HyCloud: A system for device-to-device content distribution controlled by the cloud

Jiri Danihelka
ETH Zürich
Zurich, Switzerland

Domenico Giustiniano
IMDEA Networks Institute
Madrid, Spain

Theus Hossmann
ETH Zürich
Zurich, Switzerland

***Abstract*—The exponential growing demand of mobile Internet data traffic requires a rethinking of current infrastructures and calls for innovative approaches that provide online-streaming of content to interested subscribed mobile users. We introduce a mobile system called HyCloud for device-to-device content distribution. HyCloud delivers the content from a cloud backend, and it can reduce the total amount of data transferred through the cellular network by exploiting the similarity of interests among multiple mobile users. Our preliminary real-life experiment with video streaming with a simple content scheduler shows that the system can save in average 24% (and up to 38%) of data downloaded by the cellular connection.**

I. Introduction

We are witnesses of a new paradigm in the IT industry, cloud computing. Its success is due to several factors, such as the efficient scalability with the number of clients, ease of use and high flexibility [1]. A significant portion of the cloud computing market is linked to mobile services, wherein the mobile cloud computing (MCC) paradigm realizes the vision to leverage the cloud to reduce the burden on applications running on resource-constraint mobile devices. The MCC approach is limited by the load that can be offered by the cellular network, of which capacity may not suffice the needs of society in the future [2].

In this contribution, we take a different approach, and study *how MCC could tackle the overloading of the cellular networks* and *how it can substantially help the network operators to alleviate the load of the network*. We propose an architecture, *HyCloud*, based on the principle that the cloud uses the cellular base station only for injecting a few copies of the content. It then controls that device-to-device (D2D) local communication opportunities spread it to the interested community. In this way, the content provider let the cloud service inject content to some users through the cellular network and control the opportunistic D2D delivery from these users via a local wireless interface (e.g., WiFi or Bluetooth) to other interested users upon meeting them. By exploiting the locality of users interested in the same content, such an approach has the potential to substantially reduce the load of the cellular infrastructure.

HyCloud has some key difference with respect to a traditional MCC approach:

- HyCloud is based on the interplay of the cloud and D2D communication and it aims to resolve the overloading of cellular networks.
- The traditional MCC mainly relieves the computational burden of the mobile device, but it does not attempt to reduce the user's cellular communication cost to access the service.
- With HyCloud, the cellular network is mostly used to control the delivery of the content through lightweight signaling, and only partially to deliver the content to the mobile users.
- Differently from the traditional MCC, HyCloud exploits the nomadicity of users to deliver the content using one between the cellular and D2D interface.
- With HyCloud, the content provider economically benefits from an approach where part of the content uses the D2D communication interface rather than the communication resources of the cloud. This has the advantage of reducing the billing of the content provider.

Our goal is to describe the research challenges to make this approach practical and demonstrate the functionality of the system. To validate our results, we have created a proof-of-concept application running on stock smartphones, which are capable of playing media-streaming content from online sources with granted delays. The results we show are preliminary, yet encouraging. Finally, we discuss key open issues.

II. Related Work

Previous work looked at the problem of what strategy should be applied to inject mobile application content to a subset of the interested users through the cellular network and let these users opportunistically spread the content to the subscribed community [3]–[6]. An alternative approach to download not-time-critical data is to wait until the cellular network will not be fully utilized or until another type of connection (e.g. WiFi Access Points) will be available. Lee et al. studied the concept of delayed offloading where content is downloaded from one or more WiFi hotspots until the deadline is reached [7]. At the deadline, the content is downloaded from the cellular network. Their experiments showed that significant gains (beyond 29%) of data traffic offloaded to WiFi can be achieved with one or longer hour of delay. However, the gain is almost negligible (only $2-3\%$) with a more practical delay of 100 sec. [8] proposed a system for downloading based on priorities that can postpone the less important data when the network is fully utilized. This system requires significant changes in both mobile devices hardware and cellular network

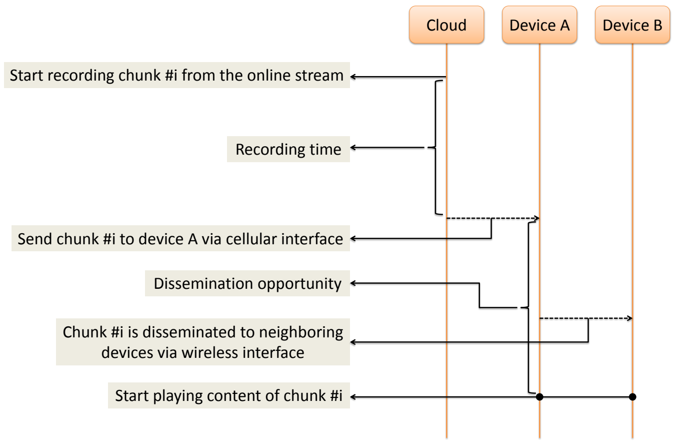

Fig. 1. Flowchart representing how online media stream is transmitted to subscribed users. The content is converted into chunks (blocks of fixed time length) and chunk must reach the user during the dissemination opportunity interval and before the deadline.

infrastructure. Patel et al. proposed a method (see [9] and therein) for managing and queuing traffic in cellular networks that can postpone less important data downloads in case of an overloaded network. None of the works above considered how MCC itself could help to relieve the cellular traffic.

III. HyCloud

We aim to create a cloud system that controls the distribution of content to mobile devices using both downloads from the cellular backhaul connection and opportunistic D2D connection. The diagram of the process is shown on Fig. 1. The cloud manages records of requested content, and decides which devices should receive it through the cellular network. Online media from 3rd party is recorded in our cloud service and converted into chunks (blocks of fixed time length). Once the chunk is recorded, it is send to some of the devices subscribed to the channel (Device A). During the dissemination opportunity interval, devices without the chunk (Device B) try to connect to devices with the chunk and download it opportunistically using either WiFi or Bluetooth. When the deadline for receiving the latest data content becomes close, each device, which has not yet received the data chunk through either cellular network or opportunistic communication, pulls the chunk via cellular infrastructure, thus allowing for guaranteed delay of the content [10].

A. Implementation

We divide the application into two main parts, cloud part and mobile phone part. The cloud part is deployed in Microsoft cloud Windows Azure. We use Windows Phone 8 devices for the mobile part. Porting the HyCloud mobile application to Android and iOS is possible in the future. All cloud components can be accessed from these mobile platforms.

1) Cloud Platform: In HyCloud, the content scheduler runs in Windows Azure Cloud Services while authentication, data storage and notification service are provided by Windows Azure Mobile Services. Applications for Microsoft Azure are written in C# using .NET libraries.

The main principles are illustrated on Fig. 2. Each device registers to subscription channels of its interest (such as popular news feed or video streams) and it expects to receive the latest content stored in the online cloud. When the scheduler decides to send a content to a device, it notifies the device to download it from Blob Storage using the signaling mechanism provided with Microsoft Push Notification Service (MPNS). A typical push notification usually requires an intervention of the user to download the content but this process is automated by our system. In the experiments, a cloud component processes available video content and audio streams to a format required by our mobile application. The scheduler running in the worker role can be configured by the content provider, and it should be designed to compute the optimal number and time of cellular deliveries to selected users based on the history of content deliveries. In this work, we use a simple strategy, where the content is initially transmitted to a fixed set of randomly selected subscribed users. For purposes of our experiments, we set the number of initial transmits to one fourth of total number of subscribers. Developing the best strategy for transmitting the content from the cloud is out of the scope of this work.

A new device can register to desired channels using the Azure Mobile Service. The worker role is then notified about the new device and its content requests. It will then consider the new device in its future data delivery schedules.

2) Mobile Platform: We build our solution on Microsoft technologies using HTC 8S devices with Windows Phone 8. D2D communication can be performed either by Bluetooth or WiFi-Direct. In this contribution, we use Bluetooth 3.1 communication which has smaller effect on battery depletion than WiFi-Direct.

Windows Phone has been designed from the beginning with cloud connectivity in mind. Its development tools have good support for cooperating with Simple Object Access Protocol (SOAP) web services, and some cloud services were even designed for seamless integration with Windows Phone devices (e.g. Azure Mobile Services). Interconnection of mobile and cloud platforms allows us to create demanding cloud solutions more robustly, swiftly and easily compared to other smartphone platforms (iOS, Android, Blackberry).

For purposes of our experimental testing, our application is always the foreground one and users cannot switch applications during the test. The phone screen can be locked or turned off, yet this does not interfere with the application. This is due to the shortcoming of Windows Phone 8 OS in terms of running multiple applications. We are aware that is approach would be rather impractical for real-life users but we hope that the future versions of Windows Phone OS will allow better multitasking capabilities, similarly to what already possible with Android OS.

IV. Evaluation

In order to evaluate our approach we make an experiment with 16 devices distributed to people working in the same

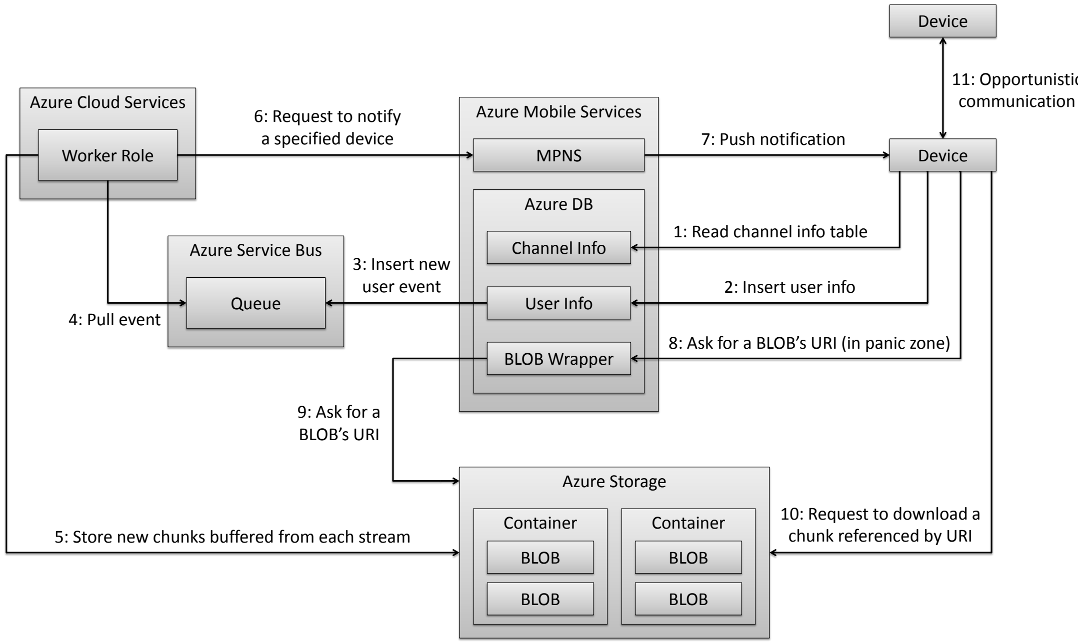

Fig. 2. *Architecture of HyCloud system*. Azure Mobile Services component manage subscription (1, 2) of devices to content channels. Content requests are stored (3) to Azure Service Bus queue to be processed. Cloud Worker Roles do recording of requested channels (4) and store the created chunks to Azure Storage (5). The device can notified (6, 7) by the Azure Mobile Service that it should download the chunk at the beginning (7, 10) of the dissemination opportunity interval. Otherwise the device will download the chunk at the end (8, 9, 10) of the dissemination opportunity interval or using opportunistic communication (11).

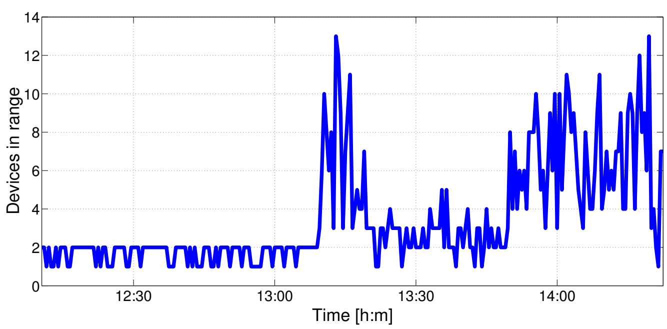

Fig. 3. Number of devices detected by Bluetooth scanning for a single smartphone

university building and we monitor data downloads of the devices during 4 hours. The devices play a video using the HyCloud approach with a bit rate close to 400 kb/s (equivalent to low-resolution YouTube video). Each participant carries one device. The aim of the experiment is to simulate the situation when multiple users want to watch the same online video-streaming content (e.g. a popular live concert performance). We are aware that in some live-streaming scenarios additional latency might not be acceptable for the users – e.g. a football match. However, similar practices have been successfully applied in the past to, e.g., P2P video streaming. In the tests, we consider that one quarter of chunks is randomly distributed to the devices in the beginning of the respective dissemination opportunity time interval (thus it could not be saved by Bluetooth offloading, and the maximum possible saving is 75%). We further fix the scanning period of Bluetooth to 30 seconds. The device cannot receive or transfer data to another device while it is scanning.

To make the chunks small enough we limit the chunk size to 500 kB (approximately 10 second of recording). We set the dissemination opportunity interval to one minute to provide enough time for D2D content dissemination. As a result of this configuration, our system provides video delayed approximately by 70 seconds (10 sec to create the first chunk on the server and 60 sec to have guarantees that all the subscribed users have received it). Video delayed time is constant since HyCloud operates with guaranteed delay. A summary of the setting used in the experiment is reported in Table I.

Because the dissemination opportunity time interval (60 sec) is longer than the recording interval (10 sec), dissemination opportunity time intervals of several chunks can partially overlap. In case of a Bluetooth connection of two devices

TABLE I
SETTING USED IN THE EXPERIMENT

Number of devices	16
Video play rate	≈ 400 kb/s
Duration of the experiment	4 hours
Dissemination opportunity time interval	60 sec
Chunk size	500 kB
Recording interval per chunk	10 sec

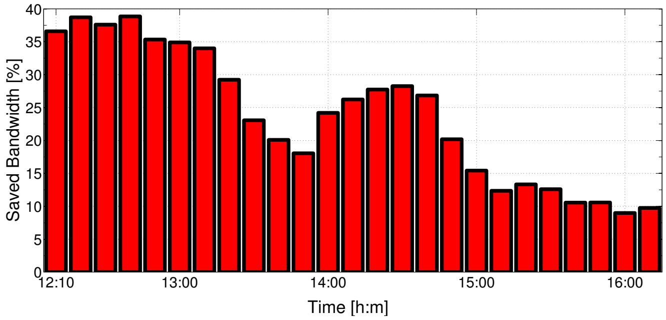

Fig. 4. Amount of saved bandwidth during our evaluation experiment. The drop near 13:50 was caused by lunch break when many participants left the building. The saved bandwidth is decreasing in time since some of the participants have left the building or their smartphone runs out of battery.

with multiple chunks designed for dissemination, it has to be decided which chunk is going to be transferred first. In our experiment we choose the chunk at random although we are aware that more sophisticated approaches could improve the results.

Fig. 3 shows the trace of nearby device observed by one smartphone. On average a phone is in Bluetooth distance of 3.53 other phones during the experiment. The percentage of saved bandwidth over time is then shown in Fig 4. Our measurements show that the system is able to correctly operate and our preliminary implementation allows to save 24% of downloaded content on average, and up to 38% according to the mobility pattern of mobile users. Not shown in the figure, by enlargement of the dissemination period by 30 seconds, the system can save another 7% of bandwidth. However, this is at the expense of additional delayed start of the video at the client that may not be acceptable for the users.

V. Open Issues and Discussion

a) Power consumption: In our initial implementation, the mobile phones have their short-range wireless interface always on to utilize the opportunistic communication, otherwise they may miss the opportunity to share data. Wireless on/off switching process could be controlled by the cloud in future, e.g., based on GPS (turn Bluetooth off when there are no expected devices nearby). Since power consumption is a principal issue in our design, the devices should search for neighboring devices periodically. Length of this period should be chosen as a trade-off between battery usage and cellular network offloading opportunities.

b) Algorithms: In some scenarios, it may happen that large number of mobile devices will request the content at the same time (deadline). In such cases, the cellular network may be overloaded by a large number of simultaneous requests. Algorithms must be designed to spread the download workload to wider time intervals.

c) Practical usage of current mobile applications: A technical challenge to practically implement our system into existing mobile applications is the support from the mobile operating system. The mobile OS should provide API using which 3rd party apps can register for content channels and state their maximum acceptable latency. The control communication with the scheduler in a cloud server would be aggregated to save phone battery and transmitted data. Similar aggregation concept is now used in mobile operating systems for push notification.

d) Incentives and attacks: The definition of some incentive mechanisms for the users that download data from the cellular network and then share it through D2D communication should be investigated. Users are billed for the data downloaded from the cellular network, and should not observe their battery largely depleted for the additional D2D transfer. In cases when multiple users use pay-as-you-go resources we have to further consider fairness of cost distribution among the users. Attackers may misuse the system downloading content through Bluetooth and saving some battery, e.g., by not sharing his chunks with others. Research effort must be put such that the cloud can detect malicious users that misuse paid resources of other participants.

e) Why the cloud rather than a traditional data center?: A main feature of content delivery is that the demand for a service varies over time. There are mainly two reasons for that. The first is popularity of the content, which changes over time (for instance on a daily scale). The second is the bandwidth of the D2D network, which drastically depends on the mobility patterns of users, whose variations are on much smaller scale than the one of content popularity. In fact, opportunistic D2D networking exploits the daily mobility of users, which enables intermittent and highly dynamic contacts whenever two mobile devices are in each others proximity. While provisioning a data center for the peak load it must sustain a few days per month leads to underutilization at other times, the cloud has a flexible and scalable management of the resources.

VI. Conclusion

HyCloud is a novel promising technology that can be used to save cellular data downloads. We described our deployment strategies, presented architectural principles, described the cloud and mobile components required in such applications, and discussed several open questions. To validate our research, we have created an application for audio- and video-streaming and measured amount of data saved using the application in a preliminary real-life experiment.

Acknowledgments

We thank Abouzar Noori for his initial contribution to the project and the anonymous reviewers for their inspiring and insightful feedbacks. The project has been sponsored by the Microsoft Innovation Cluster for Embedded Software (ICES).

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721672

[2] "Cisco visual networking index: Global mobile data traffic forecast update, 2011 – 2016," White Paper, Cisco, 2012, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.

[3] S. Ioannidis, A. Chaintreau, and L. Massoulie, "Optimal and scalable distribution of content updates over a mobile social network," in *INFOCOM 2009, IEEE*, April 2009, pp. 1422–1430.

[4] J. Whitbeck, M. Amorim, Y. Lopez, J. Leguay, and V. Conan, "Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, June 2011, pp. 1–10.

[5] B. Han, M. V. Marathe, P. Hui, D. Telekom, G. Pei, V. S. A. Kumar, and A. Srinivasan, "A.: Cellular traffic offloading through opportunistic communications : A case study. system," in *DOI 10.1145/1859934.1859943. URL http://dl.acm.org/citation.cfm?id=1859943*, 2010, pp. 31–38.

[6] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "Microcast: Cooperative video streaming on smartphones," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '12. New York, NY, USA: ACM, 2012, pp. 57–70. [Online]. Available: http://doi.acm.org/10.1145/2307636.2307643

[7] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can wifi deliver?" *Networking, IEEE/ACM Transactions on*, vol. 21, no. 2, pp. 536–550, April 2013.

[8] L. H. Linneweh Jr and M. A. Houghton, "Method and apparatus for allocating communication resources to support priority communications in a communication system," Jan. 19 1999, US Patent 5,862,485.

[9] P. R. Patel, A. R. Patel, O. T. Choksi, and M. R. Ali, "Method and system for queuing traffic in a wireless communications network," Mar. 8 2005, US Patent 6,865,185.

[10] A. Noori and D. Giustiniano, "Hycloud: A hybrid approach toward offloading cellular content through opportunistic communication," in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '13. New York, NY, USA: ACM, 2013, pp. 551–552. [Online]. Available: http://doi.acm.org/10.1145/2462456.2465728