Offloading Cellular Traffic

Sourav Kumar Dandapat, Swadhin Pradhan, Niloy Ganguly
Department of CSE, IIT Kharagpur, India
{sdandapat,swadhin.pradhan,niloy}@cse.iitkgp.ernet.in

ABSTRACT

Cellular network is becoming heavily congested. This paper exploits few opportunities to reduce the traffic of cellular network. Initial results are promising.

1. INTRODUCTION

Internet traffic is increasing in exponential rate every year and will reach zettabyte threshold by year 2015 [1]. And a significant portion of this traffic is due to wireless. As a result cellular networks are becoming heavily congested.

Given that wireless capacity is nearing Shannon's limit, researchers in academia and industry are looking for the next best solution. Of course, no single modification will cure the entire problem, rather, the evolving system will need to exploit every opportunity that comes along. This paper focuses on two opportunities to reduce cellular load -(1) exploiting the proliferation of Wi-Fi access points through offloading; (2) Exploring collaboration among wireless devices. First opportunity is applicable to an urban area where Wi-Fi is enabled whereas second opportunity suits almost everywhere.

An important trend that has been noticed that traffic due to video is significantly high and it is increasing. Moreover, an interesting statistics reveals that top 10% popular videos of YouTube accounts for 80% of views([4]). So, if we can take special care about these popular video files, we would be able to reduce traffic significantly.

Rest of the paper is organized as follows. In section 2, we explore the option of offloading cellular traffic through Wi-Fi network. In section 3, we discuss about collaboration among devices during download. In section 4, we discuss about evaluation of the above mentioned opportunities. Section 5 concludes with some future directions.

ICARE 2013 New Delhi, India

2. OFFLOADING THROUGH WI-FI NETWORK

2.1 Overview

Wi-Fi networks are being deployed across cities, university and office campuses, airports, malls etc. Augmenting Wi-Fi network to reduce cellular traffic is no more new [5]. However, intermittant connectivity specially on move reduce the effectiveness of Wi-Fi network. Obviously, there are two options to increase the effectiveness of Wi-Fi network - 1) increasing the density of Wi-Fi access points so that connectivity improves, 2) increasing effective downloading rate so that when device is connected it may download with much higher rate. First option is always not feasible and moreover it increases infrastructure cost hugely. Recent technology enables us to easily attach memory with Wi-Fi access points. This paper tries to leverage that opportunity to increase download rate. With memory attached, Wi-Fi networks can be treated as distributed servers for content distribution. In this content distribution network contents are appropriately sprikled across memories of access points, so this system is named as *Sprinkler*.

2.2 System Settings

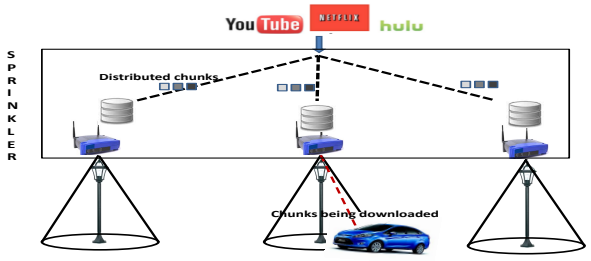

Figure 1: Video content is periodically pushed to Wi-Fi. Vehicles passing an AP downloads content from it thus offloading cellular traffic.

Fig. 1 shows the system setting of *Sprinkler*. *Sprinkler* has two functionalities - 1) pulling videos periodically from online services such as Netflix, Hulu etc. based on popularity, content category etc. and scattering different "chunks" (a video consists of several chunks) of the videos to different Wi-Fi APs, 2) interacting and serving video "chunks" to mobile devices. When a device is in contact with a Wi-Fi

AP, it attempts to download chunks that it will need in the future. These chunks need not be in order, i.e., if the device already has, say, chunks 1 to 5, and the Wi-Fi AP has chunks 1, 3, 7 and 12, the device attempts to download both 7 and 12. Of course, it is possible that the client disconnects from the AP before chunk 7 is downloaded in its entirety. In that case, the client caches the packets of chunk 7, but scans for another AP that has chunk 7 – on encountering one, it completes downloading the remaining packets. On the other hand, if downloading of chunk 7 is complete, the client proceeds to download the next available chunk (12 in this example). In this manner, *Sprinkler* hopes that the device will download chunk i before it needs it for playback. If, however, the needed chunk is not available, it switches to cellular network (it is assumed that mobile devices in vehicles have access to both Wi-Fi and 3G/LTE, and can choose to multiplex between the two.) and continues streaming the video.

2.3 Chunk Distribution Intuition

An efficient strategy, for chunk distribution, can be developed from the following intuition. Availability of initial chunks should be higher [6]. A user starting to play a video file will need the initial portion of the file – the first chunk – right away. Since the user can start watching at any location, the first chunk needs to be available at every AP. However, the 2^{nd} chunk needs to be downloaded by the time the 1^{st} chunk has played out. More generally, let us assume that a user moves past k APs during the viewing time of a single chunk. So, user will start viewing chunk X after she crosses $(X-1)\times k$ APs, then we have to ensure the availability of chunk X within $(X-1)\times k$ consecutive APs. Thus, higher numbered chunks can be made available at proportionally less frequency – the ratio of availability for chunks 1:2:3, can be modeled as $1{:}\frac{1}{k}{:}\frac{1}{2k}$.

2.4 Formulating as a Linear Program

Building on the above intuition, we formulate the general problem as a linear program. We require as input a topology of APs, and a city specific *ideal speed* at which on an average car moves in that city. The details follow.

Let us assume X_1, X_2,...,X_p represent p consecutive APs in a shortest path of a city; and n, the number of APs in the network and k is the number of APs, a client can cross within viewing time of a chunk. Let an indicator variable z_i^j be associated to AP X_i, where $z_i^j = 0$ signifies the absence of the j^{th} chunk at AP X_i and $z_i^j = 1$ signifies its presence. Availability of first chunk at every AP can be trivially expressed as below:

$$z_i^1 = 1 \tag{1}$$

To ensure that the j^{th} chunk is present at least once in a path (X_i, ..., $X_{i+(j-1)k}$) of length $(j-1)\times k$, the following condition must be satisfied:

$$z_i^j + z_{i+1}^j + ... + z_{i+(j-1)\times k}^j \geq 1 \tag{2}$$

where i varies from 1 to $p - k\times(j-1)$ and j varies from 2 to m (number of total chunks in a video file). We will get a set of constraint equations considering all shortest paths and all chunks. For ease of storage management, every AP should have a maximum (β) as well as a minimum limit (θ) on the number of chunks that can be kept in one AP. Mathematically, these constraints can be expressed as follows.

$$\theta \leq \sum_{j=1}^{m}(z_i^j) \leq \beta, \forall i = 1,\ldots,n \tag{3}$$

Our objective function is to minimize total storage (f_{obj}). f_{obj} is formally expressed as below:

$$f_{obj} = \sum_{i=1}^{n}\sum_{j=1}^{m}(z_i^j) \tag{4}$$

This optimization problem can be solved using Integer Linear Programming (ILP), where the number of variables equal $n\times m$. We use LPSOLVE [2] for solving the ILP – the package outputs the assignments of chunks for each AP.

2.5 Experimental Setup

We perform simulation based experiments using the NS3 simulator. We use constant data rate (9Mbps - 24Mbps) model for our experiments. We have used Jake's propagation loss model to realize an urban environment. We use the road network of a part of Mysore as a case study. We place APs at every Zm (Z: 100 / 150 / 200 / 250 / 300 for different simulation). Clients are moving according to map based shortest path mobility model. In this experiment chunk size is assumed to be 3 MB playback rate for the video is 700MB/hour.

3. OFFLOADING THROUGH COLLABORATION

3.1 Overview

Two important observations motivate us to exploit collaboration opportunity among mobile devices in proximity. These are 1) people with similar interest meet and interact frequently and 2) popularity of files in Internet follows a power law. These observations establish high chance of downloading similar files by devices in proximity. In recent times with the increase in traffic, as a device download independently, it incurs more congestion, more bottleneck of server and hence penalize the user in terms of both quality of service as well as the cost incurred to access that service. This scenario opens the opportunity of collaboration among devices in proximity to download a common file. Instead of downloading complete file, every device in collaboration downloads a part of the file and exchange downloaded parts among themselves to get the complete file.

3.2 Group Formation

Few important aspects need to be addressed for forming a collaborative group - 1) who should be the members of a group? 2) What is the ideal size of a group? 3) How does a group form? Devices in proximity and with similar interests should be member of a group and there must be an upper limit on the size of a group; otherwise overhead time to exchange sub-part of a file among group members will fade the gain from collaboration. A device interested for downloading a set of files, first check its proximity to find existing collaborative group with similar interest. If it finds such group, it joins the group otherwise it initiates a new collaborative group.

3.3 Fair Load Distribution

An important challenge of such collaborative group is to distribute workload among devices. Existing collaboration strategies allow some setup time to let the members join a group and then statically distributes load among the members of that group. There are few issues with this strategy 1) within setup time enough number of devices may not join the group 2) devices waste setup time without downloading anything 3) it disallows churn which is almost unavoidable in mobile environment. Here we propose a dynamic load distribution strategy. Initiator starts downloading the file as soon as it establishes a collaborative group. Whenever a new device joins an existing group, it checks other devices in group that are reachable in k hops (k is a design parameter). And the responsibility of the device (among checked devices), which is currently having highest load, is taken over by the new device. With higher value of k, responsibility gets more evenly shared however at the expense of higher message exchange. New node also finds a backup node which will be responsible to take care of the responsibility of new node if it leaves the collaboration in between.

3.4 File Exchange

Every device broadcasts its own part of the file to other devices in group.

3.5 Experimental Setup

In this experiment, it is assumed that devices join the collaborative group according to poisson distribution. Experiment is executed for 50 seconds while the size of the file being downloaded is 1500 MB. It is assumed that speed of WWAN is 1MB/sec while speed of WLAN is assumed 30MB/Sec. An area of $100m \times 100m$ is used for this experiment. It is also assumed that the two devices are in range if they are within $10m$ from each other. Initiator is placed in a random location and any other device can join the group when it is within $10m$ from any device in the group.

4. EVALUATION

In this section, we evaluate two schemes discussed above against benchmark schemes.

4.1 Schemes for Comparision

***Far-Sprinkler*:** It is a system where the APs don't locally host video, but pull them from distant servers. *Far-Sprinkler(x,y)* will denote x% of servers are nearby and y% of servers are far away.

***Existing Collaboration Strategy*:** Our result is compared with existing strategy where initiator of the group waits for group setup time (here it is assumed 10secs) to allow other devices to join the group before responsibility is statically distributed among those devices.

4.2 Metrics of Interest

First metric is used to evaluate *Sprinkler*, while last two metrics are used to evaluate effectiveness of collaboration.

1. **Fraction of data offload (FDO)** measuring the percentage of the video packets during the vehicle's journey downloaded over Wi-Fi.

2. **Effective Throughput:** Effective throughput of a device is expressed as $TotalFileSize/TotalTime$.

3. **Jain's Fairness Index (JFI):** It measures the fairness of the system. JFI of a system can be formally defined as

$$JFI = \frac{(\sum_{i=1}^{n} f_i)^2}{n \times \sum_{i=1}^{n} {f_i}^2} \quad (5)$$

where f_i is the responsibility of i^{th} device of the group.

4.3 Performance of Sprinkler

Effect of AP Density: In this experiment, for different inter-AP distances (100m/150m/200m/250m/300m), we have measured the *Sprinkler's* performance in terms of Fraction of Data Offload (FDO).

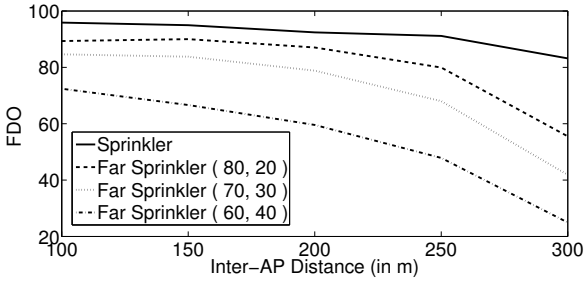

Figure 2: Fraction of data offload with different inter-AP distance. The data rate is assumed as 9Mbps.

Fig. 2 compares the FDO of *Sprinkler* with different *Far-Sprinkler* schemes with respect to different inter-AP distances. *Sprinkler's* performance in terms of FDO is over 90% even when the inter-AP distance is 250m. For *Sprinkler* the performance degrades slowly and gracefully while the degradation is quite sharp for *Far-Sprinkler*.

Effect of Vehicle's Speed: We measure the percentage of average data offload from 3G using *Sprinkler*, while the vehicle moves at different *average speeds*.

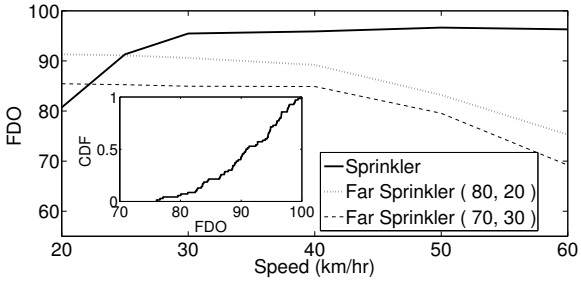

Figure 3: FDO against speed while cars move at different speeds and APs are placed at every 100m. Inset shows CDF plot of FDO.

From Fig. 3 it is evident that *Sprinkler's* data offload performance is over (90%) across a wide range of speed. When a client moves at lower than ideal speed, it remains associated with an AP for longer duration (> required time to download a chunk). In such a scenario, the client, may have to switch to other network, if it fails to download multiple consecutive chunks from AP as it takes longer time to reach next AP. However, presence of consecutive chunks is not guaranted by chunk distribution strategy. So, at lower than ideal speed, fraction of data offload becomes comparatively low. On the contrary, at lower speed, *Far-Sprinkler* client performs better as it remains associated with an AP for longer duration and there is guarantee of receiving chunks in sequence. FDO

reaches its maximum value ≈ 96% when we consider *ideal speed*. Inset shows the CDF of FDO while cars move at random speed in between 30Km/hour - 50Km/hour.

4.4 Performance of Collaboration

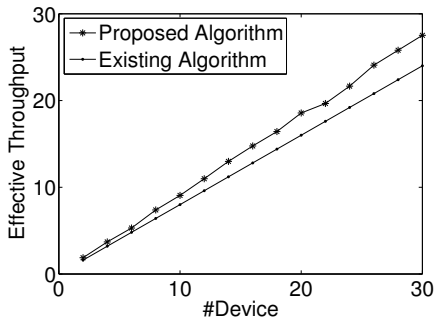

Figure 4: Throughput variation with group size

Fig. 4 shows the variation of throughput as group size increases. Result shows that, through collaboration, every device in group effectively experiences throughput which is almost equal to WLAN speed. Our algorithm outperforms the existing algorithm [3] by not wasting the group setup time.

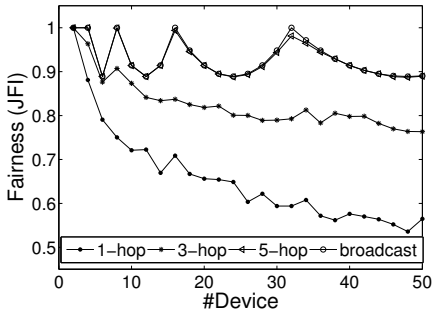

Figure 5: Fairness varies with group size for different scheme

Fig. 5 shows the variation in fairness as group size changes for different schemes. Results show that unrestricted broadcast achieves the maximum fairness while it is moderate when node with which responsibility is being shared is chosen among the 3-hop neighbors.

5. CONCLUSION

Trend of increasing vedeo demand will continue and with cellular network capacity drying up, such a service may be difficult to support over 3G/LTE connections. In such scenario, all kinds of options needs to be exploited whereever it is possible. Intial results of above mentioned two schemes are very motivationg in terms of cellular traffic offloading. This paper is an early step in this direction with much more research remaining to make it an end to end reality.

6. REFERENCES

[1] Cisco white paper.

[2] http://lpsolve.sourceforge.net/5.5/.

[3] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath. Combine: leveraging the power of wireless peers through collaborative downloading. In *MobiSys '07*, pages 286–298, New York, NY, USA, 2007. ACM.

[4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *IMC*, pages 1–14, California, USA, October 2007. ACM.

[5] S. Dimatteo, P. Hui, B. Han, and V. O. Li. Cellular Traffic Offloading through WiFi Networks. In *MASS*, pages 192–201, Valencia, Spain, October 2011. IEEE.

[6] K. A. Hua and S. Sheu. Skyscraper broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems. In *SIGCOMM*, pages 89–100, Cannes, France, October 1997. ACM.