A Cross-Layer Architecture for Service Continuity and Multipath Transmission in Heterogeneous Wireless Networks

Min-Cheng Chan, Chien-Chao Tseng
Dept. Computer Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

Li-Hsing Yen
Dept. Computer Science and Information Engineering
National University of Kaohsiung
Kaohsiung, Taiwan, R.O.C.

***Abstract*—Mobile devices equipped with two or more interfaces can roam across heterogeneous networks. This feature enables the devices to maintain service continuity by performing vertical handovers and to increase transmission capability by exploiting multiple physical transmission paths. However, this type of service continuity is not yet commonly provided. Previous approaches toward the provision of service continuity are limited either in their functionalities or in the scope of cross-layer information. In this paper, we analyze how cross-layer architecture provides service continuity from a broader perspective and propose Cooperative Layered Architecture (CoLA) implemented on Android system for service continuity during handover process. CoLA helps comprehensive information gathering from all layers for handover decisions, provides simple interfaces to make applications mobility-sensitive, and achieves efficient multipath transmissions across heterogeneous wireless links. Experimental results show that CoLA achieves its design goal in providing service continuity.**

***Index Terms*—Service continuity, multipath transmission, heterogeneous wireless networks, cross-layer architecture.**

I. Introduction

It is common for modern mobile devices to equip with two or more network interfaces. This feature allows devices to roam among heterogeneous wireless networks, which provides two benefits to the devices. First, these devices are able to maintain service continuity by performing vertical handovers. Second, these devices benefit from augmented bandwidth by exploiting multiple transmission paths. However, the ability to perform vertical handover and exploit multiple transmission paths is not yet a built-in functionality of modern mobile devices. Several issues related to handovers and multipath transmissions remain to be resolved.

Handover refers to the process of switching an access link from one base station or access point to another. It involves a sequence of activities of several layers in the TCP/IP protocol stack. In a modular yet traditional design, each layer independently conducts handovers without coordination with other layers. This design has been identified inefficient, calling for a cross-layer design [1-6] where handovers are conducted with close cooperation/coordination among multiple layers in TCP/IP stack. The cross-layer design also helps achieve efficient *vertical handovers* [8], where a vertical handover involves a change of the device's access link between two heterogeneous wireless systems.

Previous approaches to cross-layer handovers [1-6] have some limitations. A central part of these approaches is to determine at which time and to which target a handover should be performed. These approaches either make handover decisions with statuses or events inside a single layer, or attempt to improve the performance of a particular layer based on the information from another layer. It is thus common that handover decisions are made with incomprehensive information. To achieve a better result, handover decisions should be made with information from all layers that are involved in handovers. Another limitation with existing cross-layer approaches is that these approaches seldom, if ever, consider vertical handovers or concurrent transmission with multiple interfaces.

Using SCTP or similar mechanism to create a session that transmits data through two or more transport channels concurrently can potentially increase application-layer throughput. However, when these transport-layer channels are built upon different access links, extra efforts should be done to deal with problems caused by the heterogeneity of these links. For example, different links may provide different transmission capability and experience different levels of packet loss rates. It is challenging to distribute application data among transport connections and achieve a balanced load among all connections in this environment. Furthermore, when a handover occurs to one of these links, it is also required to maintain the whole session despite the change of this link. This task is termed session continuity. Without session continuity, applications need to re-establish all ongoing connections after a change of network attachment, causing unnecessary protocol overhead and extra delays.

In this paper, we propose Cooperative Layered Architecture (CoLA) to provide session continuity and thus service continuity in heterogeneous wireless networking environment. CoLA consists of a Cross-layer Cooperative Module, a Handover Decision Module and a Session Continuity Module. The Cross-layer Cooperative Module extends IEEE 802.21 media independent handover function [7] to provide command, information and event service among all TCP/IP layers. The Handover Decision Module makes handover decisions based on information obtained from Cross-layer Cooperative Module, achieving shorter handover latency. The Session Continuity Module maintains session continuity on occurrences of handovers. Moreover, it balances the loads among available paths. The latter two modules are designed as session-layer entities. With the assistance of Cross-layer Cooperative Module, they can gather information or events from all TCP/IP layers and make handover or load balance decisions according to the requirement specified by applications.

CoLA has been implemented on Android. As a middleware between the kernel and user applications, CoLA makes applications mobility-sensitive and, on the other hand, achieves application-driven interface usage. We conducted experiments to evaluate the performance of CoLA in dealing with vertical handovers. The obtained results confirm that CoLA achieves its design goal in providing better service continuity.

The rest of this paper is arranged as follows. The next section briefs handover procedure and summarizes prior work to exhibit the need for cross-layer design in handovers. We then detail the proposed design, CoLA, in Section III and describe two typical handover scenarios in Section IV. In Section V, we detail the implementation of CoLA on Android. The evaluation results of CoLA are shown in Section VI. Finally, Section VII concludes this paper.

II. Related works

Figure 1 illustrates the sequence of conventional handover procedure [1]. Handover begins with Link Layer Move Detection, in which a mobile node (MN) detects the need to switch from the current base station (BS) it associates with to a new BS. MN handovers to the target BS in Link Layer Handover phase, triggering link-down event and nullifying address configurations. Network Layer Move Detection detects the nullification of address and triggers Network Layer Handover to obtain a new address configuration. Network Change Detection discovers network changes and invokes the Upper Layer Reaction.

We mark that handover procedure is complex and relies on cooperation between those subroutines mentioned above. There have been some proposals for handover enhancement with the help of cross-layer information.

A. Cooperation between link layer and network layer for better handover performance

IEEE 802.21 [7] has been proposed to achieve seamless handover and has been widely used in handover operation among heterogeneous wireless networks [8]. IEEE 802.21 defines Media Independent Handover (MIH) standard to provide cross-layer command, information and event services for the optimization of handover procedures across heterogeneous access links. The command service enables MIH users to manage and control link behaviors relevant to handovers and mobility. The information service provides details on the properties and services provided by the serving and neighboring networks. The event service provides event classification, event filtering and event reporting corresponding to dynamic changes in link characteristics, link status, and link quality. However, services specified by IEEE 802.21 are confined to link and network layers.

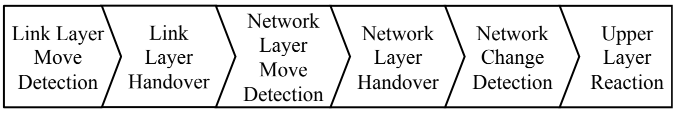

Fig. 1. Sequence of conventional handover procedure.

Yen et al. [1] showed that existing network drivers might suffer from long handover delays for mismatching settings between extended service set (ESS) and subnet (i.e., intra-ESS/inter-subnet and inter-ESS/intra-subnet handover). The solution proposed in [10] eliminates the problem by informing network layer of link layer information. Fast Mobile IPv6 [2] exploits cross-layer information to predict an imminent handover and accordingly sets up a tunnel to the subsequent access router in advance. Address configuration and duplicated address detection (DAD) can also take place before the handover, which reduces network-layer handover delay. Tseng et al. [3] demonstrated network layer pre-handover with the help of link layer information, which significantly reduced total handover latency.

B. Cooperation between link layer and transport layer for better transmission performance

Research mentioned above enhanced handover performance by cross-layer cooperation between link and network layer. However, some transport-layer parameters (e.g., the size of congestion windows) also closely relates to link status. Exploiting link-layer information can enhance transport-layer performance. Chang et al. [4] increased TCP throughput by a cross-layer design that predicts the bandwidth of target network. Without cross-layer information, TCP suffers from inaccurate congestion window prediction when roaming among heterogeneous networks with different bandwidth capacities.

Iyengar et al. [5] pointed out side effects of using SCTP for concurrent multipath transmission (i.e., transmitting data through two or more transport channels concurrently) due to unawareness of other transport channels. The author introduced a new retransmission mechanism that reacts to the packet loss rates of all transmission paths.

The use of concurrent multipath transmission requires session continuity, which handles transient disconnection and maintains ongoing sessions when MN changes its transmission paths. SCTP provides session continuity by configuring multiple secondary addresses. Multipath TCP (MPTCP) [9] not only provides session continuity but also merges several transmission paths into a single pooled resource, providing more flexible load balance mechanisms among these paths. However, SCTP and MPTCP are not suitable for multimedia transmission.

C. Session continuity mechanisms above transport layer

The provision of concurrent multipath transmissions potentially increases transmission capability. To enable the use of any transport-layer protocol for concurrent multipath transmission, the administration function of the multipath transmission should be higher than the transport layer. To provide session continuity above the transport layer, Issa et al. [6] proposed a platform that supports session continuity by adding a session layer abstraction. Senders encapsulate every outgoing packet with a session header. The header includes a session ID that uniquely identifies the session. Receivers consider packets with the same session ID (instead of IP address) belonging to the same session. Furthermore, this platform enables applications to adjust their parameters in a real-time manner based on network status fed back from network interface.

Despite the variety of these proposals, some common properties associated with prior work are summarized as below:

- Incomprehensive information: only the behavior of one specific layer is of concern when exploiting cross-layer information. Little has been done on a modular design with a systematic analysis, which is crucial to the realization of service continuity.
- Inefficient multipath transmission: cross-layer information is not fully utilized to facilitate concurrent multipath transmission.

Therefore, we propose CoLA, a modularly designed with the objective to improve handover performance, enhance concurrent multipath transmission performance and thus provide better service continuity. We have implemented CoLA on Android and used it as a prototype for performance evaluation.

III. Proposed Architecture

Figure 2 shows the architecture of CoLA, which includes three main modules: Cross-layer Cooperative Module, Handover Decision Module and Session Continuity Module. There are also three sub-modules acting as agents for each layer: Link Manager, Network Manager, and Transport Manager. An application programming interface (API) is provided for an easy access of CoLA features by applications. The operation details of each component are described below:

A. Cross-layer Cooperative Module

Previous research confirms the need of a cross-layer communication mechanism for effective handover and transmission performance. Cross-layer Cooperative Module provides cross-layer command/information/event services as specified by IEEE 802.21. However, IEEE 802.21 defines only the command/information/event services between link and network layer. CoLA additionally supports these services in transport and application layers and thus enables more comprehensive information and event gathering.

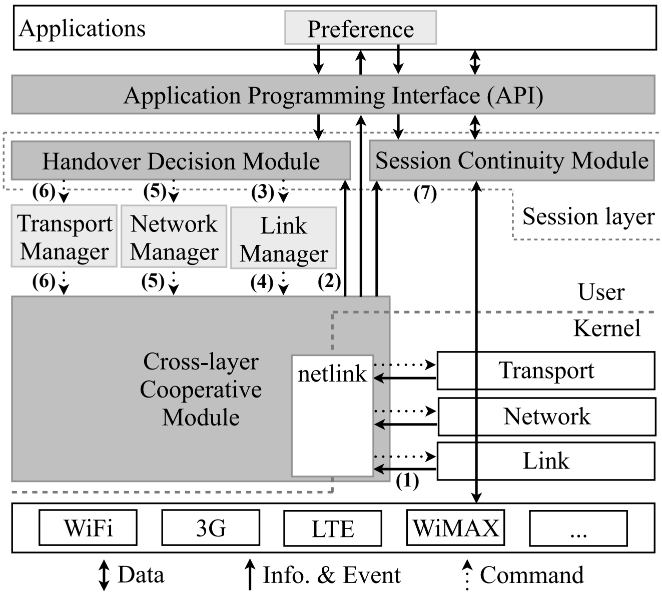

Fig. 2. System architecture of CoLA.

B. Handover Decision Module

With the consideration of application requirements, Handover Decision Module makes decision based on the cross-layer information and events gathered by Cross-layer Cooperative Module. For instance, Handover Decision Module could monitor the Received Signal Strength Indication (RSSI) reading and trigger a link layer handover in advance if the RSSI value drops below an application-specific threshold. Handover Decision Module decides the handover target based on neighbor information obtained from Cross-layer Cooperative Module.

C. Session Continuity Module

Session Continuity Module handles transient disconnections and maintains ongoing sessions when MN changes its transmission paths. At the sender side, all packets examined by Session Continuity Module are encapsulated with a session header containing session ID and session sequence number. Session ID is used to uniquely identify each session. At the receiver side, data of the same session are reordered by their session sequence numbers and then delivered to applications. In addition, Session Continuity Module manages data buffers. It merges and reorders data from different transport channels when using concurrent multipath transmissions.

For concurrent multipath transmissions, Session Continuity Module also takes care of resource aggregation and load balance among several transport channels according to application-specific transmission policies. CoLA currently implements four policies for application's resource demand: minimum latency, maximum bandwidth, power saving, and traffic shaping. The minimum latency and maximum bandwidth are trivial. The power saving policy activates only one interface to conserve energy. The traffic shaping policy limits the buffer size and thus the bandwidth of a specific path. This prevents the application from overwhelming all resource of the path. Furthermore, since the minimum latency policy selects the path with the lowest latency but does not guarantee a low jitter (i.e., variation of latency) of the selected path, the minimum latency policy is not suitable for multimedia transmission. The traffic shaping policy attempts to reduce the jitter by confining the outgoing traffic of the transmission path. Details of these policies will be described in Section V.

Session Continuity Module must coordinate all transport channels to achieve effective concurrent multipath transmission, so it should be located above the transport layer. With additional consideration of application friendliness, we put Session Continuity Module in the session layer as a user space interface such that applications could be mobility-sensitive without handling mobility issues.

D. Application Programming Interface (API)

API is an interface for application to access command/information/event services provided by CoLA. For example, API allows applications to register a specific event with a callback function, which will be invoked when Cross-layer Cooperative Module captures the event. API also enables applications to set up QoS preference, which affects the behavior of Handover Decision Module and Session Continuity Module.

E. Link/Network/Transport Manager

Link Manager, Network Manager and Transport Manager play the role of agent for the respective layer. They carry out the commands issued by Handover Decision Module. Handover Decision Module may instruct Link Manager to perform a full channel scan to refresh neighbor information. Network Manager is designed for network layer continuity and the management of IP configurations. Transport Manager maintains transport layer connections (e.g., TCP, UDP, and SCTP). Particularly, it re-establishes transport-layer connections after both link- and network-layer handovers are completed.

In the next section, we give two use cases to demonstrate how these modules cooperate with each other to achieve the objective of effective handover and transmission performance.

IV. Operation Scenarios

A. Network-triggered handover

Network triggered handovers are those reactively triggered by network-side events such as link down, address change, network congestion, etc. We use a simple network topology as shown in Fig. 3 to show how CoLA handles network-triggered handovers. A cellular base station BS covers the whole area, while one WiFi access point AP covers a smaller area. BS and AP connect to correspondent node (CN) through transmission path P1 and P2, respectively. Fig. 2 shows how CoLA reacts to network condition changes. When the MN moves away from its original cell served by AP to the area exclusively covered by BS, Link layer triggers an RSSI_CHANGE event (1). Cross-layer Cooperative Module will notify Handover Decision Module of the RSSI_CHANGED event (2). Handover Decision Module will then start the interface selection algorithm and command Link Manager to activate a secondary interface (3) when the RSSI value drops below an application-specific threshold (a soft threshold). If the RSSI value keeps decreasing, Link Manager will execute a link layer handover (4) from P1 to P2 as soon as the RSSI value drops below the hard threshold. After Link Manager finishes the link layer handover, Handover Decision Module will instruct Network Manager to discover needed network layer information (5) such as subnet number, which helps CoLA avoid the ESS-subnet mismatching problem [1]. Transport Manager will re-establish all transport layer connections (6) using the new path P2. Finally, Session Continuity Module will resume previous sessions (7).

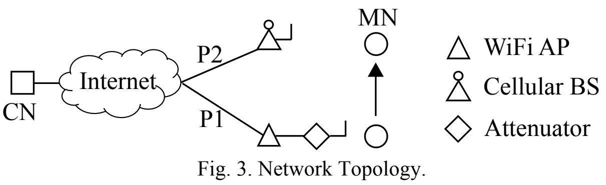

Fig. 3. Network Topology.

B. Application-initiated handover

Application-initiated handover is proactively requested by local applications. For example, a P2P videoconference application might change its QoS requirement and load balance policy when a large number of users join the group. The change of application preference setting eventually forces Handover Decision Module to choose a better network environment and thus invokes a handover procedure proactively. Session Continuity Module then distributes application data among new available paths according to application-specific transmission policies, which are described in the previous section.

V. Implementation

We implemented CoLA on Android-based mobile devices. All CoLA modules were implemented as Android applications with one slight modification on Android framework. This modification enables simultaneous activations of multiple network interfaces in Android 2.3 and later versions, which is not allowed by default.

A. Cross-layer Cooperative Module

1) Command/Information Service

Android built-in classes WifiManager, TelephonyManager and ConnectivityMananger provide link layer command/information service such as scan, connect and disconnect. However, Android framework does not support network and transport layer commands/information service. For network layer commands/information service such as routing table configuration, we use netlink/rtnetlink, which is a Linux built-in mechanism for the communication between kernel space and user space. For transport layer command/information service, we use *setsockopt*/*getsockopt*, which are Linux built-in functions that allow applications to configure/obtain socket options such as buffer size and window size.

2) Event Service

To implement event registration and notification scheme, we used Android Interface Definition Language (AIDL) for inter-process communication (IPC). AIDL enables the definition of IPC interface between service provider (e.g., Cross-layer Cooperative Module) and service consumer (e.g., Handover Decision Module, Session Continuity Module and application). Service consumers first define and register a callback function that handles and reacts to a specific event. When that event is signaled, Cross-layer Cooperative Module then invokes the corresponding callback function.

B. Handover Decision Module

We implemented a preliminary version of a pre-selection algorithm that registers and monitors the RSSI_CHANGE event. Instead of waiting for link-down event, the pre-handover procedure will take place when the RSSI of current link drops below a specific threshold.

C. Session Continuity Module

Session Continuity Module is responsible for session continuity and load balance among transmission paths. To provide session continuity, it encapsulates each outgoing packet at the sender side with a session header, in which we add a unique session ID for the identification of different sessions. In addition, a sequence number is added to session header in order to reorder packets at the receiver side.

Session Continuity Module also decides through which path an outgoing packet should be sent. According to application-specific QoS policy and link layer statuses, Session Continuity Module dynamically calculates the priority of each link for data transmissions. The four QoS policies currently supported by CoLA are as follows.

1) Minimum latency

In this case, the link priority is determined by transmission latency, which consists of transmission delay, propagation delay and queuing delay. CoLA calls *ping* periodically to measure the latency of a specific transmission path and selects the path with the lowest latency as the main transmission path.

2) Maximum bandwidth

All available paths are activated to achieve maximum available bandwidth. To reduce latency in this case, the transmission priority of each path is based on transmission queue size. Smaller transmission queue size indicates fewer pending data in the buffer of the transmission path and thus a higher priority when Session Continuity Module dispatches arriving data to a transmission path.

3) Power saving

In this case, only one interface (i.e., that with the highest RSSI value) is selected to transmit data. Data will be dispatched to one of the secondary interfaces if and only if the RSSI reading of the current interface drops below the RSSI reading of the interface.

4) Traffic Shaping

Session Continuity Module blocks data from writing to the transport layer when outgoing traffic volume exceeds a threshold.

If application additionally requires a reliable transmission, packets will be put into a queue for possible retransmissions. The contents of the queue are managed in accordance with the congestion window of underlying TCP. Packets are removed from the retransmission queue as TCP receives the corresponding acknowledgements. If a TCP connection is closed unexpectedly due to handover, all data in the retransmission queue will be retransmitted when a new transport connection becomes available.

D. Sub-modules and API

Link/Network/Transport Manager receives handover instructions from Handover Decision Module and performs corresponding actions such as scan, address acquirement or TCP connection setup. The Cross-layer Cooperative Module then executes these actions as mentioned above.

CoLA is implemented as an Android service. To use CoLA API, application developers could simply bind the service by Android *bindService()* function and access API through method invocation.

VI. EVALUATION

Two experiments were performed to evaluate the proposed approach, which are described as follows.

A. Handover latency

To know how CoLA helps shorten handover latency, we built a network topology as shown in Fig. 3. We let the MN send out one UDP datagram every 20 ms and collected packet loss and latency data at the CN side. The AP was connected to an attenuator that controls the transmit power of AP to simulate path loss due to movements of MN. Figure 4 shows the result that was obtained without the use of CoLA. The result confirms that conventional Android devices suffer from long handover delay. From sequence number 323 to 2796, the received signal strength at the MN was too weak for a successful delivery of UDP datagram; however, the MN did not initialize a handover procedure since the link was still considered connected. Figure 5 shows the result that was obtained with the use of CoLA. Although there were still some packet losses when the signal got weak, the packet loss gap (sequence number 1028-1046) is significantly reduced. With these results, we conclude that CoLA effectively improves handover performance in terms of packet loss rate.

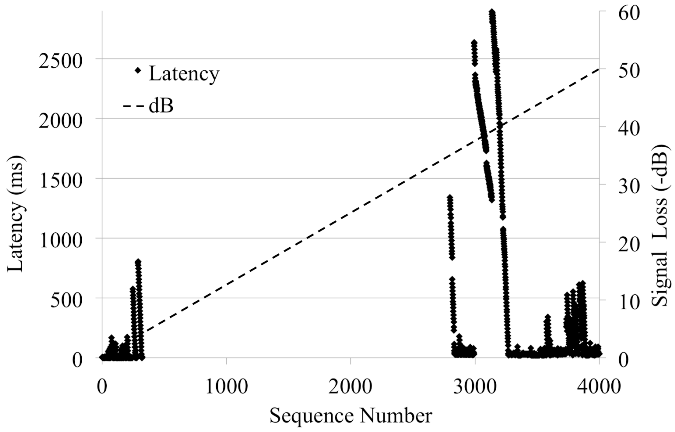

Fig. 4. Handover latency without CoLA.

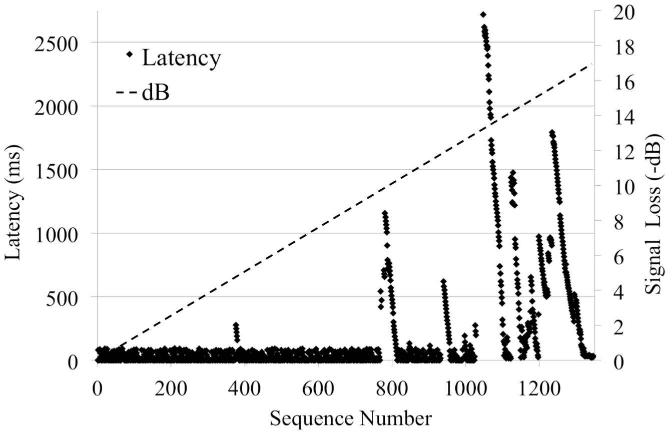

Fig. 5. Handover latency with CoLA.

B. Application-driven path selection

To demonstrate that CoLA effectively uses multipath resource according to application requirement, we deployed the MN in a location where both WiFi and 3G were available. Table I shows the evaluation parameters. The MN transmitted a 30 MB file to the CN. Table II and III show the transmission latency and bandwidth, respectively, measured at the CN. The latency mentioned in this experiment refers to the time period between the sender writing data into its transport layer and the receiver reading data from its transport layer.

1) Minimum latency and Power saving

In these cases, WiFi was selected since it provides lower latency with lower power consumption when radio signals are stable.

2) Maximum bandwidth

WiFi and 3G were both activated to provide maximum bandwidth. A large amount of reorderings was observed at the receiver side due to the heterogeneity (especially latency) of these links. The reordering degraded overall performance, making it even worse than using a single link. With our scheduling method mentioned in Section V, the measured bandwidth approximates to the sum of each link. However, the latency is still much longer than the total value due to reordering. To reduce the overhead caused by reordering, a dynamic scheduling algorithm is needed with consideration of latency, bandwidth and queuing condition of lower layer. We leave this as a future work.

3) Traffic Shaping

The bandwidth was limited to 500 KB/s. The latency was much lower than that in the first case since the queuing delay was reduced. Furthermore, the jitter was also lower than those in other cases.

The experimental results confirm that CoLA is able to use multipath resource according to application requirement.

VII. CONCLUSION AND FUTURE WORKS

Service continuity relies on handover decision and session management. As there is no systematic design to integrate current cross-layer handover mechanisms, we propose cooperative layered architecture (CoLA) that demonstrates a modular design for the provision of service continuity in handovers. The advantages of CoLA are summarized below:

- Comprehensive information or event gathering, which shortens handover latency and thus improves service continuity.
- Interface usage could be application-driven and thus provides better path quality.
- Application could be mobility-sensitive, which can react to the change of network condition.
- Effective multiple path control, which provides better transmission performance.

TABLE I. EVALUATION PARAMETERS

	WiFi	3G
Maximum upload Bandwidth	100 Mbps	512 Kbps
Average Round-trip Time	8 ms	240 ms

TABLE II. TRANSMISSION LATENCY

Latency (ms)	MIN_LAT/PS	MAX_BW	SHAPING
Average	55.22	1899.15	4.38
Standard Deviation	14.81	742.22	5.36
Jitter	17.21	85.92	1.81

TABLE III. TRANSMISSION BANDWIDTH

Bandwidth (Kbytes/s)	MIN_LAT/PS	MAX_BW	SHAPING
Average	1097.62	1134.23	468.06
Standard Deviation	81.72	101.01	81.54

We implement CoLA on Android and carry out two evaluations to verify the performance of our design. The result shows that CoLA improves handover latency as well as providing better transmission performance when using concurrent multipath transmission.

Future work will be the provision of more device information such as velocity and location to achieve better handover performance. Multipath scheduling algorithm will also be enhanced to minimize the number of out-of-order packets. In addition, more evaluation will be done to show how CoLA behaves under various applications (e.g. video streaming and VoIP) and how CoLA consumes CPU and memory resources.

ACKNOWLEDGEMENT

This work was sponsored in part by National Science Council, Taiwan, under grant numbers NSC 101-2221-E-009-031-MY3 and NSC 100-2221-E-009-072-MY3.

REFERENCES

[1] L.H. Yen, H.H. Chang, S.L. Tsao, C.C. Hung, and C.C. Tseng, "Experimental Study of Mismatching ESS-Subnet Handoffs on IP over IEEE 802.11 WLANs," *International Conference on Wireless and Optical Communications Networks* (WOCN), 2011.

[2] "Mobile IPv6 Fast Handovers," RFC 5568, Jul. 2009.

[3] C.C. Tseng, L.H. Yen, H.H. Chang, and K.C. Hsu, "Topology-Aided Cross-Layer Fast Handoff Designs for IEEE 802.11/Mobile IP Environments," *IEEE Communications Magazine*, Dec. 2005.

[4] B.J. Chang and J.F. Chen, "Cross-Layer-Based Adaptive Vertical Handoff With Predictive RSS in Heterogeneous Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 6, Nov. 2008.

[5] J.R. Iyengar, P.D. Amer, and R. Stewart, "Concurrent Multipath Transfer Using SCTP Multihoming over Independent End-to-End Paths," *IEEE/ACM Transactions on Networking*, vol.14, no. 5, pp. 951-964, Oct. 2006.

[6] O. Issa, J.C. Grégoire, and Y. Belala, "An Embedded Platform for Reliable Applications over Heterogeneous Wireless Networks," *Symposium on World of Wireless, Mobile and Multimedia Networks*, pp.1-6, Jun. 2008.

[7] "IEEE Standard for Local and Metropolitan Area Networks - Part 21: Media Independent Handover Services," *IEEE LAN/MAN Std. 802.21-2008*, Jan. 2009.

[8] G. Lampropoulos, C. Skianis, and P. Neves, "Optimized Fusion of Heterogeneous Wireless Networks Based on Media-Independent Handover Operations," *IEEE Wireless Communications*, 2010.

[9] "TCP Extensions for Multipath Operation with Multiple Addresses," IETF Multipath TCP Working Group, 2012.

[10] Y.R. Chang, "Reduce Delay in Topology-mismatched Handover under Linux," *Master Thesis*, National Chiao Tung University, 2011.