





### 3.2 Cloudlet Architecture

One of the key issues in mobile cloud computing is the end to end delay occurring in servicing a request. Since the cloud services and resources in Internet service provider is distant from the mobile clients network latency is increased. The proposed an architecture for mobile cloud computing called Cloudlet, to reduce the bandwidth induced delay between devices and cloud. A cloudlet is a set of computers connected to the internet and is accessed by the nearby mobile devices using a Wi-Fi or WLAN. In this architecture the mobile devices act as a thin client and all the computations occur in the cloudlet. Fig 2 shows cloudlet architecture. The basic idea of this approach is to reduce the distance between the mobile users and cloud services. By this architecture mobile users can access the cloudlet which is one hop away, thus reduces bandwidth utilization and efficiency.

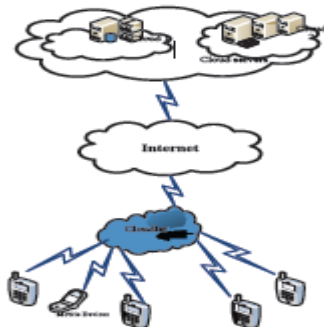


Figure 2: Mobile Cloudlet Architecture

To overcome the challenges of mobile devices, parts of computation and data storage is migrated to resource providers outside the mobile device. Offloading transfers the information needed for processing and storage to a remote server which will complete the computation task and send the results back to the mobile client. The different offloading techniques currently available are client server communication, virtualization and mobile agents [20]. Cloudlet architecture reduces the gap between the mobile devices and remote servers by offloading its workload to a local cloudlet with connectivity to remote cloud servers. Offloading in cloudlet infrastructure is based on a Virtual Machine (VM) technology. To create a transient client software environment cloudlet uses dynamic VM synthesis. In this approach the mobile device transmits a small VM overlay to the cloudlet and applies it to a compatible base VM to generate the Launch VM. Launch VM is the virtual environment temporarily created for a mobile client to execute the task. After execution results are given back to the mobile client. The cloudlet infrastructure is restored to its previous state after each execution.

The performance of cloudlet depends mainly on two factors. Overlay transmission time and overlay synthesis. Overlay transmission time can be improved by using a higher bandwidth wireless network. The different techniques proposed in [7] includes partitioning the overlay in to different chunks that can be executed in parallel, caching and perfecting and pipelining i.e. pipelining the overlay execution with transmission. In this paper we discuss a cooperative cache design for the cloudlet architecture. Mobile cloud computing has found wide applications in

many areas like speech synthesis, natural language processing, image processing, augmented reality, information sharing ,information searching , social networking, etc. While many applications like information sharing or social networking are not dependent on the speed of processing, some computation intensive applications like augmented reality, image processing demand high level of responsiveness. Cooperative caching tries to improve the response time by reducing VM synthesis time by caching previous states. If the users that use cloud services have similar interest, cooperative caching increases the response time considerably. A language translator is an interesting application, which we could look into. This is a useful tool for foreign travelers. Using mobile cloud computing, different words, sentences or paragraphs can be independently processed in the cloud. Commonly used words or sentences will be available in the local cache, which can be accessed faster during subsequent searches, thereby improving the responsiveness of the system.

Cooperative caching consists of multiple distributed caches to improve system response time. Having distributed caches permits a system to deal with concurrent client request as well as sharing contents. We can also reduce response time by concurrently retrieving objects from different cache sites. Concurrent retrieval of objects from different cache sites is beneficial as opposed to the remote cloud server which will result in latency and bandwidth issues.

In cloudlet, when a mobile client requests for a cloud service, the network searches for data in the local cloud. If the service is not available, the users should contact a distinct cloud which involves network transfers and latency. If we are able to cache different VM synthesis states, the users can get the service from cache. When the object is not present in the cache, request is given to the base layer to get the corresponding launch VM. If the corresponding base VM is not present, we have to contact distant cloud for the service. Thus data caching also increases battery life in mobile devices by reducing wireless communication.

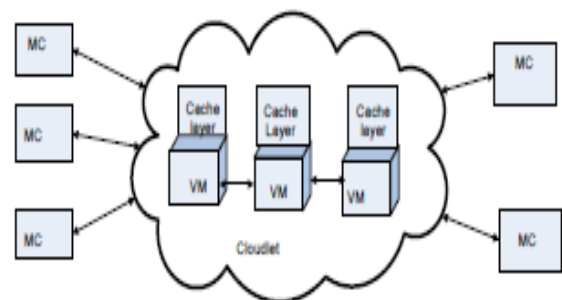


Figure 3: Cooperative Cache Framework

### 3.3 Optimal Solutions

For contents with changeable level of popularity, a greedy approach for each node would be to store as many distinctly popular contents as its storage allows. This approach sums to noncooperation and can grow to heavy network-wide data duplications. In the other excessive case, which is fully cooperative, a terminal would try to make the best of the total number of single contents stored within the SWNET by avoiding duplication.

In this paper, we show that none of the above excessive approaches can reduce the content provider's charge. We also show that for a given rebate-to-download-charge ratio, there is present an object placement policy which is somewhere in between those two ends, and can increase the content provider's cost by striking a stability between the greediness and full cooperation. This is referred to as optimal object placement policy in the rest of this paper. The proposed cooperative caching algorithms strive to attain this best object placement with the target of reducing the network-wide content provisioning price.

### 3.4 User Selfishness

The probability for earning peer-to-peer rebate may encourage selfish activities in some clients. A selfish client is one that diverges from the network-wide finest policy in order to receive more rebates. Any distinction from the optimal policy is expected to incur higher network-wide provisioning cost. In this work, we revise the impacts of such selfish behavior on object provisioning cost and the earned refund within the context of a SWNET. It is given that beyond a threshold selfish node population, the amount of per-node rebate for the selfish users is lower than that for the unselfish users. In supplementary terms, when the selfish terminal population exceeds a certain point, selfish actions discontinue producing more advantage from a refund standpoint.

## 4. Network, Service and Pricing Model

Network Model Fig.4 describes a model SWNET within a University grounds. People carrying mobile devices form SWNET partitions are the end consumers, which can be whichever multi-hop (i.e. MANET) as shown for partitions 1, 3, and 4, or single hop contact point based as shown for partition 2. A movable device can download some data (i.e., content) from the CP's server using the CSP's cellular network system, or from its home SWNET partition. In the remaining paper, the terms object and content are used synonymously. We regard as two types of SWNETs. The foremost one involves motionless SWNET partitions. Meaning, after a partition is formed, it is maintained for sufficiently long so that the cooperative object caches can be formed and reach fixed states. We also consider a second type to explore as to what happens when the still assumption is relaxed. To investigate this effect, caching is applied to SWNETs formed using human interaction traces obtained from a set of real SWNET nodes.

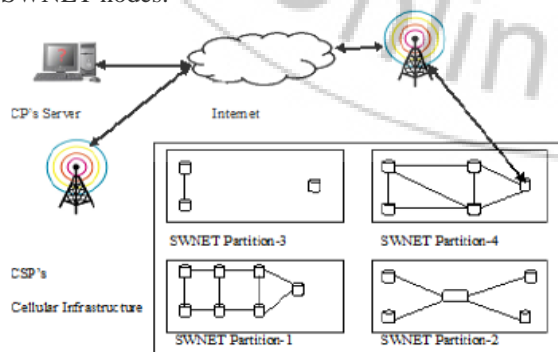


Figure 4: Content access from a SWNET in a University Campus

### 4.1 Search Model

After an object call is originated by a mobile tool, it first finds in its local cache. If the local search fails, it searches the object within its SWNET division using limited transmit note. If the search in division also fails, the data is downloaded from the CP's server using the CSP's 3G/4G cellular arrangement. In this paper, we have designed objects such as electronic books, music, etc., which does not vary on time, and therefore cache constancy is not a serious issue. We first suppose that all objects have the equivalent size and each terminal is able to store up to "C" dissimilar data in its cache. Later on, we let go this supposition to sustain objects with variable size. We also believe that all objects are popularity-tagged by the CP's server. The popularity-tag of an object points out its universal recognition; it also indicates the chances that a subjective request in the network is produced for this specific object.

### 4.2 Pricing Model

We use a pricing model similar to the Amazon Kindle business model in which the CP pays a download cost  $C_d$  to the CSP when an End-Consumer downloads an object from the CP's server through the CSP's cellular network. Also, whenever an EC provides a locally cached object to another EC within its local SWNET partition, the provider EC is paid a rebate  $C_r$  by the CP. Optionally, this rebate can also be distributed among the provider EC and the ECs of all the intermediate mobile devices that take part in content forwarding. The selling price is directly paid to the CP by an EC through an out-of-band secure payment system. A digitally signed rebate framework needs to be supported so that the rebate recipient ECs can electronically validate and redeem the rebate with the CP. We assume the presence of these two mechanisms on which the proposed caching mechanism is built.

## 5. Results



Figure 1: Admin Login



Figure 2: File Upload



Figure 3: User Searching the File



Figure 4: Downloading a File

## 6. Conclusion

The objective of this work was to develop a cooperative caching strategy for provisioning cost minimization in Social Wireless Networks. The key contribution is to display that the best cooperative caching for provisioning cost reduction in networks with homogeneous content demands requires an optimal split between object duplication and individuality. Furthermore, we experimentally (using simulation) and analytically evaluated the algorithm's performance in the presence of user selfishness. It was shown that selfishness can increase user rebate only when the number of selfish nodes in an SWNET is less than a critical number. It was shown that with heterogeneous requests, a benefit based heuristics strategy provides better performance compared to split cache which is proposed mainly for homogeneous demand. Ongoing work on this topic includes the development of an efficient algorithm for the heterogeneous demand scenario, with a goal of bridging the performance gap between the Benefit Based heuristics and the centralized greedy mechanism which was proven to be optimal. Removal of the no-collusion assumption for user selfishness is also being worked on.

## References

- [1] M. Zhao, L. Mason, and W. Wang, "Empirical Study on Human Mobility for Mobile Wireless Networks," Proc. IEEE Military Comm. Conf. (MILCOM), 2008.
- [2] "Cambridge Trace File, Human Interaction Study," <http://www.crowdad.org/download/cambridge/haggle/Exp6.tar.gz>, 2012.
- [3] E. Cohen, B. Krishnamurthy, and J. Rexford, "Evaluating Server-Assisted Cache Replacement in the Web," Proc. Sixth Ann. European Symp. Algorithms, pp. 307-319, 1998.
- [4] S. Banerjee and S. Karforma, "A Prototype Design for DRM Based Credit Card Transaction in E-Commerce," Ubiquity, vol. 2008, 2008.
- [5] L. Breslau, P. Cao, L. Fan, and S. Shenker, "Web Caching and Zipf-Like Distributions: Evidence and Implications," Proc. IEEE INFOCOM, 1999.
- [6] C. Perkins and E. Royer, "Ad-Hoc On-Demand Distance Vector Routing," Proc. IEEE Second Workshop Mobile Systems and Applications, 1999.
- [7] S. Podlipnig and L. Boszormenyi, "A Survey of Web Cache Replacement Strategies," ACM Computing Surveys, vol. 35, pp. 374-398, 2003.
- [8] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of Human Mobility on Opportunistic Forwarding Algorithms," IEEE Trans. Mobile Computing, vol. 6, no. 6, pp. 606-620, June 2007.
- [9] "BU-Web-Client - Six Months of Web Client Traces," <http://www.cs.bu.edu/techreports/1999-011-usertrace-98.gz>, 2012.
- [10] A. Wolman, M. Voelker, A. Karlin, and H. Levy, "On the Scale and Performance of Cooperative Web Caching," Proc. 17th ACM Symp. Operating Systems Principles, pp. 16-31, 1999.
- [11] S. Dykes and K. Robbins, "A Viability Analysis of Cooperative Proxy Caching," Proc. IEEE INFOCOM, 2001.
- [12] M. Korupolu and M. Dahlin, "Coordinated Placement and Replacement for Large-Scale Distributed Caches," IEEE Trans. Knowledge and Data Eng., vol. 14, no. 6, pp. 1317-1329, Nov. 2002.
- [13] L. Yin and G. Cao, "Supporting Cooperative Caching in Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. 1, pp. 77-89, Jan. 2006.
- [14] Y. Du, S. Gupta, and G. Varsamopoulos, "Improving On-Demand Data Access Efficiency in MANETs with Cooperative Caching," Ad Hoc Networks, vol. 7, pp. 579-598, May 2009.
- [15] C. Chow, H. Leong, and A. Chan, "GroCoca: Group-Based Peer-to-Peer Cooperative Caching in Mobile Environment," IEEE J. Selected Areas in Comm., vol. 25, no. 1, pp. 179-191, Jan. 2007.
- [16] F. Sailhan and V. Issarny, "Cooperative Caching in Ad Hoc Networks," Proc. Fourth Int'l Conf. Mobile Data Management, pp. 13-28, 2003.
- [17] C. Chow, H. Leong, and A. Chan, "Peer-to-Peer Cooperative Caching in Mobile Environments," Proc. 24th Int'l Conf. Distributed Computing Systems Workshops, pp. 528-533, 2004.
- [18] A. Schrijver, Theory of Linear and Integer Programming. Wiley-Interscience, 1986.
- [19] H.K. Kuhn, "The Hungarian Method for the Assignment Problem," Naval Research Logistics, vol. 52, no. 1, pp. 7-21, 2005.
- [20] N. Laoutaris et al., "Mistreatment in Distributed Caching: Causes and Implications," Proc. IEEE INFOCOM, 2006.
- [21] B. Chun et al., "Selfish Caching in Distributed Systems: A Game-Theoretic Analysis," Proc. 23th ACM Symp. Principles of Distributed Computing, 2004.
- [22] M. Goemans, L. Li, and M. Thottan, "Market Sharing Games Applied to Content Distribution in Ad Hoc Networks," IEEE J. Selected Areas in Comm., vol. 24, no. 5, pp. 1020-1033, May 2006.



## Author Profile



**Mr. A. Kalyan Kumar** received B. Tech. degree in Computer Science and Engineering from Chebrolu Engineering College Guntur from JNTU Kakinada University in 2012 currently. He is doing M. Tech. in Prakasam Engineering College, from JNTUK University, Kakinada, India.

**K. V. Srinivasa Rao** received M.TECH degree in Computer Science and Engineering from JNTUA University, and currently he is working as an Associate Professor, Department of CSE in Prakasam Engineering College, Kandukur, India

