Addressing Response Time of Cloud-based Mobile Applications

Sujit Dey, Yao Liu, Shaoxuan Wang, Yao Lu
Mobile Systems Design Lab
ECE Department, University of California, San Diego
{dey, yal019, shaoxuan, luyao}@ece.ucsd.edu

ABSTRACT

With more mobile applications being developed to take advantage of the elastic cloud computing resources instead of restricting to native mobile device resources, this paper investigates a timely question: is there any fundamental challenge that needs to be overcome to enable cloud-based mobile applications? We show that using cloud resources makes applications highly interactive and real-time, making low response time a key requirement for satisfactory user experience. Using two promising Cloud Mobile applications, Cloud Mobile Gaming (CMG) and Cloud Mobile Desktop (CMD), we demonstrate that meeting response time requirements can be a significant challenge. We show why existing Internet PC based solutions for delay and response time management may not be adequate to address the response time challenge of Cloud Mobile applications. We describe response time management techniques that have been recently developed for Cloud Mobile applications, and propose new directions, including developing Mobile Network Clouds, and Mobile Cloud Scheduling, as ways to address the response time challenge of Cloud Mobile applications.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *Cloud Computing.*

Keywords

Cloud computing, response time, mobile applications

1. INTRODUCTION

With the rapid evolution and adoption of smart phones and tablets, there is a growing desire to use them as both consumer multimedia and enterprise devices, including the capability to have rich Internet video and 3D gaming experiences, as well as use them for remote desktop computing and emerging applications like mobile medicine and mobile education. However, despite the progress in the capabilities of mobile devices, there will be a widening gap with the growing computing/power requirements of emerging Internet multimedia applications, like multi-player Internet gaming and augmented reality driven telemedicine. Mobile cloud computing can help bridge this gap, providing mobile applications the capabilities of cloud servers and storage together with the benefits of mobile devices and mobile connectivity, possibly enabling a new generation of truly ubiquitous multimedia applications on mobile devices.

Figure 1 shows the overall architecture and end-to-end flow of control and data between mobile devices and Internet cloud servers for a typical Cloud Mobile application. Though a CM application may utilize the native resources of the mobile device, like GPS and sensors, it primarily relies on cloud computing Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) resources, like elastic computing resources and storage resources, located in Internet clouds. A typical CM application has a small footprint client on the mobile device, which provides the appropriate user interfaces (touchscreen, voice, gesture, text based) to enable the user to interact with the application. The resulting control commands are transmitted uplink through cellular Radio Access Networks (RAN) or WiFi Access Points to appropriate gateways located in operator Core Networks (CN), and finally to the Internet Cloud. Subsequently, the multimedia data produced by the Cloud, either as a result of processing using the Cloud computing resources, and/or retrieval from Cloud storage resources, is transmitted downlink through the CN and RAN back to the mobile device. The CM client then decodes and displays the results on the mobile device display. From the above description, and as shown in Figure 1, a typical CM application will be highly interactive, with some CM applications needing near real-time response times.

In this paper, we investigate the viability of a few promising cloud mobile applications, including a consumer multimedia application, Cloud Mobile Gaming (CMG), and a desired enterprise application, Cloud Mobile Desktop (CMD). CMG enables a mobile user to play rich Internet games using any mobile device; the game engine is executed on a cloud server instead of on a mobile device. CMD enables a mobile user to

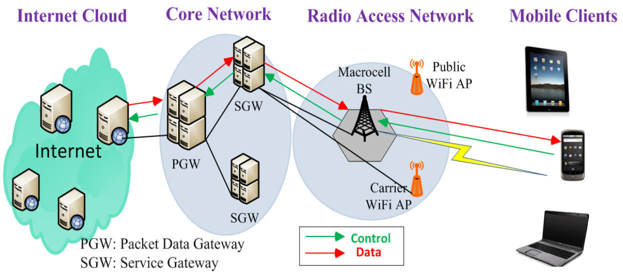


Figure 1: Cloud Mobile applications: architecture, control/data flow.

MobileCloud'13, July 29, 2013, Bangalore, India.
Copyright 2013 © ACM 978-1-4503-2206-5/13/07...$15.00.

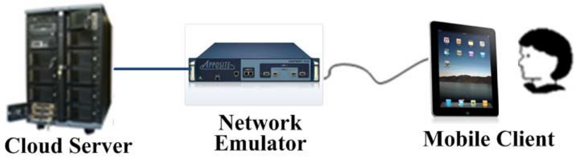


Figure 2: Experiment Testbed.

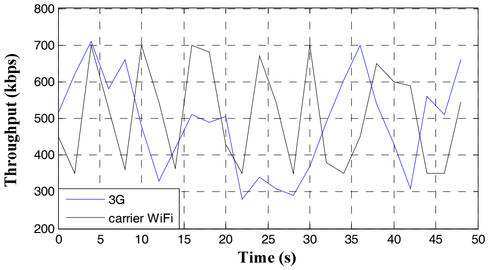


Figure 3: Throughput of Cellular (3G) and carrier WiFi networks.

remotely control a desktop executing on a cloud server by displaying the cloud desktop on the mobile device. For both CMG and CMD, rendered video is generated and encoded on the cloud server, and streamed downlink to mobile device through a mobile network. User commands are transmitted uplink from mobile device to cloud server to enable interaction. We conduct experiments with CMG [1] and CMD prototypes we have developed to investigate the quality of experience of CM applications using cellular and WiFi networks,. We demonstrate that response time can be a serious challenge that needs to be overcome, in particular for highly interactive applications like CMG and CMD.

We review recent techniques that have been developed to address delay and response time for interactive applications, including existing video conferencing and remote desktop techniques which seem to work well for wireline networks and PC clients. We demonstrate the deficiencies of such techniques for cloud based mobile applications like CMG and CMD. Subsequently, we discuss several response time management techniques, including cloud media adaptation techniques. We also propose extending the cloud beyond the traditional Internet to the edge of wireless networks. We describe the advantages of the resulting Mobile Network Clouds, and discuss the problems that need to be addressed. Finally, we introduce Mobile Cloud Scheduling, which simultaneously considers mobile network resources as well as cloud computing resources when making resource allocation decisions, so as to maximize the number of scheduled mobile application sessions whose response time requirements can be satisfied.

Table 1. Response Time Requirement for CMG and CMD

	CMG	CMD	
		Slide Show	Typing
Acceptable	440ms	835ms	390ms
Excellent	280ms	445ms	125ms

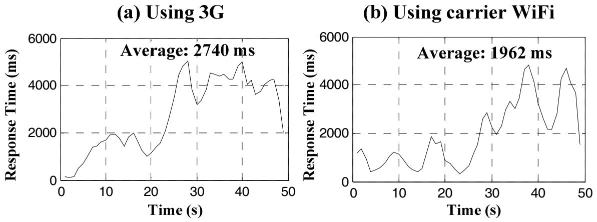


Figure 4: Response time of CMG

2. RESPONSE TIME CHALLENGES

Since streaming large amount of data over bandwidth constrained mobile network may lead to unexpected network congestion, it may be a challenge to achieve and maintain low response time for cloud based mobile applications. In this section, we first describe subjective experiments we have conducted using the CMG and CMD prototypes to determine response time values that will provide acceptable and excellent user experience. Next we conduct experiments with CMG and CMD prototypes in 3G and carrier WiFi network. The experimental results show that for both the CM applications, the response times under cellular and carrier WiFi networks are far from acceptable.

Firstly, in order to figure out the response time requirement for CMG and CMD, we conduct a set of subjective experiments. For CMG, we choose an open-source game PlaneShift [2], and for CMD we choose two most common user activities: Slide Show and Typing. Figure 2 shows our experiment testbed. A network emulator is inserted between the cloud server and mobile client to control the network response time. As the users are playing and issuing commands on the mobile client/device, we vary the response time using the network emulator and ask for users' opinion about whether their experience is acceptable or excellent. By taking the average of subjects' opinion, we list the response time requirements in Table 1.

We then characterize the response time obtained using a cellular 3G network and a WiFi hotspot of a mobile network operator. Figure 3 shows throughput profiles collected at a congested time, 5pm, which are used by the network emulator in our subsequent evaluations. Figure 4 shows the response time of playing a CMG game under the network conditions shown in Figure 3. We can see that the response time in Figure 4 keeps increasing when the network throughput is low in Figure 3. The average value of response time also shows that there can be a

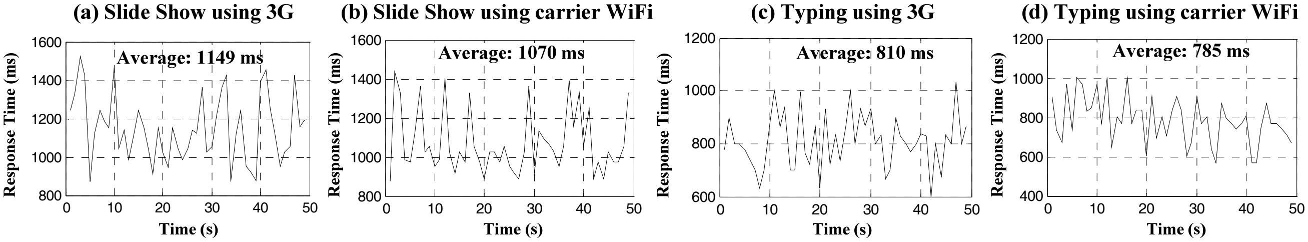


Figure 5: Response time of CMD.

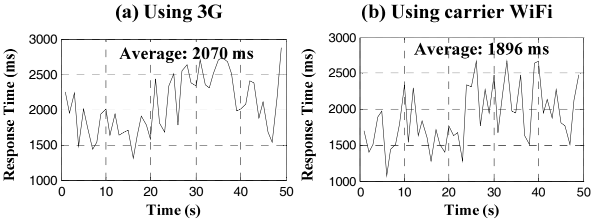


Figure 6: Response time using video conferencing application Skype to stream CMG video.

significant gap between what can be achieved by CMG during peak hours and the minimum requirement to ensure acceptable user experience. Similarly, Figure 5 shows the response time of two typical CMD activities (slide show and typing) under the emulated 3G and carrier WiFi network conditions (Figure 3). We observe from these graphs that while the response time using carrier WiFi is better than when using 3G, the average response time with carrier WiFi is still significantly more than acceptable values shown in Table 1.

The experiments for CMG and CMD above together with the subjective results shown in Table 1 reveal that meeting response time requirements for highly interactive CM applications can be a significant challenge.

3. CAN WE USE RELATED TECHNIQUES

We next review recent techniques that have been developed to address delay and response time for interactive applications, including existing video conferencing and remote desktop techniques, and investigate whether they can be used to address the response time challenge for Cloud Mobile applications. In the case of video conferencing, video adaptation techniques [3][4] mainly use frame rate adaptation to meet network constraints as typical video conferencing scenarios involve low motion video. However, such techniques may not be effective for Cloud Mobile Gaming, which typically has high frame rate requirement, and any substantial compromise on frame rate may result in perception of increased response time. Similarly, the existing remote desktop and desktop sharing applications mainly use frame rate adaptation to meet network constraints, and hence cannot be effective for CMG applications. Moreover, as our experimental evaluations reported later will show, while the remote desktop techniques may be adequate for wireline networks, they are not able to cope with the dynamic fluctuations seen in mobile networks, and hence cannot address the response time challenge of CMD.

To evaluate the effectiveness of the existing techniques for CM applications, we conducted the following two experiments: 1) streamed CMG video using the commercial video conferencing software Skype [5]; and 2) used the commercial remote desktop software Citrix [6], instead of our CMD prototype, to perform the two selected CMD activities, slide show and typing. We have captured the following two videos (accessible using the embedded or referenced links) to show the performance of Skype and Citrix for CMG and CMD applications respectively, using the cellular network profile shown in Figure 3. From the CMG video using Skype [7], we observe that the adaptation technique in Skype results in significant stalling and high response time, together with low frame rate, all leading to unacceptable CMG user experience. From the CMD video using Citrix [8], we see that every time the user issues an instruction to the desktop (such as navigating to the next slide show page), he has to wait for a long time for response, which leads to an unacceptable user experience.

The response time measured with 3G and carrier WiFi conditions (using the profiles shown in Figure 3) are shown in Figures 6 and 7. We observe that using the commercial video conferencing and remote desktop solutions, which will adapt encoding setting to network conditions, the response times obtained are better than the original response times shown in Figures 4 and 5; however, they are still higher than the acceptable response times shown in Table 1.

4. TECHNIQUES TO ADDRESS RESPONSE TIME

In this section, we give a brief overview of what constitutes response time in cloud based applications. Next, we introduce several techniques that can be used to reduce the delays and achieve acceptable response time.

4.1 Overview of Response Time

Figure 8 presents a round-trip flow of response time in cloud based applications, from the issuance of a command on the mobile device to the receiving of the resulting application data (for example images or videos) on the mobile device. At time T1, mobile client sends out a command. Cloud server receives this command at time T2, and then generates the application data at time T3. The generated application data is encoded and packetized, and then is sent to the client at time T4. The application data will be received by mobile client at T5 after a network downlink delay, and displayed onto the mobile device at time T6 after a client buffering and decoding delay. Based on the above analysis, the response time in the cloud based application mainly includes four sub-components: *network uplink delay* D_{UL} (T1-T2); *server delay* D_S (T2-T4); *network downlink delay* D_{DL} (T4-T5); and *client delay* D_C (T5-T6). Thus, response Time (R_T) can be formulated as

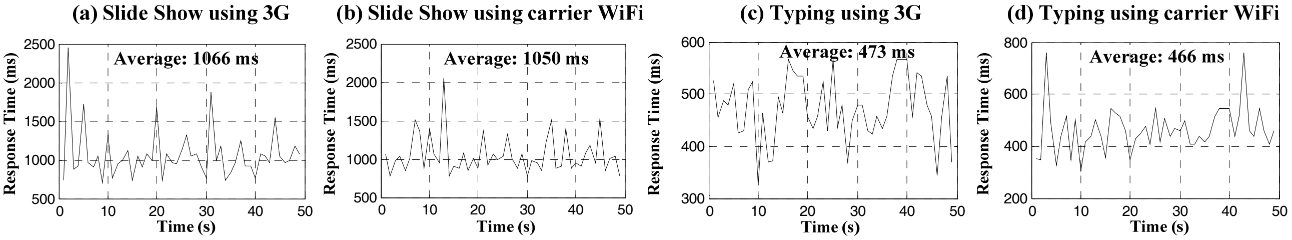


Figure 7: Response time when using remote desktop application Citrix.

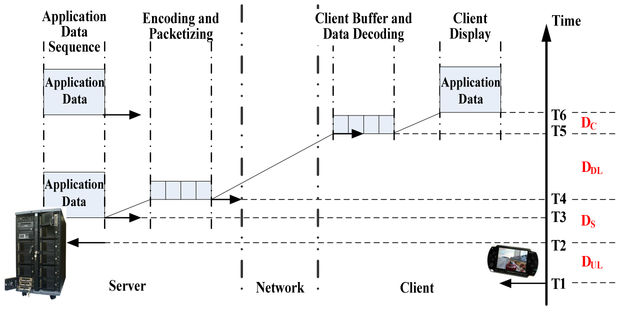


Figure 8: Round-trip flow of response time.

$$R_T = D_{UL} + D_S + D_{DL} + D_C \qquad (1)$$

The server delay D_S depends on the nature of cloud computing that needs to be done (for example, graphics rendering and video encoding for the CMG application) and cloud server utilization. The client delay D_C depends on the size of the client buffer and client processor utilization. For the same CM application, variation in D_S and D_C can be kept marginal if the computing resources on server and client are not over utilized. However, streaming data over bandwidth constrained and error-prone wireless networks can cause unexpectedly high and fluctuating network uplink and downlink delays. In the next section, we take a deeper look at addressing network uplink and downlink delays, as they are the major challenges for response time in cloud based mobile applications.

4.2 Uplink Delay Optimization

The uplink transmissions of cloud based applications will primarily consist of control commands issued on the mobile devices. Due to the light traffic of control commands, the uplink delay will be seldom caused by network congestion. However, the user commands from the client to the cloud server must be delivered within a user acceptable latency threshold. This is not only due to its important effect on user experience, but also because any delayed delivery of the user commands may lead users to repeat actions, causing duplicate commands to be issued, and if acted on the cloud server, will lead to undesired and unpredictable effects. Therefore, to optimize the uplink delay, we should choose a proper transport layer protocol such that the user commands delivered over the uplink channel can be reliably and timely received by the cloud server.

The known reliability advantage makes the use of TCP attractive, but it may introduce unexpected delays due to packet retransmissions under noisy mobile channel conditions. While UDP does not have the problem of retransmission delay, it may fail to provide the reliability needed for transmitting the commands.

To address the above challenge, we have developed an enhanced UDP based protocol for CMG, which can ensure highly reliable but low latency uplink transmission of commands from mobile devices; it performs controlled re-transmissions of a command at the application layer to ensure reliability against packet loss and out-of-order delivery, with re-transmissions stopped when the user acceptable delay threshold is exceeded. In our proposed enhanced UDP based protocol, it first calculates the uplink delay threshold D_{UL}^{TH} based on downlink delay D_{DL}, client delay D_C and server delay D_S, and user-acceptable RT threshold RT_A:

$$D_{UL}^{TH} = RT_A - (D_S + D_{DL} + D_C) \qquad (2)$$

When the user issues a gaming command, the client will send this command to the server with an assigned index number using UDP. The server will send back an acknowledgement notification for the received command to the client through a TCP connection. In the condition when the client periodically retransmits the command with the same index number, it will stop the retransmission if the user issues a new command, or it receives the command acceptance notification from the server, or it does not receive the acceptance notification within the user-acceptable threshold.

Since the server has the knowledge of the command sequence from the index number, it can ignore the out-of-order and redundant commands. Moreover, a lost packet will not lead to a lost command, because the command in the lost packet can be recovered from the next coming packet, with a small increase of delay, which is the interval between two consequent packets. Therefore, besides the fast transmission speed achieved by UDP transportation, our proposed approach also provides reliability for the delivery of gaming commands.

Similar to the cloud mobile gaming application, the right transport protocol will need to be determined for other cloud based applications, based on each application's delay and reliability needs.

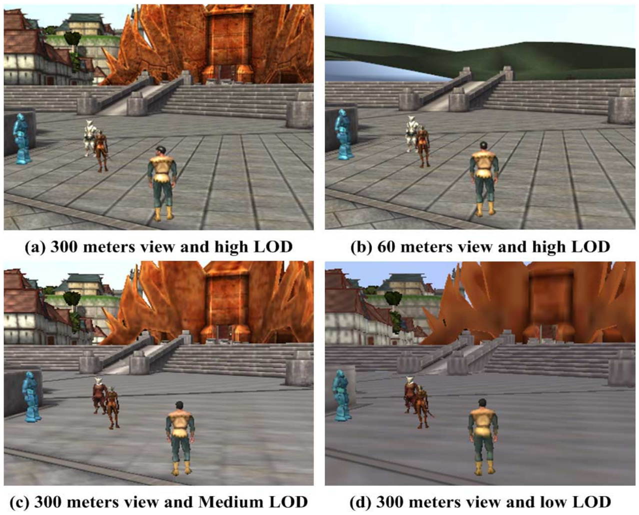
4.3 Application Adaptation Techniques to Address Downlink Congestion Delay

The network downlink delay of cloud based application is much more challenging than the uplink delay, because it has to deliver a large amount of application data from server to the client over a rate-varying wireless channel, which may lead to an unexpected downlink congestion delay. Video bit rate adaptation techniques [9][10][11] have been widely used to address downlink congestion delay, adjusting the compression level used and hence the encoding bit rate of streamed video to match fluctuating network bandwidth. However, these video rate adaptation techniques may not be always feasible for cloud based applications like CMG. Unlike other delay sensitive applications like real-time video streaming and video conferencing, CMG is not only delay sensitive, but also much less tolerant to loss in video quality and video frame rate, factors that are typically compromised by video streaming and video conferencing applications respectively to achieve low delay. Hence, we need to develop innovative techniques that can reduce the amount of data transmission needed from the cloud servers to the mobile clients, without additional data compression and hence without affecting encoded data quality.

One approach that is promising to meet the above requirement is adapting the content complexity at the cloud server so as to reduce the encoding bit rate needed without further compression and degradation in encoded data quality. For example, for the CMG approach, we have proposed a novel rendering adaptation technique, where we vary graphic rendering settings to adapt the resulting video complexity and hence the video encoding bit rate needed. There are mainly two principles to change rendering settings to affect the compressed video bit rate [1][12]. The first principle is to reduce the number of objects in the graphic scene, as not all of these objects are necessary for playing the game. For example, in a Massively Multiplayer Online Role-Playing Game (MMORPG), a player mainly manipulates one object, his avatar, in the gaming virtual world. Many other unimportant objects (e.g.

flowers, small animals, and rocks) or far way avatars will not affect the user playing the game. Removing some of these unimportant objects in the graphic scene will reduce the complexity of the resulting video frames, and thereby reducing the compressed video bit rate without compromising video quality. For example, one parameter we can adapt to reduce graphic objects is rendering view distance. Figure 9(a)(b) compare the visual effects in two different view distance settings (300m and 60m) in the game PlaneShift (PS) [2]. Note that the resulting video frame of Figure 9(b) has significantly less complexity than the video frame of Figure 9(a), and consequently needs much less encoding bits for the same level of compression as the video frame of Figure 9(a). The second key principle for rendering adaptation is related to the complexity of rendering operations. In the rendering pipeline, some of the operations have significant impact on content complexity, such as adjusting texture detail. If we can scale these operations, we will be able to adapt the compressed video bit rate as needed. For example, Figure 9(a)(c)(d) show the results of rendering with progressively reduced texture detail, with the resulting video frames needing progressively less encoding bits for the same compression level.

Though rendering adaptation is promising, it may not be appropriate to address all the wireless network challenges, in particular frequent variations in wireless network bandwidth during a gaming session. This is because game users may be sensitive if rendering complexities are changed too frequently too fast during a gaming session. To address this, we can take the help of encoding adaptation [1]. For relatively moderate network fluctuations, it may be better to apply encoding adaptation, while for large network bandwidth changes we can use rendering adaptation. We have developed a technique incorporating the above ideas, called Joint Rendering and Encoding Adaptation (JREA). To evaluate the effectiveness of our proposed adaptation techniques in addressing network congestion delay thereby ensuring user experience, we carried out multiple experiments using a 3G network and multiple test environments (locations, times) having different network conditions. Figure 10(a) shows the mobile network throughput measured for one such test environment. Figures 10(b)(c) present data collected from experiments when CMG is played in two scenarios: without using any adaptation technique and using our proposed JREA technique, respectively.


(a) 300 meters view and high LOD

(b) 60 meters view and high LOD

(c) 300 meters view and Medium LOD

(d) 300 meters view and low LOD

Figure 9: Screenshots of game PlaneShift in different settings of view distance and texture detail.

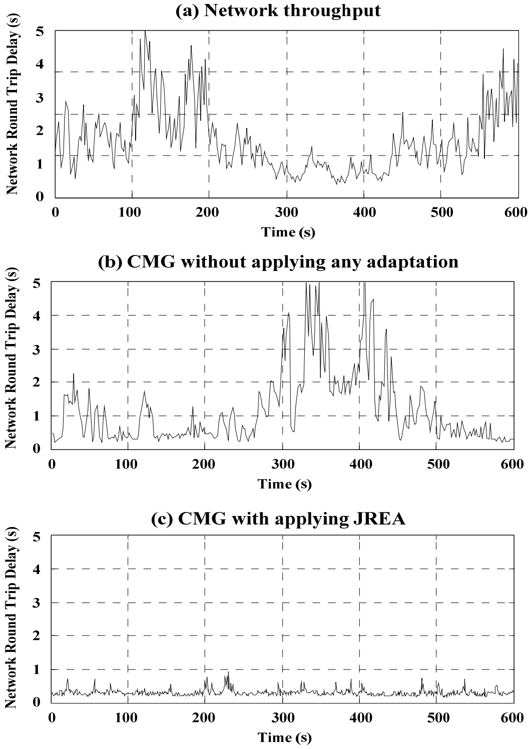


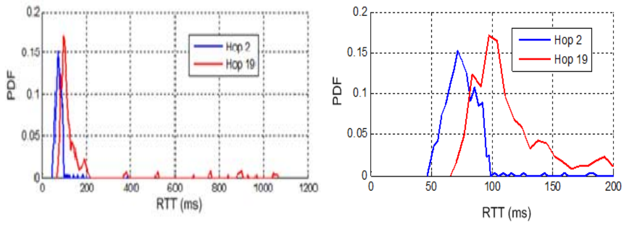
Figure 10. Experiment results of network round trip delay with and without applying adaptation techniques.

In the experiment without applying any adaptation technique, we encode and stream the rendered CMG video at 700kbps, which is sufficient for good video quality. However, as shown in Figure 10(b), network congestion leads to unacceptably high network round trip delay, and hence unacceptable response time., . On the other hand, Figure 10(c) shows that application of JREA can greatly improve the network round trip delay, thereby enabling acceptable response time.

4.4 Mobile Network Cloud: Computing and Storage at the Edge of the Mobile Network

The various experimental results presented in the paper point to the challenge of round-trip network delay associated with cloud mobile applications, between the mobile device and the Internet Cloud servers. However, the results do not pin-point which part(s) of the path between the cloud server and the mobile client (cloud server to/from base station, or base station to/from mobile device) is contributing most to congestion and the resulting delay. To get deeper insight, we conducted another experiment, where we used the traceroute tool to measure Round Trip Time (RTT) of packets from different hops/nodes in the path from the mobile client to an Amazon public cloud server. This experiment is conducted with both 3G and 4G LTE network of a US mobile carrier.

Traceroute shows some information (including IP address) of the intermediate hops within the path from mobile client to the Amazon data center. For 3G user, there are totally 19 nodes, of which 1-7 are in the carrier core network, 8-14 are associated with the Internet, and 15-19 are in the Amazon cloud; for 4G LTE



(a) (b)

Figure 11: RTT measurements with 3G network: (a) pdf of experienced RTT, (b) pdf of experienced RTT magnified for RTT of 0ms to 200ms.

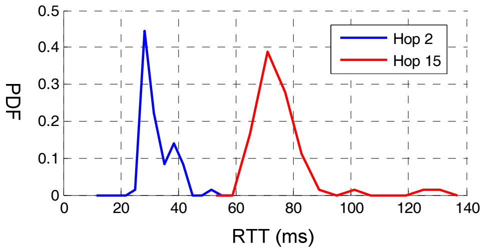


Figure 12: PDF of experienced RTT with 4G LTE network.

user, there are totally 15 intermediate nodes, of which 1-4 are in the carrier core network, 5- 9 are associated with the Internet, and 10-15 are in the Amazon cloud.

Figure 11(a) and Figure 11(b), magnified version of Figure 11(a), show the distribution of the RTT to hop 2 (within carrier core network) and hop 19 (within Amazon cloud) based on 500 rounds of running the traceroute command. The results show that the RTTs between the mobile client and the cloud node 19 can be significantly larger compared to the RTTs from the client to one of the nodes within the carrier networks. For example, RTTs of greater than 200ms are experienced from the cloud node, while not from node 2 inside the carrier core network. From Figure 11 (b), we can see the probability of experiencing RTT of 100ms or less is very high from node 2 (> .98), and quite low for node 19 (< 0.43). From Figure 11(a), it is also evident that the standard deviation of RTT from node 2 is significantly less than from node 19.

Figure 12 shows the distribution of the RTT to hop 2 (within carrier core network) and hop 15 (within Amazon cloud) when using the carrier LTE network. It is obvious that compared with 3G network, 4G LTE network provides a much better RTT performance, both at node 2 and node 15. The results show that the RTTs between the mobile client and the cloud node 15 can be significantly larger compared to the RTTs from the client to one of the nodes within the carrier networks. RTT of greater than 60ms is only experienced from the Amazon cloud node, not from node 2 inside the carrier core network. Node 2 will also experience a much more stable RTT than node 15.

From the above results, we can infer that if cloud processing data can be fetched from within the carrier network, as opposed to from Internet cloud servers, we can significantly reduce round-trip network delay, and thus significantly increase the probability of meeting aggressive response times, like 100ms needed by CMG for First Person Shooter (FPS) games. Such aggressive response time requirements may not be obtainable using Internet clouds.

Motivated by the results shown in Figure 11 and Figure 12, we propose the development of Mobile Network Clouds, bringing the benefits of cloud computing platforms to the edge of the mobile networks. A Mobile Network Cloud will consist of computing and storage resources supplementing the gateways in the Core Network (CN) and base stations in Radio Access Networks (RAN), and possibly carrier WiFi access points, so that content processing and retrieval can be performed at the edge of the mobile networks, as opposed to in Internet Clouds, thereby reducing round trip network latency, as well as reducing congestion in the wireless CN and RAN. Figure 13 shows a Mobile Network Cloud based on LTE network, with the Gateways (PGWs and SGWs) and Base Stations (BS) supplemented by small-scale Mobile Computing platforms (including computing and storage).

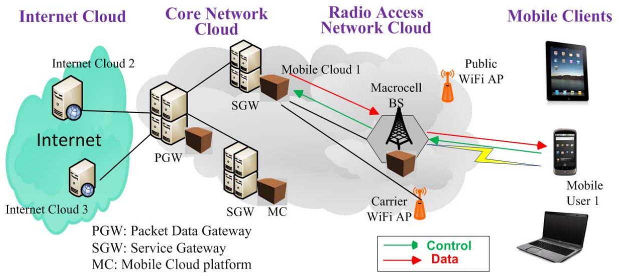


Figure 13: Mobile Network Cloud architecture.

Since there are thousands of base stations and access points, the proposed Mobile Network Cloud is a massively distributed network of much smaller computing and storage resources, as opposed to the more centralized architecture of Internet clouds consisting of a few data centers with much larger computing and storage footprints. The above difference has interesting implications and challenges. In [13], we investigated the use of Mobile Network Clouds, consisting of small caches in the RAN BSs to improve the latency of video delivery to mobile devices, and the capacity of networks to support concurrent video requests. We concluded that conventional caching techniques are not as effective due to the relatively small RAN cache sizes. Hence, we had to develop RAN aware caching techniques. Our simulation results show that Mobile Network Clouds, together with a suitable backhaul scheduling approach and RAN aware caching policies, can improve the probability of video requests that can meet initial delay requirements by almost 60%, and the number of concurrent video requests that can be served by up to 100% [13].

In [14], we consider a hierarchical Mobile Network Cloud like shown in Figure 13, where the SGW and PGW nodes (besides RAN nodes) are supplemented by caches, which can be beneficial to support user mobility across cells. Our investigations show that the hierarchical cache architecture, together with new caching policies, can enhance cache hit ratio of video requests by almost 25% compared to caching only in the RAN, without increasing the total size of caching used [14]. When considering mobility of the users across cells, almost up to 50% gain in capacity to support concurrent video sessions can be obtained. The above research illustrates the potential benefits of Mobile Network Clouds, and the potential challenges that need to be addressed to get the benefits.

As with CN and RAN caching to improve latency and capacity of mobile video delivery, the challenges of response time faced

Table 2. Example of Cloud Scheduling Options for User 1

Available Options For Mobile User 1	Mobile Cloud #1 (Avail. Capacity = 20%)	Internet Cloud #2 (Avail. Capacity = 60%)	Internet Cloud #3 (Avail. Capacity = 90%)
Macrocell Base Station (Avail. Capacity = 40%)	Choice A: 83ms	Choice B: 135ms	Choice C: 171ms
Carrier WiFi AP (Avail. Capacity = 25%)	Choice D: 74ms	Choice E: 127ms	Choice F: 167ms
Public WiFi AP (Avail. Capacity = 65%)	Choice G: NA	Choice H: 93ms	Choice I: 145ms

by cloud mobile applications like CMG and CMD can be potentially eased by adding suitable computing capabilities to CNs and RANs, such that some of the rendering and video processing tasks can be performed in the mobile network cloud closer to the edge.

And finally, the Mobile Network Cloud can be potentially extended to include the mobile devices themselves. Service discovery protocols for mobile ad-hoc networks [15], peer-to-peer content distribution techniques [16], and access technologies such as WiFi Direct [17] which enable practical ways to connect mobile devices, may be leveraged in the future to enable mobile devices to discover other mobile devices with required content or processing capabilities, and share content. The above capabilities can then be used by cloud mobile applications to consider mobile devices as caching and processing resources. For example, the Mobile Network Cloud can now not only cache content in the CN and RAN, but also opportunistically download and cache content on mobile devices when the mobile network has spare capacity. Subsequently, some of the requested videos may be discovered in the caches of the mobile devices, significantly improving response time and also improving network capacity.

4.5 Mobile Cloud Scheduling

Though it is promising to reduce response time by bringing cloud computing resources to the edge of the mobile networks, mobile users may not always be able to use the mobile cloud platforms due to capacity constraints, thus necessitating the use of the traditional Internet cloud resources also. Moreover, mobile network capacity will always be a challenge, in particular for mobile cloud applications which may require very high downlink bandwidth as discussed before. Therefore, besides meeting response time requirement of individual mobile cloud users, a new problem of mobile cloud scheduling needs to be addressed, which allocates mobile network and cloud resources in a way that maximizes system capacity, in terms of the number of mobile cloud users that can be scheduled with acceptable response time.

As shown in Figure 13, we assume the availability of heterogeneous access networks (like macrocell, microcell, carrier WiFi, public WiFi, etc.), and heterogeneous cloud resources - both mobile cloud as well as Internet cloud resources. Consequently, there may be multiple scheduling choices for a mobile cloud user. For instance, Table 2 shows the possible scheduling choices for mobile user 1 in Figure 13 at a given time, including the possible access networks and the available capacity for each access network, and the possible clouds and the available capacity for each cloud. For each pair of access network and cloud, we have provided the average network round-trip delay obtained from thirty measurements conducted using commercial wireless networks and clouds. The networks used in the measurements are AT&T 3G/4G (macrocell BS), AT&T WiFi (carrier WiFi AP), and Time Warner Cable WiFi (public WiFi AP). The selected clouds are the cloud associated with AT&T Packet data Gateway, Amazon cloud located in Southern California, and Amazon cloud located in Eastern US.

Among all the available choices for user 1, choice D has the lowest response time, though the current capacities of its access network and cloud are low. If minimizing the response time for cloud based application is the primary objective, then choice D should constitute the right scheduling decision However, taking into account response time without considering the effect of resource utilization may not be good for maximizing system capacity. For example, if we keep assigning mobile users to choice D, the carrier WiFi network or the mobile cloud, whose current capacities are already low, may become fully utilized, causing problems to schedule subsequent users which are only available to connect to carrier WiFi AP, or which need the fast response time afforded by the mobile cloud, thereby hurting overall system capacity. In contrast, if some of the mobile users are scheduled to the public WiFi network according to choice H (if their response time can be satisfied), the system capacity may be improved.

In addition, as we mentioned before, different applications may have different acceptable response time requirements RTA. Mobile cloud scheduling should consider the response time requirement when scheduling each individual user. For example, in Figure 13, if RTA of mobile user 1 is 100ms then choice H may be optimal, while if RTA of mobile user 1 is 200ms, choice I may be better as it has the highest access network and cloud capacity while having a response time below the RTA requirement. The tradeoff between satisfying response time for different user requests and maximizing system capacity will need to be considered for efficient mobile cloud scheduling.

Several techniques have been developed for cloud scheduling, which efficiently assign cloud resources (computation, storage, bandwidth) to application tasks, given the characteristics of the tasks (such as deadline constraints, priority, and quality-of-service requirements) and heterogeneous property of the computing resources typically available in the cloud, to achieve any or multiple of the following objectives: increase resource utilization, ensure fairness among requesting tasks [18], reduce cloud cost [19], and reduce energy consumption[20]. While cloud scheduling techniques can be potentially used for scheduling cloud based applications efficiently to Internet cloud resources, these techniques do not consider the mobile network constraints. On the other hand, the conventional mobile network scheduling techniques do not consider cloud computing constraints when making decisions to schedule users to network resources.

In [21], we have proposed a preliminary approach for mobile cloud scheduling, which simultaneously considers the availability of mobile network bandwidth as well as Internet cloud computing capacity to optimally allocate mobile bandwidth and cloud resources to mobile application users with different network and computing needs. To address the capacity challenge of cellular networks, the technique also utilizes WiFi networks when available. Preliminary results show that mobile cloud scheduling can significantly increase the number of simultaneous mobile cloud sessions whose response time requirements can be met. In the future, mobile cloud scheduling will need to leverage evolving heterogeneous access networks (HetNet), and mobile clouds besides Internet clouds, to maximize number of concurrent mobile cloud sessions that can satisfy their response time requirements.

5. CONCLUSIONS

In this paper, we take a comprehensive look at one of the key issues that need to be addressed to make cloud based mobile applications viable: response time. We perform extensive experimentations to understand the requirements and challenges of meeting response time for two promising cloud mobile applications. We discuss various solutions and directions to address response time of cloud mobile applications, and provide early experimental evidence of the promise of such solutions.

6. ACKNOWLEDGEMENTS

The material presented in this paper is based on work partially supported by the National Science Foundation under Grant No. CCF-1160832, Cisco Systems, Inc., and Intel Corporation.

7. REFERENCES

[1] Wang, S., Dey, S. Adaptive Mobile Cloud Computing to Enable Rich Mobile Multimedia Applications. *IEEE Transactions on Multimedia*, vol. 15, no. 4, Jun. 2013.

[2] Planeshift, http://www.planeshift.it/

[3] Cycon, H. L., et al. A Temporally Scalable Video Codec and its Applications to a Video Conferencing System with Dynamic Network Adaption for Mobiles. *IEEE Trans. on Consumer Electronics*, vol. 57, no. 3, pp. 1408-1415, Aug. 2011.

[4] Cycon, H. L., et al. Peer-to-Peer Videoconferencing with H.264 Software Codec for Mobiles. *WoWMoM08 – WS on Mobile Video Delivery (MoViD)*, 2008.

[5] Skype, http://www.skype.com/en/

[6] Citrix, http://www.citrix.com/

[7] Skype for CMG, http://youtu.be/ORrzAiHPUkw

[8] Cytrix for CMD, http://youtu.be/SHLn0GxzTNM

[9] Chang, S., and Vetro, A. Video Adaptation: Concepts, Technologies, and Open Issues. *Proc.of IEEE*, vol. 93, no. 1, pp. 148–158, Jan. 2005.

[10] Schwarz, H., Marpe, D., and Wiegand, T. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Trans. On Circuits System and Video Technology.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[11] Wang, S., Dey, S. Addressing Response Time and Video Quality in Remote Server Based Internet Mobile Gaming. In *Proc. of IEEE WCNC,* Sydney, Australia, Apr. 2010.

[12] Wang, S., Dey, S. Rendering Adaptation to Address Communication and Computation Constraints in Cloud Mobile Gaming. In *Proc. of IEEE GLOBECOM,* Miami, USA, Dec. 2010.

[13] Ahlehagh, H. and Dey, S. Video Caching in Radio Access Network. In *Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC 2012),* April 2012.

[14] Ahlehagh, H. and Dey, S. Hierarchical Video Caching in Wireless Cloud: Approaches and Algorithms. In *Proceedings of the 2012 IEEE International Conference on Communication (ICC'2012), Workshop on Realizing Advanced Video Optimized Wireless Networks (ICC'12 WS – ViOpt),* June 2012.

[15] Mian, A., Baldoni, R. and Beraldi, R. A Survey of Service Discovery Protocols in Multihop Mobile Ad Hoc Networks. In *IEEE Pervasive Computing*, pp. 66-74, January-March 2009.

[16] Theotokis, S. and Spinellis, D. A survey of peer-to-peer content distribution technologies. In *ACM Computing Surveys*, Volume 36 Issue 4, December 2004.

[17] Wi-Fi Direct™, http://www.wi-fi.org/discover-and-learn/wi-fi-direct%E2%84%A2

[18] Ghodsi, A., et al. Dominant Resource Fairness: Fair Allocation of Multiple Resource Types. *Berkerley Technical Report,* March 2011.

[19] Bossche, R., et al. Cost-optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workload. In *Proceedings of IEEE International Conference on Cloud Computing,* July 2010.

[20] Beloglazov, A. and Buyya, R. Energy Efficient Resource Management in Virtualized Cloud Data Centers. In *Proceedings of 2010 IEEE/ACM Conference on Cluster, Cloud and Grid Computing,* July 2010.

[21] Wang, S., Liu, Y., and Dey, S. Wireless Network Aware Cloud Scheduler for Scalable Cloud Mobile Gaming. In *Proc. of IEEE ICC,* Ottawa, Jun. 2012.