Distributed Content Storage for Just-in-Time Streaming

Sourav Kumar Dandapat
Department of CSE,
IIT Kharagpur, India
sdandapat@cse.iitkgp.ernet.in

Sanyam Jain
Department of CSE,
IIT Kharagpur, India
jain.sanyam@gmail.com

Niloy Ganguly
Department of CSE,
IIT Kharagpur, India
niloy@cse.iitkgp.ernet.in

Romit Roy Choudhury
Department of ECE and CS,
Duke University, USA
romit@ee.duke.edu

ABSTRACT

We propose a content distribution strategy over municipal WiFi networks where Access Points (APs) collaboratively cache popular multimedia content, and disseminate them in a manner that each mobile device has the portion of the content *just-in-time* for playback. If successful, we envision that a child will be able to seamlessly watch a movie in a car, as her tablet downloads different parts of the movie over different WiFi APs at different times.

Categories and Subject Descriptors

C.2.4 [**Distributed System**]: Distributed Applications

General Terms

Design, Performance

Keywords

Content Distribution, Municipal WiFi Network, Distributed Content Storage

1. INTRODUCTION

Video on Demand (VoD) has gained immense popularity in recent years whereby almost 50% of the entire Internet traffic is due to video. Video distribution becomes a challenging task especially when we consider distribution over wireless networks (which carries 18% of overall traffic). The challenges arise from low wireless data rates, high packet loss probability, link disconnections due to mobility, etc. Hence attention needs to be paid towards designing strategies to distribute content over wireless network [1] to fulfill the dual objectives of upholding user-experience and congestion control.

The increasingly large number of WiFi APs deployed across cities opens up new opportunities for seamless distribution of such content. A feasible strategy can be to cache popular content in such APs, whereby a traveler moving (say, by car) past those APs can download content from them. Specifically, a traveler during her journey might be interested in a movie, so she starts the process of simultaneous download and playback. However, it is almost infeasible to download a complete movie (assumed to be large in size) from a single AP due to the short association duration of the mobile device. So the download needs to be performed through successive APs and hence (different parts of) the same movie needs to be available in other APs in order to support just-in-time download. Caching the entire (movie) file in *every* AP reduces the total number of movies available to users, and is undesirable. We propose a content distribution strategy that ensures minimum redundancy storage while also offering uninterrupted playback to all users, regardless of their travel path.

SIGCOMM'12, August 13–17, 2012, Helsinki, Finland.
ACM 978-1-4503-1419-0/12/08.

2. CONTENT DISTRIBUTION

Let us assume that the total number of APs in a municipal WiFi network is n and every AP is capable of storing m files (each of same size). So the described network can cache up to $n \times m$ number of unique files.

To allow seamless download on the move, one trivial solution is to keep same set of m files across all n APs. However, it would limit the total number of cached file to m from $n \times m$. Almost similar efficiency can be achieved if we break each file into set of chunks of identical size and place 1^{st} chunk at the 1^{st} AP that client comes across, 2^{nd} chunk at the 2^{nd} AP and so on. This solution, while requiring less storage, would not be useful for another client (say) C, which travels the same path in the reverse direction. In this case, C would still download all the chunks but in reverse order, hence has to wait until the end (referred as *idle* time) before it can start playback.

Chunk Frequency: An intuition which can be derived from the above mentioned situation is that to eliminate *idle* time, the 1^{st} chunk of the file must be available at every AP given that a user can start her journey from any point. If we assume that a car normally moves at 40 km/hour and the range of an AP is 50 m (radius), then the car would stay associated with an AP for 9 seconds. If a file can be downloaded at 1 Mbps rate then at most 9 Mb can be downloaded from an AP, which would then require approximately 61 seconds to play (assuming 80 minutes is needed to watch a 700 MB movie). By this time of 61 seconds, a user may encounter other APs, depending on the density of WiFi deployment. Assuming, a WiFi AP every $100m$, the vehicle may encounter 6 APs. Thus, the 2^{nd} and 3^{rd} chunks of the file must be available at least in one of the next 6 and 12 APs respectively, starting from the source where the user

starts moving. To generalize, the q^{th} chunk must be available within $(q-1) \times p$ APs, while $q \geq 2$ or frequency distribution of q^{th} chunk f_q is $\geq 1/((q-1) \times p)$, where p is the number of APs a user encounters while playing a video chunk.

Chunk Distribution Strategy: We assume that each traveler follows the shortest path from the source to her destination. To determine which chunks must be cached at each AP, we need to consider all shortest paths between pairs of locations. We assume X_1, X_2, ...X_n represent n APs of the network. We also assume that a binary variable z_i^j is associated with every X_i AP, where $z_i^j = 0$ signifies the absence of the j^{th} chunk at node X_i and $z_i^j = 1$ signifies presence. To ensure that the q^{th} chunk is present at least once in a path (X_i, ..., $X_{i+(q-1)\times p}$) of length $(q-1) \times p$, the following condition must be satisfied: $z_i^q + z_{i+1}^q + ... + z_{i+(q-1)\times p}^q \geq 1$. We assume that file has been divided into κ chunks and each AP can host at most k ($\leq \kappa$) chunks.
With these constraints in mind we minimize the objective function f_{obj} that represents total storage required for the file caching. We use a Integer Linear Programming (ILP) solver (LPSOLVE) – the ILP package outputs the assignments of chunks to AP.

Random Distribution Strategy (RDS): Since the ILP package also provides the frequency of different chunks in the entire network for the proposed solution, a random distribution strategy can be to distribute identical number of chunks randomly among APs. We use the RDS as a baseline scheme for comparison.

3. EXPERIMENT

We have considered a region of the IIT Delhi campus (rectangular area with dimension $1km \times 1km$) for our case study [2]. 83 APs are assumed to be placed in the entire region. The movie file distributed is of size 45 Mb and divided into 5 equal sized chunks.
We use network simulator (NS3) to simulate and evaluate the performance of our proposed solution. "The ONE" [3] simulator is used to generate shortest path mobility model trace of clients. We place chunks in an AP cache as obtained from the ILP solution stated above. A client sends request to the associated AP for the desired file. In response, AP sends available chunks of requested file. The client selects a particular chunk for downloading depending on chunk selection strategy. We have done experiments for different *storage constraints* (percentage of AP memory allocated to storing a given file).

4. RESULTS AND DISCUSSION

Metrics: The performance of the proposed algorithm is evaluated using the following metrics:

(a) **Interruption Fraction(%)($\mathcal{IF}$):** Lets assume that a client takes t_1 unit time in her total journey and was idle (could not watch movie) for t_2 unit time. $\mathcal{IF}$ of this client can be formally expressed as $\frac{t_2 \times 100}{t_1}$.

(b) **Download Efficiency($\mathcal{DE}$):** It signifies average number of chunks downloaded from each AP. If a device crosses p APs and collects q chunks then $\mathcal{DE}$ of the device is $\frac{q}{p}$.

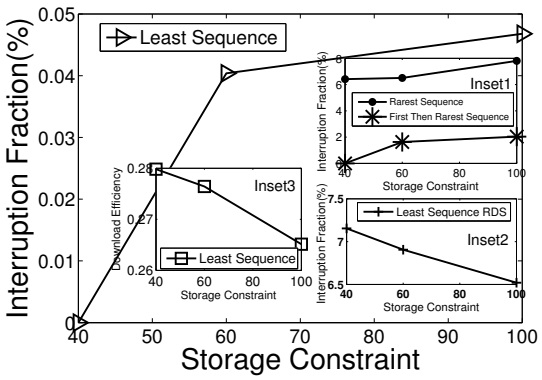

Figure 1: $\mathcal{IF}$ and $\mathcal{DE}$ of proposed and random schemes w.r.t chunk selection strategies.

Discussion: Fig. 1 shows variation of $\mathcal{IF}$ with different storage constraint of the proposed algorithm while client always downloads the chunk with Least Sequence First (LSF). Result shows that the $\mathcal{IF}$ is almost negligible in this strategy. Inset1 of Fig. 1 shows the variation of $\mathcal{IF}$ while the chunk selection strategies are first then rarest (priority of first chunk is highest then it selects rarest chunk) and rarest respectively. Rarest selection strategy has significantly higher $\mathcal{IF}$ as client needs to wait long for the initial chunks. Inset2 of Fig. 1 shows the variation of $\mathcal{IF}$ of RDS while chunk selection strategy is LSF. $\mathcal{IF}$ of RDS is significantly high compared to the proposed algorithm. An interesting observation is that in our algorithm the performance degrades (though very slowly) with loosening of storage constraint while it has opposite effect in case of RDS. This is because with loosening of constraint, rare pieces have a tendency of clustering, which sometimes makes them unavailable, whereas the average pieces are relatively easier to find, thus improving performance of RDS. The same tendency is noticed when we capture the $\mathcal{DE}$ (Inset3).

5. CONCLUSION & FUTURE DIRECTION

The key contribution of this poster is in showing that chunks of a (video) file can be scattered in distributed caches (in WiFi APs) in a manner that a mobile user may be able to download the needed chunk just in time for playback. This may allow the user to watch streaming content without interruption, thereby offloading cellular network traffic. Our simulation results present early promise, motivating a full scale implementation and evaluation on a real end-to-end system.

6. ACKNOWLEDGEMENT

Authors would like to thank Vodafone for their partial financial support.

7. REFERENCES

[1] J. Eriksson, H. Balakrishnan, S. Madden, *Cabernet: Vehicular Content Delivery Using WiFi*, Mobicom 2008
[2] http://www.openstreetmap.org/
[3] The Opportunistic Network Environment simulator www.netlab.tkk.fi/tutkimus/dtn/theone/