A Testbed Implementation of Hybrid Decision Model Based Seamless Lightweight Vertical Handover

Tusher Chakraborty[1], Md. Shohrab Hossain[1], and Mohammed Atiquzzaman[2]
[1]Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh
[2] School of Computer Science, University of Oklahoma, Norman, Oklahoma, USA
Email: tusherchakraborty@matholympiad.org.bd, mshohrabhossain@cse.buet.ac.bd, atiq@ou.edu

***Abstract*—The next generation ubiquitous wireless networks will comprise of mobile users moving between heterogeneous networks with terminals having multiple access interfaces and services. Vertical handover techniques deal with the scenario where a user seamlessly moves between different network access technologies. The most crucial issue in such environment is seamless connectivity. Earlier works demand for infrastructural modification with cumbersome implementations. Here, we propose an automated, transparent, seamless, hybrid decision model based easy-to-implement vertical handover mechanism between WiFi and cellular network. The implementation of the mechanism does not supplant the current infrastructure. We carry out experiments on a testbed to demonstrate the efficacy of our proposed scheme. Experimental results show that a handover-held communication slightly differs from a handover-less communication in terms of diversified network metrics.**

***Index Terms*—Seamless Vertical handover, Testbed implementation, transparent handover, hybrid decision model.**

I. Introduction

There has been increase in interest in interconnection of heterogeneous access technologies to create a common hybrid network. However, the foremost task is to interconnect these diversified networks to provide seamless connection to mobile wireless devices. The deployment of miscellaneous technologies with evolving mobile terminals having multiple network interfaces has allowed users with access to the services at any time through any type of network [1]. The next generation wireless systems will comprise of a heterogeneous environment with different access networks technologies that differ in bandwidth, cost, etc.

Handover is the process of transferring an ongoing data session from one channel connected to the core network to another channel. Here, mobility and user preferences are crucial issues. The foremost challenge of handover is to make it seamless, i.e., without seeking user notice. Soft handover resolves this key challenge where a new connection is activated before the old connection is broken, i.e., make-before-break.

The mobility and user preference issues are resolved by vertical handover [2]. Vertical handover refers to a network node changing the type of connectivity it uses to access a supporting infrastructure, usually to support node mobility. Vertical handover goes through three basic steps: (1) initiation, (2) decision, and (3) execution. In initiation step, mobile terminal (or network controller) must know which wireless networks are reachable. Decision step evokes selection through a vertical handover decision algorithm [3], [4], [5], [6] of the access point on the basis of proper performance metrics (e.g., RSSI, network connection time, available bandwidth, power consumption, cost, security level, user preferences). Execution step invokes mechanism of switching between networks.

A sizable proportion of the world's population is under cellular network coverage. In parallel, WiFi deployment is swelling up. Schools are providing WiFi services to their students, cities are availing WiFi services for their citizens, many homes use WiFi routers, and even coffee shops are offering WiFi for their customers. The ubiquitous availability of cellular and WiFi networks would offer transcendent user experience if mobile devices could seamlessly switch and access both of these technologies. Here, the seamless vertical handover between WiFi and cellular network plays a vital role.

Over the years, researchers have shown profound interest in developing vertical handover mechanism [7], [8], [9], [10]. However, in most of the cases, they are not feasible for the deployment using existing infrastructure. Most of these proposals demand for modification of current infrastructure, complex hardware, high energy consumption. Moreover, they are not suitable for practical mobile devices.

We propose an automated, transparent, seamless, lightweight, bidirectional vertical handover mechanism between WiFi and cellular network. Nearly all vertical handover mechanisms from state-of-the-art literature focus on user preference, i.e., monetary cost optimality of network usage or attempts to ensure better QoS. Here, our proposed handover decision algorithm invokes hybrid model of user-centric (UC) strategy and multiple attribute decision (MAD) strategy. Hence, a common platform is established to simultaneously consider both user preference and QoS. Moreover, the handover mechanism can be implemented using existing Internet infrastructure. Nonetheless, the handover mechanism supports both TCP and UDP protocol.

We summarize our *contributions* as follows:

- We propose a seamless handover mechanism between WiFi and cellular networks. We conduct sundry testbed experiments to substantiate our proposal through constructing a chat group with existing Internet infrastructure, where each client is able to evoke our proposed

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

handover mechanism.

- We conduct experiment considering four networking scenarios : (1) WiFi connection only, (2) Cellular connection only, (3) WiFi to cellular handover, and (4) Cellular to WiFi handover. Nonetheless, we conduct experiment for both TCP file transfer and UDP stream data transfer.
- At the end, we present a comparison between handover-held and handover-less situations based on our experimental results through using assorted networking metrics such as throughput, RTT, delivery ratio, inter-packet arrival delay, etc.

Our experimental results show that the proposed handover mechanism influences aforementioned networking metrics in a diminutive scale.

This lightweight handover mechanism can be adopted by any cellphone based video conference, VOIP, chat, video surveillance, etc., service providers to offer seamless alternative access of WiFi and cellular network, to the subscribers, without any infrastructural modification.

The rest of the paper is organized as follows. Section II discusses state-of-the-art in handover algorithms, Section III introduces our proposed handover mechanism, followed by the experimental setup and result analysis in Section IV. We discuss our limitation and future work in Section V. Finally, Section VI has the concluding remarks.

II. Related Work

Mobile wireless network technology can gain immense popularity by providing pervasive network access to users on the move. In next generation wireless network, mobile users are connected to the best available networks that suit their service requirements and switch between different networks based on their service needs. Vertical handover offers this switching between networks. Various strategies [4] have been proposed to handle vertical handovers between heterogeneous wireless networks. Here, the key challenge is to make handover seamless and transparent to users.

To address the challenge, IEEE 802.21 working group has specified an open standard called MIH Services [11]. The purpose of IEEE 802.21 is to improve the user experience by providing a middleware, i.e., MIH layer, which facilitates both mobile-initiated and network-initiated handovers. This functionality ensures seamless connectivity across both similar and different access technologies. However, an extra layer of implementation adds to its complexity and still needs more experimental study.

Mobile IP (Mobile IPv6) [12] enables transparent mobility at the network layer. However, previous works have shown that mobility and multipath provides better performance when being plugged at the transport layer [13]. Accordingly, Paasch et al., implemented Multipath TCP (MPTCP) handover between 3G/WiFi. However, running MPTCP demands for high energy consumption to provide better throughput. Moreover, MPTCP needs to detect the WiFi failure and then reinject the lost data segments in another subflow, thereby causing a drop in throughput.

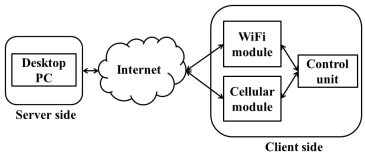


Figure 1: Simplified block diagram of the system.

Some projects [8], [9], [10] have proposed improved handover mechanisms. Most of these proposals demand for modification of current infrastructure and complex hardware. Moreover, they are not suitable for practical mobile devices. The solution proposed in HOPOVER [14] relies on buffering techniques. Buffering techniques are widely subsumed by all proposals that target at achieving seamless handovers. A new approach [15] in taking vertical handover decisions enables each mobile user to autonomously apply the handover decision policy which is more convenient to the user's specific needs.

We attempt to make our handover mechanism *simple without including any additional layer*. Our proposed mechanism only evokes transport and application layers. The mechanism *does not demand for any infrastructural change and unusual redundant hardware*. The handover decision algorithm subsumes *hybrid handover decision model* based on both networking attributes and user preferences. Nonetheless, we conduct real testbed implementation of our proposed mechanism with existing networking framework.

III. System Description And Proposed Handover Mechanism

In this section, we present the system associated with our proposed handover mechanism. We illustrate simplified block diagram and describe the operational procedure of handover mechanism in the system.

A. System Overview

Fig. 1 represents the simplified block diagram of the system corresponding to our proposed handover mechanism. This simplified system is based on ad-hoc networking mode. It uses existing Internet infrastructure for the communication. There are two sides available like classical peer to peer ad-hoc mode: (1) Server side and (2) Client side.

Vertical handover takes place at the Client side. Hence, it contains the handover decision maker and handover execution portions along with proper networking modules. The Client side comprises three fragments: (1) WiFi module, (2) Cellular module, and (3) Control unit. The WiFi module is an IEEE 802.11 based networking module. It exchanges data packets between the Internet and the Control unit. The Cellular module is a GSM telecommunication-based networking module. Task of the Cellular module is same as the WiFi module.

The Control unit governs the data communication between the client and the Internet. In case of outgoing communication, at first, it takes data (e.g., string, file, etc.) from connected

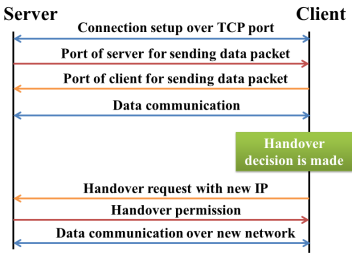


Figure 2: Proposed handover mechanism.

device (e.g., laptop) through wired medium. Then, it slices data and generates data packets. These packets are sent to the Internet through one of the two available networking modules. In case of incoming communication, the reverse operation occurs. The Control unit decides on the handover decision algorithm to select a networking module for the communication purpose. The algorithm outputs a decision on network module switching.

The Server side comprises a desktop PC which runs a server program that accepts both TCP and UDP connections. It has multiple TCP and UDP ports available. The public TCP ports are known to the Client. It communicates with the Internet via a wired medium.

B. Handover Mechanism

We propose a transparent and automated handover mechanism. This mechanism is transparent to users. Moreover, this mechanism does not involve any infrastructure that supplants the existing Internet infrastructure. The handover decision is made at the Client side. Fig. 2 presents the handover mechanism. Here, the Sever offers both TCP and UDP ports with a public IP address.

Connection setup: At first, the Client sets up a connection with the Server through a known public TCP port of the Server. The Server assigns a UDP port for a new client. Then, the Server replies with the assigned UDP port number to the Client for data communication. Upon receiving the UDP port number from the Server, the Client sends its own UDP port number to the Server for the incoming data stream. This initial stipulation is mandatory for each new server-client connection setup. The control messages, exchanged for this initial stipulation, are transferred through a TCP connection to ensure a reliable and sequential transmission.

Data exchange: The data exchange between the Server and Client takes place both over TCP and UDP connections. The Control unit decides whether to use TCP or UDP connection based on the application. For example, files are transferred over TCP and audio/video streams are transferred over UDP.

Handover decision making: Alongside, the Control unit invokes a handover decision algorithm on a regular interval. The length of this interval is tunable. If the algorithm outputs a decision of handover, the Control unit starts handover execution procedure.

Handover execution: All the control messages for handover execution are exchanged over TCP connection to ensure a reliable and sequential transmission. To distinguish between data and handover control message, a control message initiation indicator is sent before starting the control message exchange. Again, a control message termination indicator is sent after finishing control message exchange. At the beginning of handover execution procedure, Client side decides its new IP for the networking module to which the connection will be switched. Client side requests the Server side, for a network switch permission with this new IP, through the existing TCP connection of currently active networking module. Upon receiving the handover request, server checks for handover resource (e.g., TCP/UDP port) availability. Note that, the UDP communication continues through currently active networking module, in between control packet exchange period. If resource for handover is available, the Server sends handover permission with new TCP and UDP ports. Upon receiving handover permission, Client sets up a new TCP connection with the Server through new networking module. Afterwards, all UDP packets are also exchanged through new networking module. However, the Server continues to listen for packets that may have been in transit prior to the handover for a short period of time. After the beginning of successful data transmission through new networking module, the Client closes previous connections.

C. Handover Decision Algorithm

Our handover decision algorithm comprises a sequential hybrid model of user-centric (UC) strategy and multiple attribute decision (MAD) strategy [4]. At first, the algorithm invokes MAD strategy which uses well-known grey relational analysis (GRA) model [16] to rank WiFi and cellular network based on the two attributes: throughput and RSSI. The ranking of candidate networks is performed by building grey relationships with a positive ideal network. The grey relationship between the ideal network scenario and candidate network scenario can be achieved by calculating grey relational coefficient (GRC) according to the level of similarity and variability. The GRC is the score used to describe the similarity between each candidate network and the ideal network. The highest ranked network is that one which has the highest similarity to the ideal network.

An integer point ($P_{MAD,i}$, where i = WiFi or Cellular and $0 \leq P_{MAD,i} \leq 2$) is assigned to each network based on its rank. The network with higher rank gets higher point. Nonetheless, two threshold values are set for throughput and RSSI to ensure a certain level of QoS. If a network has any attribute (i.e., throughput or RSSI) value lower than the corresponding threshold value, it gets zero point.

The algorithm invokes the UC strategy. The UC strategy uses a monetary cost comparison policy to select a network and assigns an integer point ($P_{UC,i}$, where i = WiFi or Cellular and $1 \leq P_{MAD,i} \leq 2$) to each network. The network with higher usage cost gets lower point.

$P_{MAD,i}$ represents the QoS and $P_{UC,i}$ represents the mon-

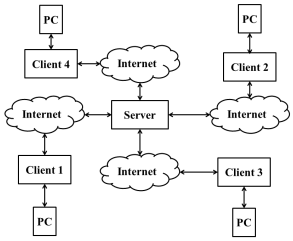


Figure 3: Experimental setup.

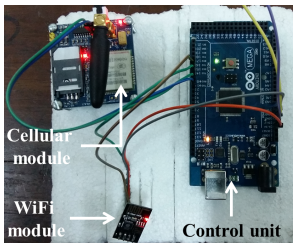


Figure 4: The components of the Client module.

etary cost optimality of i^{th} network. To combine these two attributes, we use the simple additive weighting (SAW) model as follows:

$$P_{combine,i} = w \times P_{MAD,i} + (1 - w) \times P_{UC,i} \qquad (1)$$

where w, $0 \leq w \leq 1$, is the weight assigned for QoS attribute. $P_{combine,i}$ is the total point assigned to the i^{th} network combining two attributes.

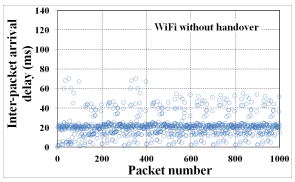
The network with higher $P_{combine,i}$ is selected, which resolves the trade-off between QoS and monetary cost optimality of network usage. User may push his/her preference for tuning the value of w.

IV. Experimental Evaluation

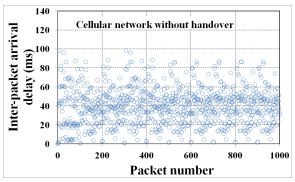
We conducted a set of testbed experiments with our proposed handover mechanism to evaluate its performance in real implementation scenario. In this section, we present the experimental setup and then analyze the experimental results.

A. Experimental Setup

We have used a chat group consisting of four clients and one central server. A client can transmit file and data stream to another connected client. The assembly of client and server is identical to the system described in Section III. Fig. 3 presents the experimental setup. Each client is connected to a PC through a USB port. The PC provides data (i.e., string, file, binary stream) along with identity of the destination client. All the clients initially set up connection with the server following the similar procedure mentioned in Section III. The Server stores the IP address of connected clients. If a client wants to chat or share files with another client, it sends a request to that client through the server. If the second client accepts the request, the server updates the connection status in a connection table. The clients are not allowed to have connections with more than one client at a time. If a third client tries to chat with any of two connected clients, the server sends a busy message. However, two pairs of clients can distinctly communicate in parallel. Any data packet received from a client registered in the connection table is forwarded to the connected client. If a client sends a connection termination control message to server, the server deletes the connection entry from the table and notifies the other client. For the result analysis purpose, the client module stores the log of networking metrics in the corresponding PC.

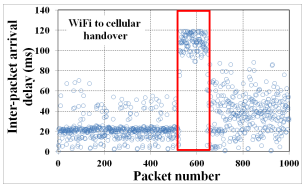


(a) WiFi network

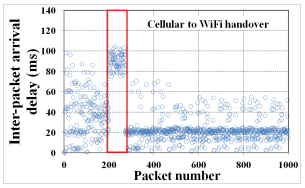


(b) Cellular network

Figure 5: Inter-packet arrival delay in handover-less scenarios.

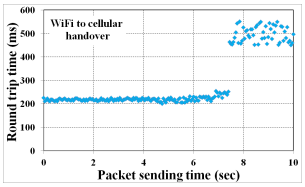


(a) WiFi to cellular handover

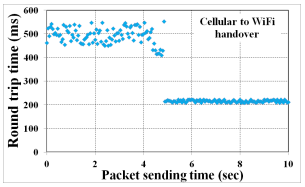


(b) Cellular to WiFi handover

Figure 6: Inter-packet arrival delay in handover-held scenarios.



(a) WiFi to cellular handover



(b) Cellular to WiFi handover

Figure 7: Effect of handover on round trip time.

Fig. 4 shows the client module in our experimental setup. We use ESP8266 [17] as the WiFi module, which follows the IEEE 802.11 protocol. SIM900A [18] is adopted as the cellular module. An Arduino mega board [19] is used as the Control unit. This Arduino mega board is connected to the PC through the USB COM port. Note that similar type of modules are also used in cellular mobile devices (e.g., smartphone, GSM activated tab). Consequently, our proposed mechanism can be easily adopted by any cellular mobile device.

B. Experimental Results and Analysis

We conduct our experiments for both WiFi to cellular and cellular to WiFi handovers. We also perform experiments for both TCP file transfer and UDP stream transfer. We conduct our experiments under four possible scenarios: (1) WiFi connection only, (2) Cellular connection only, (3) WiFi to cellular handover, and (4) Cellular to WiFi handover. Each experiment is performed for all three possible combinations of inter-client connections. We present the results of a particular experiment by taking average over the results from all possible combinations of inter-client connections.

1) Effect on Inter-packet Arrival Delay: First, we measure the inter-packet arrival delay of a UDP data stream sent from one client to another. Two consecutive packets are separated by 15ms interval. We present the avg. result for 1000 packets.

Fig. 5a shows the inter-packet arrival delay values when packets are sent through the WiFi connection only. The values

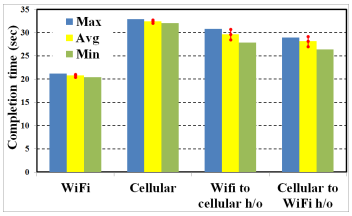


Figure 8: Comparison of avg. file transfer completion time.

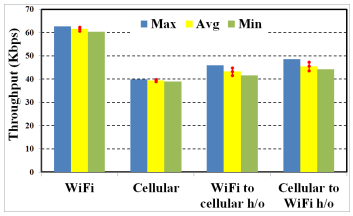


Figure 9: Comparison of avg. throughput.

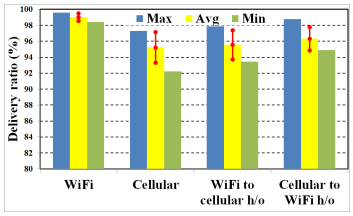


Figure 10: Comparison of avg. delivery ratio.

of inter-packet arrival delays are clustered between 19ms to 24ms. The maximum inter-packet arrival delay is lower than 80ms. Fig. 5b shows the inter-packet arrival delay values when packets are sent through the cellular network only. The values have a significant variance and ranges up to 100ms.

To initiate a WiFi to cellular handover, we move the client node and connected laptop far from the WiFi router at the time of transmission. Fig. 6a shows the inter-packet arrival delays when a WiFi to cellular handover takes place. As shown in the figure, there is a region between packet number ∼520 and ∼650 with incongruous and higher inter-packets arrival delays clustered at ∼98ms to ∼120ms. This one is the handover region. The values of inter-packet arrival delays for UDP data packets get higher due to TCP handover-control packet exchange between server and client. Note that, though the inter-packet arrival delays get higher, the connection is not broken. This indicates seamless or soft handover. The region before the handover region in Fig. 6a is similar to Fig. 5a and the region next to the handover region is similar to Fig. 5b.

To initiate a cellular to WiFi handover, we move the client node and connected laptop closer to the WiFi router at the time of transmission. Fig. 6b shows the inter-packet arrival delays when a cellular to WiFi handover takes place. As shown in the figure, there is a region between packet number ∼200 and ∼270 with incongruous and higher inter-packet arrival delays clustered at ∼80ms to ∼100ms. Here, this one is the handover region. The region before handover region in Fig. 6b is similar to the Fig. 5b and the region next to the handover region is similar to the Fig. 5a.

Here, the notable fact is both inter-packet arrival delay and number of packets in handover region are higher in case of WiFi to cellular handover than that of cellular to WiFi handover. This happens so when a handover takes place from a low latency high capacity network (WiFi) to a high latency limited capacity network (cellular), the inter-packet arrival delay increases. For the same reason, the number of packets in WiFi to cellular handover region is relatively higher.

2) Effect on Round Trip Time: RTT is measured by the difference in time between the client sending a packet and receiving the corresponding acknowledgement packet from the server. In case of lost (data or acknowledgement) packets, the RTT takes time difference between successful reception time of Acknowledgement at client side and first time sent time of the data packet.

Fig. 7a shows the RTT between client and server measured at the client side for WiFi to cellular handover and Fig. 7b shows RTT for the other handover scenario. Both figures suggest that the values of RTT remain stable during the handover events. The stability of RTT implies that packet loss during handover is minimum.

3) TCP File Transfer: In this experiment, we send data packets over TCP connection by transfering a 160kB file from one client to another client for 100 times. The receiver client PC keeps the log of file transfer completion times. The TCP data packets are separated by 15ms.

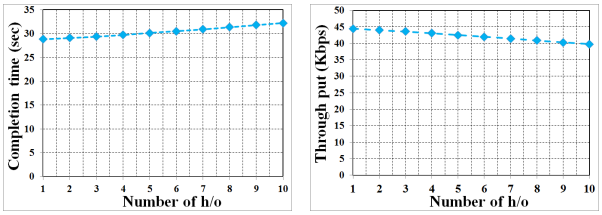
Fig. 8 shows the comparison of average file transfer completion times of all four scenarios. Note that both of the handover scenarios present values for a corresponding single handover event. In case of handover-less transmission, cellular network takes significantly more time than WiFi for sending the file, as cellular network is a high latency limited resource network compared to WiFi. On the other hand, in case of handover-held transmission, the completion time deviates a little from the handover-less transmissions. Nonetheless, in some cases, both of the handover-held transmissions perform better than the handover-less cellular transmission. Note that no corrupted file transmission is reported during handover-held transmissions.

4) Effect on Throughput: In this experiment, we send 100 data packets per second from one client to another client. Here, the data stream is transferred for ten times and the sender client keeps log of the throughput.

Fig. 9 shows a comparison in average throughput for all four scenarios. Here, both of the handover scenarios present values for a corresponding single handover event as before. In case of handover-less transmissions, low latency high capacity WiFi network offers significantly higher throughput compared to high latency low resource cellular network. In case of handover-held transmissions, throughput deviates a little from handover-less transmissions. Thus, we can extrapolate that, the handover control message exchange influences overall throughput in a very small scale.

5) Effect on Delivery Ratio: Here, a stream of 5000 evenly apart UDP packets are transmitted from one client to another client in aforementioned four scenarios. Alongside, UDP packets are evenly apart by 15ms. The receiver client keeps the track of total number of the received packets.

Fig. 10 shows a comparison in delivery ratios for all four scenarios. From the figure, we may conclude that the delivery ratio is merely influenced by the handover event. In some cases, delivery ratio of handover-held transmissions are found



(a) File transfer completion (b) Throughput

Figure 11: Effect of the number of handovers.

higher than the handover-less cellular transmission.

6) Effect of Number of Handovers: Up to this point, we consider a single handover event at the time of transmission. Now, we intend to study how the number of handovers in the middle of a transmission influences the networking metrics. To do so, we move a client module back and forth for a reasonable distance with respect to a WiFi router. Here, this procedure introduces a combination of two different types of handovers.

First, we study the effect of the number of handovers on TCP file transfer completion like above mentioned file transfer experiment. Fig. 11a shows file transfer completion time for different number of handovers. We observe a slight increment in file transfer completion time with the increment of number of handovers. Each additional handover event increases file transfer completion time by 1.23% on average. This happens so each handover event needs a certain time to exchange handover control message. The higher completion time requirement for transmission through cellular network also influences the increment of completion time. We observe the intuitive opposite scenario in case of throughput. Fig. 11b shows the effect of number of handovers on throughput. Here, throughput decreases in a small scale with the increasing number of handovers. Each additional handover event decreases throughput by 0.012% on average.

V. Future Work

We conduct our experimentation in a specific scenario consisting of only four clients with a constant load on the network. However, in real world, we can hardly expect such a scenario with a constant load on the network. To inspect the robustness of our proposed handover mechanism against varying load on the network, in future, we will implement our handover mechanism with a large number of clients.

VI. Conclusion

Vertical handover methodology can play an important role for seamless heterogeneous wireless access. We have proposed a transparent, seamless, easy-to-implement vertical handover mechanism between WiFi and cellular networks. Our proposed mechanism subsumes a hybrid decision model. We have conducted experiments on a testbed to validate our proposed method which does not require change in the Internet infrastructure. According to experimental results, a handover-held communication exploiting our handover mechanism has almost similar throughput, RTT, delivery ratio, inter-packet arrival delay compared to handover-less situation.

References

[1] O. Ormond, G.-M. Muntean, and J. Murphy, "Network selection strategy in heterogeneous wireless networks," in *Information Technology and Telecommunications Conference (IT&T)*, 2005.

[2] S. Balasubramaniam and J. Indulska, "Vertical handover supporting pervasive computing in future wireless networks," *Computer Communications*, vol. 27, no. 8, pp. 708–719, 2004.

[3] X. Yan, Y. A. Şekercioğlu, and S. Narayanan, "A survey of vertical handover decision algorithms in fourth generation heterogeneous wireless networks," *Computer Networks*, vol. 54, no. 11, pp. 1848–1863, 2010.

[4] M. Kassar, B. Kervella, and G. Pujolle, "An overview of vertical handover decision strategies in heterogeneous wireless networks," *Computer Communications*, vol. 31, no. 10, pp. 2607–2620, 2008.

[5] A. Ahmed, L. M. Boulahia, and D. Gaiti, "Enabling vertical handover decisions in heterogeneous wireless networks: A state-of-the-art and a classification," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 776–811, 2014.

[6] V. A. Narayanan, A. Rajeswari, and V. Sureshkumar, "An intelligent vertical handover decision algorithm for wireless heterogeneous networks," *American Journal of Applied Sciences*, vol. 11, no. 5, p. 732, 2014.

[7] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure, "Exploring mobile/wifi handover with multipath tcp," in *ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, 2012, pp. 31–36.

[8] A. Pandolfi, A. Surtees, R. Finking, and S. McCann, "Optimising simultaneous interface usage in a converged multimode terminal," in *Proc. of the 14th IST Mobile & Wireless Communications Summit*, 2005.

[9] B. Hongyan, H. Chen, and J. Lingge, "Intelligent signal processing of mobility management for heterogeneous networks," in *International Conference on Neural Networks and Signal Processing*. IEEE, 2003, pp. 1578–1581.

[10] P. M. Chan, R. E. Sheriff, Y.-F. Hu, P. Conforto, and C. Tocci, "Mobility management incorporating fuzzy logic for heterogeneous a ip environment," *IEEE Communications Magazine*, vol. 39, no. 12, pp. 42–51, 2001.

[11] L. Oliva, A. De, A. Banchs, I. Soto, T. Melia, and A. Vidal, "An overview of ieee 802.21: media-independent handover services," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 96–103, 2008.

[12] C. Perkins, D. Johnson, and J. Arkko, "Mobility support in IPv6," IETF RFC 6275, Jul 2011.

[13] D. Wischik, M. Handley, and M. B. Braun, "The resource pooling principle," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 5, pp. 47–52, 2008.

[14] F. Du, L. M. Ni, and A.-H. Esfahanian, "Hopover: a new handoff protocol for overlay networks," in *IEEE ICC*, 2002, pp. 3234–3239.

[15] A. Calvagna and G. Di Modica, "A user-centric analysis of vertical handovers," in *2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, 2004, pp. 137–146.

[16] Q. Song and A. Jamalipour, "A network selection mechanism for next generation networks," in *IEEE ICC*, 2005, pp. 1418–1422.

[17] N. Space, "Esp8266," 2015, accessed on April 7, 2016. [Online]. Available: http://nurdspace.nl/ESP8266

[18] SIMCOM, "Sim900a," 2015, accessed on April 7, 2016. [Online]. Available: http://www.simcom.ee/modules/gsm-gprs/sim900/

[19] Arduino, "Arduino mega 2560," 2016, accessed on April 7, 2016. [Online]. Available: http://www.arduino.cc/en/Main/ArduinoBoardMega2560