. The ideal utility here is defined as the QoE that a user would have in the case of unlimited bandwidth and continuous connectivity for each requested content. As we increase p_{new} the ideal utility rises as well, since demand for bandwidth resources up to $p_{new} = 6\%$ does not exceed supply. After that point (*i.e.*, $p_{new} \approx 6\%$) demand exceeds supply and therefore the ideal utility stabilises.

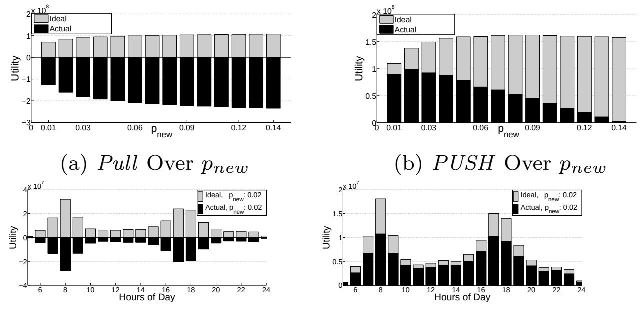
The actual utility that users obtain for the *PUSH* case (Fig. 4b) reaches a peak for $p_{new} = 2\%$ after which it steadily declines for the same reason (*i.e.*, as demand exceeds supply, the actual utility that users get is limited by bandwidth and connectivity constraints).

In the *pull* case, on the other hand, the actual utility that users obtain is always negative, given the bandwidth and disconnection constraints. This highlights the importance of collaborative download in challenged environments. At this point we should stress that in absolute values the ideal utility for the *PUSH* case is significantly higher. This owes to the fact that users have the chance to (collaboratively) download and watch more videos than in the *pull* case within the same time window.

In Figs. 4d and 4c, we plot the ideal and the actual utility that users achieve over time (for both approaches) for one operational day of the London Underground Victoria line, that is from 6am to midnight. We see a clear increasing trend in the demand for underground services (and therefore content delivery as well) during rush hours (namely, between 07:00-09:00am and between 16:00-19:00). As expected, more demand for content delivery in a bandwidth constrained environment results in reduced $\frac{actual}{ideal}$ utility. That is, for instance, in the PUSH case, while in the off-peak hours the $\frac{actual}{ideal}$ utility ratio reaches up to 90%, *e.g.*, at 11:00am and 22:00, during peak hours this ratio might fall as low as 60%, *e.g.*, at 08:00am.

In relative terms, the situation is similar for the *pull* approach (see Fig. 4c). The non-collaborative download, however, constantly results in negative utility. Although this might be relatively bearable in off-peak hours, where the utility is only slightly negative, the difference widens as demand for content increases during the peak hours.

[4] `http://www.tfl.gov.uk/info-for/open-data-users/`



(a) *Pull* Over p_{new} (b) *PUSH* Over p_{new}

(c) *Pull* Over Time (d) *PUSH* Over Time

Figure 4: The punctual utility of two approaches over the probability of new content, p_{new}, and time

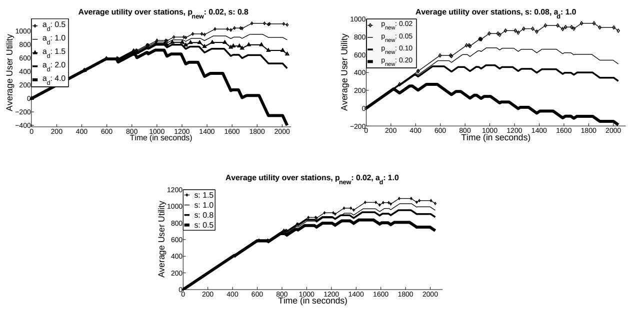


Figure 5: Utilities over epochs for *PUSH* approach

Finally, in Fig. 5, we present the utility function of the *PUSH* approach over time, for the end-to-end journey of the Victoria Line of the London Underground network. As expected, we observe in the top left plot of Fig. 5 that the higher the sensitivity of users to playback disruption, the lower the utility they get. Given the bandwidth constraints in the WiFi setting investigated here, we also see in the top right plot of Fig. 5 that as demand for new contents increases, the utility again declines. That is, as the system is requested to carry more contents through, the limited bandwidth offered causes more playback disruptions. Finally, in the bottom plot of Fig. 5, we observe, as expected, that the higher the exponent of the Zipf distribution s, the higher the utility that users get. As the demand for already requested content increases, the group formation algorithm creates larger groups therefore utilising the limited bandwidth resources more efficiently.

In all three plots of Fig. 5 we witness similar trends of the respective utility functions. That is, initially the utility increases, before it reaches a peak (around the middle of the plot), after which it starts decreasing. This trend is different to the one we have seen in Fig. 1b for the theoretical evaluation of the utility function and is explained by the fact that time here (*i.e.*, the $x-$axis) reflects the end-to-end journey of trains. That is, trains start from end-stations empty, where there are enough bandwidth resources for the commuters; trains then reach more central areas, where more commuters join in and form larger groups; finally the trains reach the other end of the line, where commuters leave the system. During this last period, as commuters leave, and although more resources are made gradually available, in absolute terms, the groups are left with less resources (*i.e.*, a group of less users will be allocated less resources). This explains the declining trend of the utility function towards the right end of the plots in Fig. 5.

4. CONCLUSIONS

We have designed a model to capture users' *QoE* when streaming video from WiFi access points in cases of intermittent but expected periods of disconnection, such as the ones experienced in public transportation systems. We have found that under the simple *pull case*, where users individually download real-time video content, the experience might be unpleasant, due to continuous gaps in the playback. We have then extended our model to capture the case of collaborative video download between users interested in the same content and found that the quality experienced can improve significantly. We plan to extend our study for the cellular case, where each user utilises his/her own channel and also build incentives for content sharing.

Acknowledgment

This work was supported by EU FP7/NICT (GreenICN project) grant no. (EU) 608518/(NICT)167.

5. REFERENCES

[1] Atheros AR6001, <http://www.atheros.com>.

[2] Average Web Page Size, <http://www.websiteoptimization.com/>.

[3] S. Buchegger and J.-Y. Le Boudec. Performance analysis of the CONFIDANT protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks). In *Mobihoc '02).*

[4] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018, *White Paper, [Online] http://goo.gl/l77HAJ*, 2014.

[5] A. Detti, B. Ricci, N. Blefari-Melazzi: Peer-To-Peer Live Adaptive Video Streaming for Information Centric Cellular Networks, *24th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRCÕ13)*

[6] G. Ding, B. Bhargava. Peer-to-Peer File-Sharing over Mobile Ad hoc Networks. In *Proc. PerCom Workshops*, 2004.

[7] J.J. Jaramillo and R. Srikant. DARWIN: Distributed and Adaptive Reputation mechanism for WIreless ad-hoc Networks. In *Proc. Mobicom '07.*

[8] M. Joa-Ng and I.-T. Lu. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. In *IEEE JSAC*, 1999.

[9] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou. Microcast: Cooperative video streaming on smartphones. In *MobiSys 2012.*

[10] J. LeBrun and C. Chuah. Bluetooth Content Distribution Stations on Public Transit. In *MobiShare'06*, 2006.

[11] G. Lyons and K. Chatterjee. A Human Perspective on the Daily Commute: Costs, Benefits and Trade-offs. *Transport Reviews*, 28:181-198, March 2008.

[12] L. McNamara, C.Mascolo and L.Carpa. Content Source Selection in Bluetooth Networks. In *Mobiquitous 2007.*

[13] L. McNamara, C.Mascolo and L.Carpa. Media Sharing based on Colocation Prediction in Urban Transport. In *Mobicom 2008.*

[14] M. Ramadan, L. El Zein, and Z. Dawy. Implementation and evaluation of cooperative video streaming for mobile devices. In *PIMRC 2008.*

[15] P. Rodriguez, R. Chakravorty, J. Chesterfield, I. Pratt, and S. Benerjee. MAR: A Commuter Router Infrastructure for the Mobile Internet. In *MobileSys 2004.*

[16] H. Seferoglu, L. Keller, B. Cici, A. Le, and A. Markopoulou. Cooperative video streaming on smartphones. In *Allerton Conf. on Communication, Control, and Computing, 2011.*

[17] M. Stiemerling and S. Kiesel. A system for peer-to-peer video streaming in resource constrained mobile environments. On *ACM U-NET Workshop 2009.*