Impact of Cloudlets on Interactive Mobile Cloud Applications

Debessay Fesehaye, Yunlong Gao, Klara Nahrstedt
Department of Computer Science
University of Illinois at Urbana-Champaign, USA
{dkassa2,ygao12,klara}@illinois.edu

Guijun Wang
Boeing Research & Technology
The Boeing Company, USA
guijun.wang@boeing.com

***Abstract*—In this paper we analyze and study the impact of cloudlets in interactive mobile cloud applications. To study the impact we propose the design of cloudlet network and service architectures. Our study focuses on file editing, video streaming and collaborative chatting which are representative enterprise application scenarios.**

Initial simulation results show the performance gains of using cloudlets over using clouds in terms of data transfer delay and system throughput. When not more than two cloudlet wireless hops are used to transfer data, the cloudlet-based approach outperforms the cloud-based approach for all three application scenarios. With more cloudlet wireless hops under mobility, the cloud-based approach can give a better performance for some of the data transfers even though the cloudlet-based can outperform the cloud-based approach for most of the flows. In such scenarios, we suggest that an adaptive scheme should be used. For example, a scheme making an intelligent decision on either the cloudlet network or the cloud network, whichever gives minimum delay, can be used.

***Index Terms*—Interactive; cloud; cloudlet; mobile**

I. Introduction

Mobile communication systems have shown an explosive growth in past couple of years [30]. This growth will continue as users enjoy the convenience of mobility. Such growth of mobile systems demands very efficient ways of communications for all kinds of applications. The most common applications of mobile systems are interactive in nature. File editing, video streaming, chatting are all interactive enterprise applications.

Cloud computing has become the commonly used platform to serve these kind of applications. Cloud computing [17] is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Current cloud-based mobile systems use a topology as shown in Figure 1. Each node in an area uses the backbone network to connect to the main cloud and to communicate with a node in the same area or in another area.

However, such approaches of using the backbone network for every mobile communications can result in high latency. The bandwidth cost in 3G networks is higher than in WiFi [6]. Energy consumption of a mobile node using 3G/4G network is higher than a node using WiFi [2], [22]. Communication using

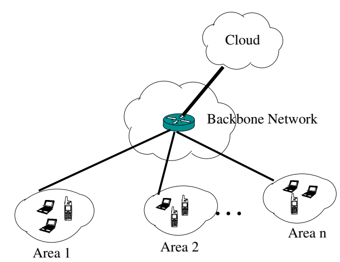

Fig. 1. Current mobile network topologies

backbone links may not be always present in some isolated areas, during rescue missions, uprisings and disaster scenarios [16]. More mobile communications using the same backbone network can also cause serious congestion at the backbone links and as a result, the average request (file) completion (transfer) time (AFCT) will be higher.

On the other hand, there is an increasing emergence of cloudlets [28] and a widespread presence of high bandwidth WiFi [2]. As defined in [28], *cloudlets* are decentralized and widely-dispersed Internet infrastructure whose compute cycles and storage resources can be leveraged by nearby mobile computers. A cloudlet may be a cluster of multi-core computers, with gigabit internal connectivity and a high-bandwidth wireless LAN. A cloudlet can also be a very powerful multi-core server with Internet connectivity depending on the application scenario.

Cloudlets have so far been proposed to assist mobile users, directly connected to them in terms of storage and processing [28]. However, to the best of our knowledge, none of the studies have shown the impact of cloudlets on interactive mobile cloud applications (IMCA) such as file editing, video streaming and chatting with mobile nodes moving from one cloudlet coverage to another cloudlet coverage area.

In this paper, we take the cloudlet usage scenario a step further. In addition to assisting mobile devices associated with them in storage and processing, cloudlets can also be used to cache and transfer content to mobile nodes using affordable wireless technologies such as WiFi and WiFi repeaters [4], [6] and/or Flashlinq [5]. Flashlinq can give a wireless range of upto at least 500 meters. In our scheme, the cloudlets are equipped with wireless access-point-like devices, similar to

1541-7719/12 $26.00 © 2012 IEEE
DOI 10.1109/EDOC.2012.23

what are called gateways in CloudTrax [4]. These devices connect a cloudlet with a cloud and other cloudlets (with gateways). Unlike CloudTrax which specializes in connecting multiple wireless nodes to the Internet using DSL, our focus is content-centric proximal (local) networking using cloudlets. Cloudlets can serve multiple mobile nodes in terms of processing, storage and content delivery.

As discussed in [4] every one hop repeater reduces the wireless bandwidth by roughly half. So upto two-hop wireless repeaters to reach a cloudlet can be a feasible WiFi range extender solution for an at least 54Mbps WiFi if the cloudlet of a mobile node can reach the destination cloudlet by a single hop. Mobile nodes also use cloudlets to communicate with each other. We consider a network of cloudlets which cover a specific area to serve IMCA, essentially forming a *proximal community network*. In this paper we evaluate the impact of such cloudlet network along with WiFi-like high bandwidth wireless technology on IMCA.

To study this impact we first present designs of network and service architectures for IMCA. In the cloud-based scenario, users (mobile nodes) use the main cloud to transfer data. In the cloudlet-based approach, the mobile nodes use the cloudlets to transfer data. The cloudlets serve as temporary service devices. They update the states of the mobile nodes in their respective clouds at the end of every communication with the mobile nodes. For example, for file editing, the file to be edited is first downloaded to the nearest cloudlet of the editing mobile node. When the editing is done, the edited file and associated states of the mobile node are uploaded to the main cloud.

To study scenarios where it is feasible to use cloudlets rather than cloud, we have implemented the cloudlet/cloud network and service architectures in the well-known NS2 [14] simulator. We consider a town mobility scenario and a periodic mobility scenario for our simulation. We then compare the cloudlet-WiFi-based communication against the cloud-based communication focusing on file editing, streaming and chatting. These interactive cloud applications, we consider in this study, use the transmission control protocol (TCP) [10]. We use the average request transfer delay and throughput metrics for the comparison. Our analysis scheme can be extended to other applications.

Simulation results show that when the nodes are static the cloudlet-based scheme gives a lower request transfer delay and higher throughput than the cloud-based approach, even if packets have to travel up to four cloudlet hops from the source to the destination. Under mobility scenario however, the cloudlet-based approach outperforms the cloud-based approach when the maximum number of cloudlet wireless hops is 2. When the maximum number of cloudlet wireless hops is 4, the cloudlet-based approach performs poorly for some of the requests made, though the cloudlet-based approach can outperform the cloud-based approach for most of the requests made. Under such a scenario we suggest that a scheme where each user registered at the main cloud uses either cloudlet network or cloud whichever gives higher throughput and lower delay for the interactive communications.

The main contributions of the paper are as follows:

- We propose mobile cloud infrastructure as a service (IaaS) for interactive mobile cloud applications (IMCA). The IaaS consists of cloud and cloudlets connecting to mobile nodes in a high bandwidth wireless communication (WiFi).
- To study the performance impact we propose and implement a new hybrid ad-hoc-like wireless routing protocol by modifying the Destination-Sequenced Distance-Vector routing (DSDV) [20] ad-hoc routing protocol in NS2.
- We then study the impact of cloudlets on IMCA comparing them with pure cloud-based communications. Our tradeoff analysis shows the benefits of cloudlets. The cloudlet-based approach results in lower request transfer delay and higher throughput when nodes are static and when the maximum number of cloudlet hops is 2 under mobility. When the maximum number of cloudlet wireless hops is higher than 2, the cloud-based scheme can outperform the cloudlet-based scheme. In this case an adaptive combination of both schemes is recommended.

The rest of the paper is organized as follows. In sections II and III we present the cloudlet network and service architectures. Some discussion of related work is given in section IV. Our simulation approach and results are discussed in section V. Finally, section VI presents a brief summary of the paper.

II. Cloudlet-Based Network Architecture

We assume the network architecture for today's interactive mobile cloud applications (IMCA) as shown by Figure 2, where mobile nodes use the backbone network via 3G/4G to communicate with servers and with each other. As discussed above, such a communication scheme can have many drawbacks. For instance the propagation delay for two nodes in the same geographical area to communicate with each other in such centralized 3G scheme can be higher than a communication using local cloudlets with high bandwidth wireless transmission.

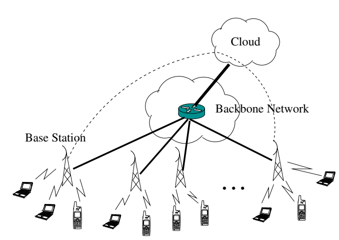

Fig. 2. User-Cloud Communication

To study the impact of cloudlets we first present the cloudlet-based network architecture as shown in Figure 3. In this architecture, we have mobile nodes affiliated with their nearest cloudlet (s) with long range wireless coverage. The mobile nodes can send data to or receive data from the reachable cloudlets or other mobile nodes in the network. We next discuss how packets are routed from one node in the cloudlet network to another node. The source and destination nodes can be the cloudlets or any of the mobile nodes.

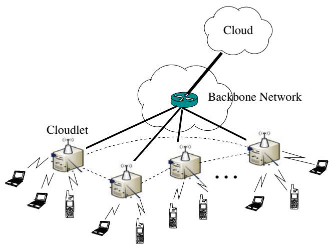

Fig. 3. User-Cloudlet Communication

A. *Routing Algorithm*

The cloudlet network is composed of cloudlets which have high bandwidth wireless functionality and many other mobile nodes connected with the cloudlets via wireless links. So the network consists of *heterogeneous* set of nodes with different roles. Any node can be mobile even though the results we report in this paper assume that cloudlets are stationary.

The routing algorithm for the IMCA is neither purely ad-hoc nor purely hierarchical. It is not purely hierarchical because all cloudlets are peers to each other and can have the same role. It is not purely ad-hoc as the mobile nodes are not involved in the routing of packets.

Cloudlets are used to communicate and process data on behalf of mobile users (other nodes). The cloudlets form a peer-to-peer network to communicate. The other mobile nodes use the cloudlets to communicate. Our routing design is simple and efficient. It excludes the mobile wireless users from the routing scheme. We propose two types of routing schemes namely *distributed* and *centralized* as discussed below.

1) Distributed Routing and Signaling Approach: In the distributed approach the routing algorithm is left to the cloudlets which form a peer-to-peer network. Routing updates are performed by the cloudlets and the routing tables are stored in the cloudlets saving energies for the mobile nodes.

Each cloudlet periodically broadcasts its presence information indicating that it is a cloudlet. Each other cloudlet can then propagate the reachability information of this cloudlet to other cloudlets. As all cloudlets exchange their reachability information, each of them constructs a routing table to reach each cloudlet in the given mobility area. Once this routing table to reach each cloudlet is constructed, other cloudlet routing updates can be triggered by any change in the cloudlet network. The change can be a new cloudlet added, a cloudlet removed or a cloudlet moved.

When a mobile user hears a broadcast message from a cloudlet, it enters the latest cloudlet ID into its cloudlet table for future communications. Each mobile user also periodically broadcast its ID (name). Each cloudlet in range of this mobile user registers the mobile user and broadcasts mobile node's ID to all its neighboring cloudlets. Each cloudlet then keeps a table of mobile node IDs along with the mobile node's cloudlet ID. Here it must be noted that each cloudlet broadcasts only the ID of the node associated with it without any hop count or other routing metric information of the node. This is because each cloudlet knows how to reach any other cloudlet from the routing table information the cloudlets exchange. So when a cloudlet receives a nodeID originating from a cloudlet with a cloudlet ID then it just stores this pair of nodeID and cloudletID.

When a mobile node wants to communicate with any other node, its cloudlet looks for the reachability information to that node using its routing table. If there are multiple such cloudlets, the source mobile nodes choose the cloudlet which gives the best routing metric (hop count). The initiator mobile node then sends communication request to the other nodes it wants to communicate with via its cloudlet.

The list of reachable mobile nodes can be made public or private, based on each mobile node's choice. This allows mobile nodes to be easily found by other nodes. The list of all nodes in the cloudlet network can also be made public with some or all requiring access password. The simulation results we obtain in this study are based on this distributed scheme.

2) Centralized Routing and Signaling Approach: In the centralized routing scheme for IMCA, routes are computed by a central server as is the case in openFlow [15]. Each cloudlet sends its ID and the ID of its reachable (neighboring) cloudlets to the central server. The central server periodically computes the routing table for each cloudlet and installs the forwarding tables into the cloudlets. Once the routing tables are computed by the server, consequent routing table computation can be triggered by new changes in the cloudlet network.

After mobile users register to a cloudlet, the cloudlet periodically sends the IDs (names) of its mobile users along with its ID to the centralized server. This is because some nodes may move out of the coverage of the cloudlet and others may join in. The central server then keeps a big table of mobile node names (IDs) along with the ID of their respective cloudlets. If the table size grows big, a hierarchical approach where some servers are responsible for some cloudlets can also be used.

When a node wants to communicate with another node, it sends the name of the node it wants to communicate with to its cloudlet. Its cloudlet asks the central server (network of servers) to look up the name of the requested node. The central server responds with the cloudlet of the requested node. In this case the central server serves as a proxy server used in session initiation protocol (SIP) [24]. Each cloudlet can also cache the list of nodes its users want to communicate with along with the cloudletID.

III. Cloudlet-Based Service Architecture

In section II, we presented the design of data routing architecture. To study the impact of cloudlets we also present a general design of service architectures for *file editing*, *video streaming* and *collaborative chatting* in this section. The description of this design is meant to discuss the type of service architecture under which we study the impact of cloudlets on mobile cloud applications.

A. File Editing

Users can directly edit a file, stored in the main cloud using remote connection. Part or all of the file to be edited can also be downloaded to a cloudlet and the mobile nodes can edit it by using wireless connection to the cloudlet. Typical applications involving file editing are document editing, database updating, facebook profile updating, google map/reduce framework and the like.

Remotely editing a file, stored in a cloud or cloudlet, entails constantly downloading and uploading a chunk of data from the file. A user downloads a chunk of data, edits it and uploads it to the cloud or cloudlet. This is similar to the way the Andrew File System (AFS) [9] works.

A file in a cloud or cloudlet can be edited by a single or multiple users. If multiple users are editing a file stored in a cloudlet, then a mobile node may have to connect to a cloudlet node using multiple cloudlet hops. This can happen when a node editing a file in one cloudlet moves to another cloudlet coverage before the editing is complete. To deal with data consistency issues in cases of multiple concurrent editing, global consistency management methods like those presented in [8] can be used.

We assume that a file to be edited by multiple users is divided into multiple small chunks with each chunk having its own read/write lock. For the case, where multiple users edit a file, our simulation experiment doesn't take into account the delay caused due to reader/writer locking of chunks of the file. We focus on the chunk transfer delay from the cloud or cloudlet to a user and vice versa after a user is granted a lock to edit the chunk.

The service architecture for editing file is given as follows.

- A node senses a nearby cloudlet via its wireless device and measures its throughput (available bandwidth) or delay to/from the cloudlet (WiFi). Available bandwidth estimation techniques such as the one discussed in [13], [27] can be used. If the file is located multiple hops away (multi-user editing), it measures the round trip delay or bandwidth to/from the cloudlet which contains the file.
- A node contacts its main cloud server for its file and measures its delay/throughput to/from its main cloud server.
- If the delay to the cloudlet is smaller, it downloads its file to the cloudlet and edits it in the cloudlet. Otherwise it edits it in the main cloud.
- The cloudlet servers upload the updates of the file to the main cloud of the node in according with a predetermined synchronization protocol and consistency management as discussed in [8].

B. Video Streaming

Video streaming here means mobile nodes view a video stored in the main cloud. If a cloudlet architecture is used, mobile users can view the cached video from their cloudlet. The service architecture for such video streaming is given as follows.

- A node senses a nearby cloudlet and measures its throughput or delay to/from the cloudlet (WiFi).
- A node contacts its main cloud server for its file and measures its delay/throughput to/from its main cloud server.
- If the delay to the cloudlet is smaller, the cloud streams the video to the cloudlet sequentially. Unlike the case of file editing, a mobile node can play the video while cloud is streaming video to the cloudlet.
- If the delay to the cloudlet is larger than the delay to the main cloud, the mobile node streams video content from the main cloud.

C. Collaborative Chat

Collaborative chat entails two or more nodes reachable nodes exchanging small size messages. As discussed in section II, a mobile node in a proximal community with cloudlet coverage can find another node in the community to chat with by contacting its cloudlet. This is because each of the cloudlets in the proximal area know how to reach with each other. They also exchange the list of mobile nodes registered with them. As discussed in section II, the list of mobile nodes is then made public or mobile nodes privately exchange their IDs with those they want to chat with. The service architecture for such short messaging systems is similar to the Extensible Messaging and Presence Protocol (XMPP) [26] protocol. It is presented as follows with either the cloudlet or cloud of a mobile node serving as an XMPP server as discussed in [11].

- A mobile node which wants to chat with other mobile node(s) in its community (within cloudlet coverage) contacts its cloudlet.
- The cloudlet displays the list of other active mobile nodes in the community after possibly some authentication with the contacting mobile node.
- The mobile node then sends a chat request to another node in the list via the cloudlet. If the other node a mobile node wants to chat with is not visible in the list, the requesting mobile node can still ask the cloudlet for the specific other mobile node it wants to chat with. The cloudlet can then forward the request to the destination node or send "node not found" message if the requested node is not active or if its security settings don't allow the specific request.
- The mobile nodes which want to chat then check their reachability and delay using their cloudlets. They also check their (3G/4G) delay and connectivity to the main cloud.
- If the cloudlets give better connectivity than the cloud, they chat on the cloudlet network or else they use the main cloud servers.
- When users finish chat using the cloudlet servers, the cloudlet servers save the states of the chatting users in their respective main cloud server.

IV. Related Work

Performance of 3G mobile data offloading through WiFi networks has been studied in [12]. The paper shows that if users can tolerate about a two hour delay in data transfer (e.g, video and image uploads), the network can offload 70% of the total 3G data traffic on average. A work by [1] shows that WiFi throughput is lower than 3G throughput in mobile environments, and WiFi loss rates are higher. This work also designs a system which switches from 3G to WiFi for delay tolerant applications. Some companies [29] have started to provide new mobile data offloading solution for 3G and Long Term Evaluation (LTE) networks. However these schemes don't consider a scenario where a network of cloudlets can be used for IMCA. Cloudlets can provide the additional benefit that some memory and energy intensive executions can be offloaded to them in ways discussed in [7]. LifeNet [16], a new ad hoc routing method that can handle transience such as node-mobility, obstructions and node failures has also been proposed. However, LifeNet doesn't use the concept of cloudlets as it doesn't distinguish between the nodes. Besides, LifeNet's main focus is reachability at the expense of throughput.

A *CoopNet* scheme for distributing streaming media content using cooperative networking was proposed by Padmanabhan, *et.al* [18]. The scheme distributes streaming media content, both live and on-demand to users who cooperate in the streaming. In CoopNet, when the central server is overloaded, it redirects new client requests to other clients which previously downloaded the content. Searching for content in a pure P2P system is done using expensive distributed search [23]. Content Distribution Networks (CDNs) [31] have also been widely used to distribute contents. CDNs can be formed using a network approach, where some logic is deployed in the network elements such as routers, switches to forward traffic to caching servers/proxies that are capable of serving client requests. Caching proxies within a CDN may also communicate with each other [19]. In contrast to CoopNet, pure P2P systems and CDNs, our focus in this paper is proximity networking where we leverage the use of cheaper wireless technologies such as WiFi [6] and Flashlinq [5] along with cloudlets to minimize delay and maximize throughput by reducing backbone network congestion. Besides, the cloudlet-based scheme we present in this paper is not a pure P2P as mobile nodes (none-cloudlets) are not used in forwarding of data. Like the CoopNet scheme the cloudlet-based scheme we discuss in this paper can use a central cloud server to locate multi-media or file content in one of the cloudlets which may already have the content.

V. Simulating IMCA in NS2

To evaluate the performance of the above cloudlet-based architectures, we have implemented a new wireless routing algorithm by modifying the DSDV implementation in NS2 [14]. NS2 offers the DumbAgent and the ad-hoc routing protocols. If a DumbAgent is used for the mobile users and ad-hoc routing is used for the cloudlets, then the DumbAgent doesn't allow the mobile users to even communicate with the cloudlets for reachability information. Hence, we modified the DSDV [20] code so that it excludes the mobile nodes from the routing process. In this modified version of DSDV only the cloudlets exchange the routing information and each mobile node uses the cloudlet to which it is associated as its default gateway.

A. Simulation Topology and Parameters

To study the impact of cloudlets on the interactive mobile cloud applications (IMCA), we consider a $670m \times 670m$ square shaped mobility region taken from the Champaign town where our university is located. This region is given by Figure 4. This selected area is where many of the main student centers in the town with many restaurants and recreation places.

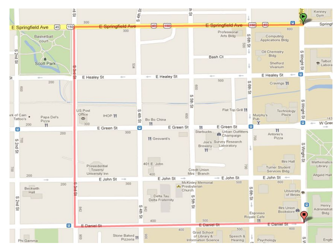

Fig. 4. Champaign Mobility Area

For the 3G/4G communication we consider one base station which we place at the center of the area as shown in Figure 5. The 3G/4G communication range is large enough to cover this mobility area. In the mobility area we can have all sorts of wireless devices. They can be laptops, smart phones or other wireless devices.

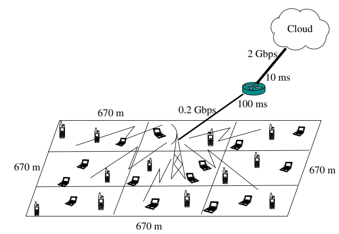

Fig. 5. Cloud-based topology

On the other hand, we consider multiple wireless enabled cloudlets to cover the same mobility area for the cloudlet experiments as shown in Figure 6. The number of cloudlets needed to cover the area depends on the communication range of the wireless devices in the cloudlets. In our experiments we consider two scenarios. The first one also shown in Figure 6 uses a transmission range of 250m. With this transmission range, we use 9 cloudlets to cover the area and to have cloudlet connectivity. We have also used a transmission range of 500m for the cloudlet scenario. Usually, the range of WiFi is up to 250m. This WiFi range can however be extended [6]. Wireless repeaters (range extenders) can be used to extend the WiFi range of mobile nodes to reach the target cloudlet. Old wireless routers can be used for this purpose for instance [21]. Besides, Flashlinq [5], which is a new licenced wireless technology, can give a coverage range of 500m. The applications we simulate are file editing, video streaming and collaborative chatting.

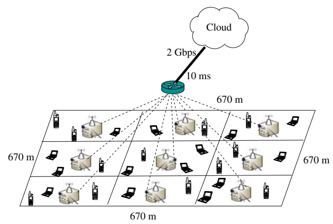

Fig. 6. Cloudlet-based topology

We consider a total number of 99 nodes in the mobility area. Each mobile node can launch up to a maximum of two applications at a time. The simulation experiments we conduct are for scenarios when the nodes (users of the interactive applications) are fixed and mobile. We consider two mobility models.

We use the area of interest (AOI) mobility scenario where all nodes move to the center of the mobility area at a speed of 11m/sec. The nodes then wait at the area of interest for 5 minutes = 300 seconds and then return to their original area at the same speed. We use the Champaign town are shown in Figure 4 for the mobility.

We have also created the Champaign town mobility model by modifying the Manhattan mobility model. In this mobility model nodes move along the two way streets in both directions following the Manhattan mobility model [25] within the Champaign mobility area given in Figure 4. We also used a mobility speed of 11m/sec for this scenario. We have used this town driving speed to evaluate how the applications are impacted by the maximum speed compared with nodes not moving.

B. Results: File Editing

To simulate file editing we consider an editing request size which is Pareto distributed with average 2KB and Pareto shape of 1.6. We allow scenarios where multiple clients can edit the same big file. The packet size in this study is 1000 Bytes. The file to be edited can be a database, a facebook profile, a company (enterprise) document, an svn document or a map/reduce type data. In the case of cloud-based approach, all users edit the file directly in the main cloud. In the cloudlet-based case, however, the requested file is first downloaded to one cloudlet and then edited by one or multiple users in the cloudlet network. As discussed in section III-A we assume that the file to be edited is divided into many small chunks which have their own read/write locks. When multiple users edit a file, the delay we discuss in this simulation doesn't include delays due to locking mechanisms. We focus on the delay from when a cloud or cloudlet sends a chunk to edit until when the user gets it and vice versa. So the delay we consider in our study is the chunk transfer delay from/to the user to/from the cloudlet or cloud.

In the simulation, the requests to edit are Poisson distributed with various mean requests/sec. So each user requests file editing at every exponential distributed intervals. We assume that when a user makes requests at every exponential time interval, the user receives a lock to edit the chunk. After a user finishes editing the requested size, it uploads it back to the cloud or cloudlet after an exponential time. In the AoI and Champaign town mobility scenarios for file editing we assume nodes travel at a speed of 1m/sec (pedestrian speed) and 11m/sec. The driving speed limit in the specified area in Champaign, which we consider as mobility area for our simulation, is 11m/sec.

Figures 7 through 14 show how the average file completion time (AFCT) and cumulative distribution function (CDF) of FCT of the cloud-based approach compare with the cloudlet-based approach for various cloudlet transmission ranges. Here by file we mean a single editing request. FCT of a file is the time from when the file starts until it finishes. AFCT of a file of some size is the average of the FCT of all files of that size during the simulation time.

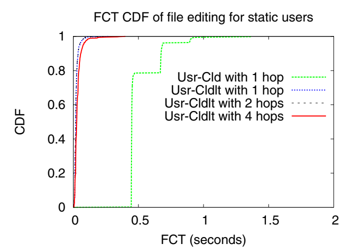

Fig. 7. Static Nodes: Cloud vs Cloudlet-based for File Editing $\lambda = 2\,req/sec$

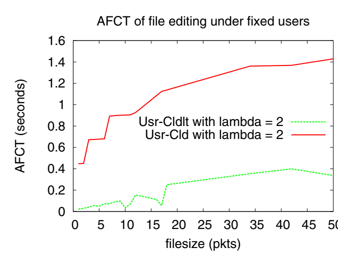

Fig. 8. Static Nodes: Cloud vs Cloudlet-based (4 hop) for File Editing $\lambda = 2\,req/sec$

For a fixed scenario, as shown in figures 7 and 8, the cloudlet-based approach achieves a smaller AFCT (request transfer time) and FCT, when compared against the cloud-based approach for even a wireless transmission range of 250m (with a maximum of 4 cloudlet hops) as shown in the figure. Hence, a user can on average finish editing a file faster using the cloudlet approach than using the cloud approach.

For the area of interest mobility scenario, the cloudlet-based approach outperforms the cloud-based approach as shown in

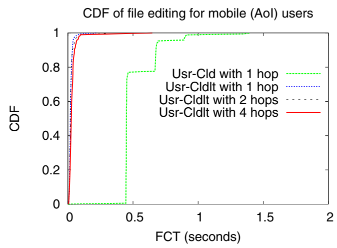

Fig. 9. AoI Mobility: Cloud vs Cloudlet-based for File Editing with $\lambda = 0.5\,req/sec$

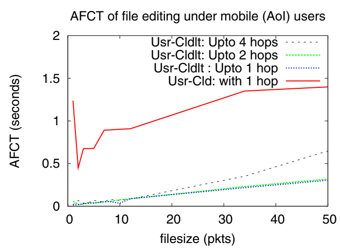

Fig. 10. AoI Mobility: Cloud vs Cloudlet-based for File Editing with $\lambda = 0.5\,req/sec$

figures 9 through 12. The cloudlet-based approach with a maximum of 2 cloudlet wireless hops outperforms the cloudlet-based approach with a maximum of 4 cloudlet wireless hops by providing smaller request transfer delay.

For the Champaign town mobility scenario as shown in figures 13 and 14 the cloudlet-based approach gives lower request (file) completion time than the cloud-based approach when the maximum number of cloudlet hops is 2. When the maximum number of wireless cloudlet hops is 4, the cloudlet-based approach provides lower editing request completion time (FCT) than the cloud-based approach for about 60% of the requests made as shown in the same figures 13 and 14. However, for over, 40% of the requests made under the Champaign mobility scenario, a cloudlet-based scheme with up to 4 hops results in very high request transfer time when compared with the cloud-based approach. This is because the cloud-based and cloudlet-based approaches with a maximum of 2 hops give a much higher coverage. This allows many nodes to stay under the same cloudlet coverage until they finish their transmission. On the other hand if the transmission range is very low, nodes have to cross multiple cloudlets. This results in higher transfer time as the destination nodes move from one network to anther resulting in many packet losses and TCP retransmissions. Figures 13 and 14 also show that a higher request rate of 1 req/sec results in higher FCT than the 0.25 req/sec.

One way to solve this low cloudlet performance when the cloudlet transmission range is small (250 m) is to increase the transmission range of the cloudlets using new technologies such as the Flashlinq [5] or longer range WiFi [6]. This is shown by figures 14 and 13 where high wireless transmission

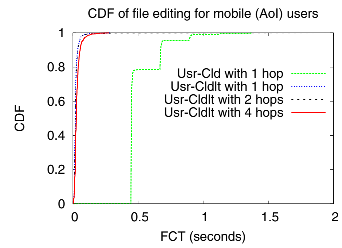

Fig. 11. AoI Mobility: Cloud vs Cloudlet-based for File Editing with $\lambda = 2\,req/sec$

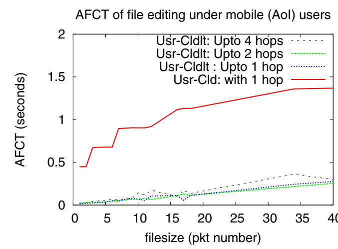

Fig. 12. AoI Mobility: Cloud vs Cloudlet-based for File Editing with $\lambda = 2\,req/sec$

range with a maximum of only 2 hops can outperform the cloud-based approach significantly.

Based on the simulation results we suggest that cloudlets use a maximum of 2 hops for file editing under a town mobility scenario. To achieve this, the transmission range of the cloudlets and the mobile devices should be set higher accordingly. With the growth of WiFi and Flashlinq type technologies, we think that this is not difficult to do.

C. Results: Video Streaming

For the simulation of video streaming we use a total video size which is Pareto distributed with mean of 8.4MB and Pareto shape of 1.6. We also use a maximum video size of 27MB. We obtained the mean and maximum video sizes from a study on YouTube [3] video size distributions. We allow scenarios where multiple clients can stream from the same video in the same cloudlet. For the cloud-based approach all users stream the video directly from the main cloud. In the cloudlet-based case, however, the requested video is first streamed to a cloudlet. While the video is being streamed to the cloudlet from the cloud, the mobile nodes can stream it from the cloudlet. The streaming requests are Poisson distributed with various λ mean requests/sec.

As in the file editing case discussed above, we consider the Champaign town area given by Figure 4 for the video streaming simulation. In the Champaign mobility scenario nodes travel at a speed of 11m/s. To evaluate the performance of video streaming for both the cloud-based and cloudlet-based approaches, we use the throughput metric. This throughput is the number of bytes received by the destination per second.

When all nodes are static, the cloudlet-based approach with a maximum of 2 cloudlet hops outperforms the cloud-based

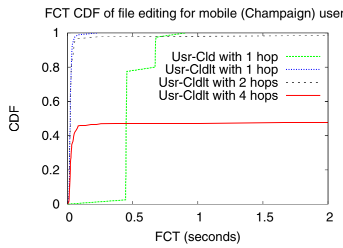

Fig. 13. Champaign Mobility: Cloud vs Cloudlet-based for File Editing with $\lambda = 0.25\, req/sec$

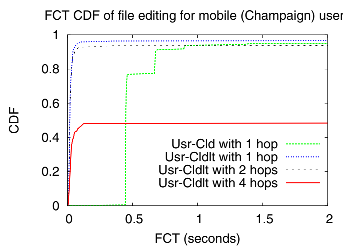

Fig. 14. Champaign Mobility: Cloud vs Cloudlet-based for File Editing with $\lambda = 1\, req/sec$

approach for most of the time as shown in Figure 15. The cloud-based approach for this static scenario shows a more consistent throughput which is better than the cloudlet-based approach with a maximum of 4 wireless cloudlet hops. With the increase in the simulation time the number of concurrent flows accumulates as each flow is of big size. This results in a lower throughput towards the end of the simulation time of 500 seconds we used in our experiments. Nonetheless, even at that simulation time when many videos don't finish streaming, the average throughput is above 30 KBps = 240 kbps which is also still within the YouTube videos bit rate range as shown in [3] which balances quality and streaming rate.

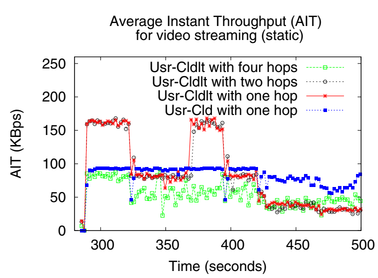

Fig. 15. Static Nodes: Cloud vs Cloudlet-based for Video Streaming with $\lambda = 0.08\, video\, streams/sec$

Under the mobility towards an area of interest (AoI Mobility) in the Champaign town mobility area shown in Figure 4, the cloudlet-based approach with a maximum of 2 hops gives a higher average instantaneous throughput (AIT) than the cloud-based approach for most of the simulation time as shown in Figure 16. The instantaneous throughput (IT) is obtained counting the number of stream bytes received every two simulation seconds. The AIT is the average of the IT of all flows within the 2 seconds interval. The interval we used can be any value other than 2 seconds. As more video streams start, the AIT of all schemes decreases towards a similar value. The cloud-based approach however outperforms the cloudlet-based scheme with up to 4 hops for most of the simulation time. This is mainly because the smaller the transmission range, the more a moving node goes out of the coverage of the cloudlet and the more cloudlet hops it needs for the video streams to go from one cloudlet to the destination node. As nodes move from one cloudlet coverage to another cloudlet coverage, the video packets can get delayed and lost until the mobile nodes associate themselves with the new cloudlets.

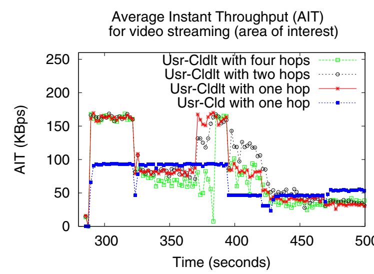

Fig. 16. AoI Mobility: Cloud vs Cloudlet-based for Streaming with $\lambda = 0.04\, video\, streams/sec$

For the Champaign town mobility scenario, the cloudlet approach with up to 4 hops does not look feasible as shown in figures 17 and 18. Reducing the number of hops by increasing the cloudlet coverage area can however solve this issue with cloudlets as shown in the plots. A cloudlet approach up to 2 hops can give a higher throughput than the cloud-based approach. One important thing to notice here is the size of the transmission. There doesn't seem to be a significant difference between the different cloud and cloudlet approaches as the simulation time progresses. The main reason for this is that the video streaming TCP sources dynamically adjust their window sizes in response to network delay and loss changes as TCP is designed to do.

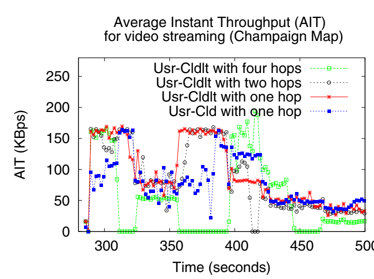

Fig. 17. Champaign Mobility: Cloud vs Cloudlet-based for Video Streaming with $\lambda = 0.04\, video\, streams/sec$

Figure 17 when compared with Figure 18 shows that with the increase in the request arrival rate to the cloudlet network,

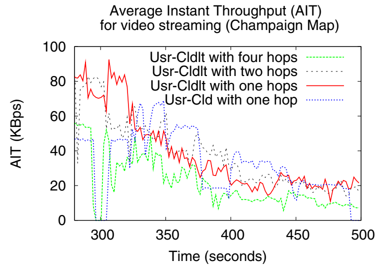

Fig. 18. Champaign Mobility: Cloud vs Cloudlet-based for Video Streaming with $\lambda = 0.08\, video\, streams/sec$

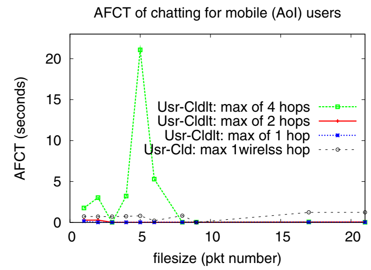

Fig. 20. AoI Mobility: Cloud vs Cloudlet-based AFCT for Chatting with $\lambda = 1.5\, req/sec$

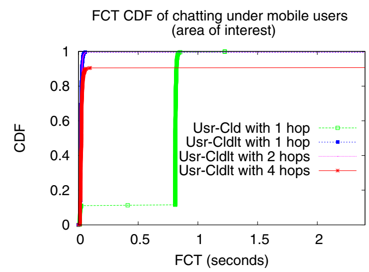

Fig. 21. AoI Mobility: Cloud vs Cloudlet-based CDF for Chatting with $\lambda = 1.5\, req/sec$

the AIT decreases further.

D. Results: Chat Messaging

To simulate collaborative chatting application, we used a chat message size which is Pareto distributed with average 140 Bytes and Pareto shape of 1.6. In cloud-based chatting two mobile nodes use the main cloud to chat. In cloudlet-based approach however the nodes use the cloudlets to exchange chat messages. Chat generation requests are Poisson distributed with various mean requests/sec. In the Champaign mobility scenario we consider in this study nodes travel at a maximum speed of 11m/s.

When the chatting nodes are static, the cloudlet-based approach provides a much smaller chat transfer time which we call average file completion time (AFCT) in the plots as shown in Figure 19 than the cloud-based approach.

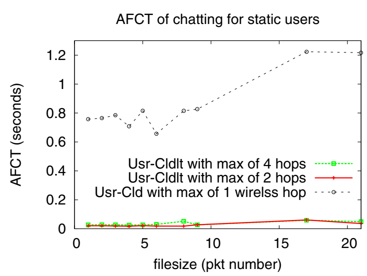

Fig. 19. Static Nodes: Cloud vs Cloudlet-based for Chatting with $\lambda = 2\, req/sec$

However under the AoI mobility scenario the cloudlet-based approach outperforms the cloud-based approach only if the number of cloudlet hops is not more than 2 as shown in Figure 21. When the number of cloudlet hops is up to 4, the cloudlet-based approach can provide a smaller chat transfer time for about 90% of the total chat requests made. As the distribution of file sizes is Pareto, there are more small size flows (flows whose size is around the mean). For some of these flows under the cloudlet-based approach, there are spikes in the chat transfer time (AFCT) as can be seen in Figure 20.

Similar to the AoI mobility scenario the cloudlet-based schemes can outperform the cloud-based schemes when the maximum number of cloudlet hops is 2 for the Champaign town mobility scenario. However, as can be seen from figures 22 and 23 the performance of the cloudlet-based approach using up to a maximum of 4 hops is a lot worse than the cloud-based approach.

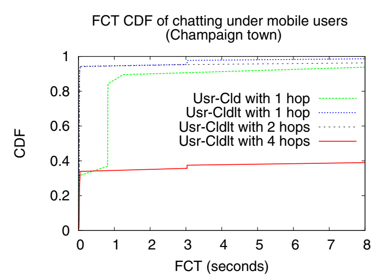

Fig. 22. Champaign Mobility: Cloud vs Cloudlet-based low load CDF for Chatting with $\lambda = 0.25\, req/sec$

One of the main reasons why the chat message transfer time is very high under the mobility scenarios is that both the sender and the receiver of the chat messages are mobile.

Our results agree with the results reported in [22] that TCP performs poorly under WiFi with mobility. In this scenario as discussed in the paper Multi-path TCP (MPTCP) [32] can be used to solve such mobility related TCP issues.

VI. Summary

In this paper we present the design of cloudlet network and service architectures (IaaS) to study the performance impact of cloudlets in interactive mobile cloud applications. Our study shows the benefits of cloudlets in reducing data (content)

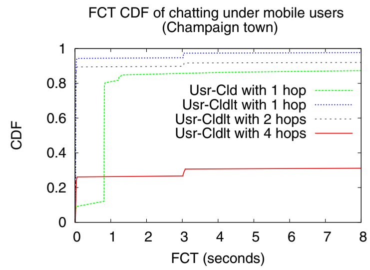

Fig. 23. Champaign Mobility: Cloud vs Cloudlet-based high load CDF for Chatting with $\lambda = 0.75\, req/sec$

transfer delay and in increasing content delivery throughput. The cloudlet-based approach always outperforms the cloud-based approach when the maximum number of cloudlet hops is 2. In scenarios where the maximum number of cloudlet hops is more than 2 (4 in our experiments), the cloudlet-based approach doesn't always outperform the cloud-based approach. So we suggest that if the cloudlet-based approach is to be used, the maximum number of cloudlet hops should not exceed 2. This can be achieved by using latest technologies such as the Flashlinq or by using WiFi repeaters.

REFERENCES

[1] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting mobile 3g using wifi. In *Proceedings of the 8th ACM international conference on Mobile systems, applications, and services*, MobiSys '10, pages 209–222, 2010.

[2] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, IMC '09, pages 280–293, 2009.

[3] X. Cheng, C. Dale, and J. Liu. Statistics and social network of youtube videos. In *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*, pages 229 –238, june 2008.

[4] CloudTrax. *CloudTrax (A cloud-based network controller) network planning guide*. CloudTrax, 2012.

[5] M. S. Corson, R. Laroia, J. Li, V. Park, T. Richardson, and G. Tsirtsis. Toward proximity-aware internetworking. *Wireless Commun.*, 17(6):26–33, Dec. 2010.

[6] R. Flickenger, S. Okay, E. Pietrosemoli, M. Zennaro, and C. Fonda. Very long distance wi-fi networks. In *Proceedings of the second ACM SIGCOMM workshop on Networked systems for developing regions*, NSDR '08, pages 1–6, 2008.

[7] X. Gu, A. Messer, I. Greenberg, D. Milojicic, and K. Nahrstedt. Adaptive offloading for pervasive computing. *IEEE Pervasive Computing*, 3(3):66–73, July 2004.

[8] T. Hara and S. Nishio. Global consistency management methods based on escrow approaches in mobile ad hoc networks. *Mob. Inf. Syst.*, 6(1):5–26, Jan. 2010.

[9] J. H. Howard, M. L. Kazar, S. G. Menees, D. A. Nichols, M. Satyanarayanan, R. N. Sidebotham, and M. J. West. Scale and performance in a distributed file system. *ACM Trans. Comput. Syst.*, 6(1):51–81, Feb. 1988.

[10] V. Jacobson. Congestion avoidance and control. In *ACM SIGCOMM '88: Symposium proceedings on Communications architectures and protocols*, pages 314–329, 1988.

[11] D. F. Kassa, K. Nahrstedt, and G. Wang. Analytical models of short-message reliability in mobile wireless networks. In *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, MSWiM '11, pages 369–376, 2011.

[12] K. Lee, I. Rhee, J. Lee, Y. Yi, and S. Chong. Mobile data offloading: how much can wifi deliver? In *Proceedings of the ACM SIGCOMM 2010 conference*, SIGCOMM '10, pages 425–426, 2010.

[13] M. Li, M. Claypool, and R. Kinicki. Wbest: A bandwidth estimation tool for ieee 802.11 wireless networks. In *IEEE Local Computer Networks.*, pages 374 –381, Oct. 2008.

[14] S. McCanne and S. Floyd. ns-2. http://www.isi.edu/nsnam/ns/.

[15] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, Mar. 2008.

[16] H. Mehendale, A. Paranjpe, and S. Vempala. Lifenet: a flexible ad hoc networking solution for transient environments. *ACM SIGCOMM Comput. Commun. Rev.*, 41(4):446–447, Aug. 2011.

[17] P. Mell and T. Grance. The NIST definition of cloud computing. *National Institute of Standards and Technology*, 53(6):50, 2009.

[18] V. N. Padmanabhan, H. J. Wang, P. A. Chou, and K. Sripanidkulchai. Distributing streaming media content using cooperative networking. In *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, ACM NOSSDAV '02, pages 177–186, 2002.

[19] M. Pathan and R. Buyya. *A Taxonomy of CDNs*, volume 9 of *Lecture Notes in Electrical Engineering*. Springer Berlin Heidelberg, 2008.

[20] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*, ACM SIGCOMM '94, pages 234–244, 1994.

[21] K. Purdy. Turn your old router into a range-boosting wi-fi repeater. http://lifehacker.com/5563196/turn-your-old-router-into-a-range+boosting-wi+fi-repeater, 2010.

[22] C. Raiciu, D. Niculescu, M. Bagnulo, and M. J. Handley. Opportunistic mobility with multipath tcp. In *Proceedings of the sixth international workshop on MobiArch*, ACM MobiArch '11, pages 7–12, 2011.

[23] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. *SIGCOMM Comput. Commun. Rev.*, 31(4):161–172, Aug. 2001.

[24] J. Rosenberg, J. Rosenberg, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. United States, 2002. RFC3261 Editor.

[25] G. S., S. Sarkar, and T. G. Basavaraju. Simulation based performance comparison of community model, gfmm, rpgm, manhattan model and rwp-ss mobility models in manet. In *Proceedings of the 2009 First International Conference on Networks & Communications*, IEEE NETCOM '09, pages 408–413, Washington, DC, USA, 2009.

[26] P. Saint-Andre. Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence draft-ietf-xmpp-3921bis-12). *XMPP Standards Foundation*, (12), 2010.

[27] C. Sarr, C. Chaudet, G. Chelius, and I. Lassous. Bandwidth estimation for ieee 802.11-based ad hoc networks. *IEEE Transactions on Mobile Computing*, 7(10):1228 –1241, oct. 2008.

[28] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4):14 –23, oct.-dec. 2009.

[29] L. Shemesh. Alvarion announces new mobile data offloading solution for 3G and LTE Networks. http://finance.yahoo.com/news/Alvarion-Announces-New-Mobile-bw-2111179422.html, 2012.

[30] S. Theodoropoulou. More mobile-ready devices and mobile video to fuel 39-fold global growth from 2009-2014. http://www.cisco.com/web/MT/news/10/news_220210.html, 2010.

[31] D. C. Verma. *Content Distribution Networks: An Engineering Approach*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[32] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath tcp. In *USENIX NSDI'11*, Berkeley, CA, USA, 2011.