Wireless Pers Commun (2012) 66:57–79
DOI 10.1007/s11277-011-0326-3

An Architecture for Exploiting Multihoming in Mobile Devices for Vertical Handovers & Bandwidth Aggregation

Muhammad Yousaf · Amir Qayyum · Shahzad A. Malik

Published online: 8 May 2011

Abstract In recent years, mobile devices with multihoming capabilities i.e. equipped with multiple network interfaces have gained large scale popularity. This multihoming capability enables the mobile devices to connect with multiple diverse access networks simultaneously. However, networking protocol stack implemented in current devices is not capable of exploiting the availability of multiple network interfaces. Multihoming can be used to provide two important services: *vertical handovers* and *bandwidth aggregation*. Vertical handover enables a multihomed device to switch its connectivity from one access network to another access network without disrupting the communication session. Bandwidth aggregation enables multihomed device to achieve higher throughput by establishing simultaneous connections over multiple available network interfaces. A number of solutions have been proposed to exploit multihoming for vertical handovers and bandwidth aggregation. However, most of these solutions either require the support of additional network entities such as host agent, foreign agent, mobility gateway, proxy, etc. or they require changes in current widely deployed protocol stack in operating system kernels. Dependence on either network operator, administrator or operating system vendors hinders the large scale deployment of these solutions. This paper presents an end-to-end architecture that offers the vertical handover and bandwidth aggregation services to TCP applications. This architecture neither requires any additional network entity nor it requires the changes in current networking protocol stack in operating system kernels. The paper presents the design, implementation and performance analysis of the proposed architecture.

M. Yousaf (✉)
Faculty of Computing, Riphah International University (RIU), Islamabad, Pakistan
e-mail: myousaf@ymail.com

A. Qayyum
CoReNeT, M. A. Jinnah University (MAJU), Islamabad, Pakistan
e-mail: aqayyum@ieee.org

S. A. Malik
Comsats Institute of Information Technology (CIIT), Islamabad, Pakistan
e-mail: smalik@comsats.edu.pk

Keywords Multihomed devices · Diverse access technologies · Vertical handover · Bandwidth aggregation · End-to-end solutions · TCP

1 Introduction

Mobile devices with multiple network interfaces are getting wide spread popularity. Current laptops usually have Ethernet, Wireless LAN and Bluetooth interfaces and optionally can have WiMAX and 3G interfaces. Similarly, the availability of GPRS/EDGE, 3G and WLAN interfaces has rendered smart phones as multihomed devices. Characteristics and capabilities of these diverse access technologies largely vary from each other. Existence of these multiple network interfaces can be utilized to provide the services of vertical handover and bandwidth aggregation. Vertical handover enables a multihomed device to switch its connectivity from one access network to another access network without disrupting the application data flow. Bandwidth aggregation enables the multihomed devices to transmit data of a single flow over multiple network interfaces simultaneously. Many protocols have been proposed to provide these services, but these protocols could not have been deployed widely due to the issues like requirement of additional supporting entities in the network, change in current protocol stack and thus change in operating system kernel, lack of complete specifications, etc. Although, users today, possess multihomed mobile devices but communication protocol stack implemented in these devices does not facilitate the exploitation of multihoming. This paper presents the design and implementation of an end-to-end architecture that overcomes these limitations and provides the vertical handover and bandwidth aggregation services for multihomed devices.

The proposed architecture is designed on the end-to-end principle that relies only on currently deployed network infrastructure and requires no additional network entity to provide its services. Also, it does not require changes in TCP/IP protocol stack as implemented in operating system kernels. Besides vertical handovers and bandwidth aggregation, it also provides the services of willful handovers. *Willful handover* allows the connection to be shifted from one network interface to the other even if first interface is not down. This service facilitate the multihomed device to manage its network connectivity in accordance to the user preferences. Whereas, *forced handovers* are those when mobile node is either loosing its connectivity or has lost the connectivity with its current access network and it has no other choice except to handover to any other available access network.

To protect handover and bandwidth aggregation control messages from redirection attacks, proposed architecture also provides an authentication mechanism. This avoids the necessity of a separate security protocol for the authentication of control messages. Moreover, issues like simultaneous handover of both communicating mobile devices and the handover during bandwidth aggregation have also been addressed in the paper. Performance of the proposed architecture has also been evaluated in the paper.

Next section discusses the work related to the provisioning of vertical handover and bandwidth aggregation services to multihomed mobile devices. Section 3 discusses the proposed architecture, purpose and details of its constituent modules. Section 4 describes how vertical handover and bandwidth aggregation services can be provided using the proposed architecture. The performance of the proposed architecture is analyzed in Section 5 and Section 6 concludes the paper.

2 Related Work

Many protocols have been proposed to provide handover management and bandwidth aggregation using multihoming property of mobile devices. Mobility and multihoming extension for Host Identity Protocol (HIP) is defined in [1]. To support mobility in HIP, a *locator* parameter is defined that is used in HIP exchange messages to notify peer node about the alternate address at which mobile host can be reached. Although mobility support is defined but support for multihoming and bandwidth aggregation over multiple interfaces is not yet properly defined in HIP.

Stream Control Transmission Protocol (SCTP) has been proposed to benefit from multihoming property of mobile devices. A concurrent multipath transfer for SCTP has been discussed in [2,3] that mostly addresses the issues of unnecessary fast retransmissions, congestion window growth problem and increased ack traffic during bandwidth aggregation. An Extension of SCTP called mSCTP has also been proposed in [4,5] to provide vertical handover service using SCTP. However, SCTP is a separate transport layer protocol that does not help TCP applications to benefit from multihoming.

To enable TCP for getting benefit from multihoming capability of mobile devices, some variants of TCP have also been proposed. For example, a virtual pipe concept was proposed in [6] and [7] to provide the bandwidth aggregation and mobility management services for TCP applications. But, problem with these TCP variants is that the deployment of these protocols require changes in operating system kernel and operating system vendors are not showing interest in implementing these proposed variants.

Recently a new working group has been formed in IETF as Multipath TCP (MPTCP) Working Group. Objective of MPTCP is to enhance the application performance through *Path Diversity* mechanism [8]. WG intends to develop the mechanisms that simultaneously use multiple paths for a regular TCP session. MPTCP is being designed on the end-to-end principle that will be deployable without significant changes to the existing Internet infrastructure. However, changes in operating system kernel will be required to implement the MPTCP. Also, MPTCP requires changes in protocol stack on both peer nodes. Focus of MPTCP is to identify and utilize the multiple paths that may be independent of the network addresses of the end nodes; and also devise a common congestion control mechanism for these multiple paths. So far, work is in its initial stages and only architectural guidelines have been defined as an informational RFC.

Another IETF working group is working as Multiple Interfaces (MIF) that intends to handle the configuration issues of devices having multiple network interfaces [9,10]. These interfaces can be physical as well as logical. MIF mainly addresses the issues related to the global configurations that may vary among the interfaces. Currently, MIF does not handle the mechanisms that enable traffic flow to move from one network interface to the other. MIF has presented couple of internet drafts defining problem statement and current practices. How to efficiently exploit the existence of these multiple interfaces for user applications is currently out of scope of the working group.

Baseline idea of the architecture, described in this paper, was presented in [11]. This paper provides the implementation design and performance analysis of the proposed architecture. Moreover, this paper describes how bandwidth aggregation can be provided over multiple interfaces. Also, some complex scenarios like handover during bandwidth aggregation and simultaneous handover of two communicating mobile devices are addressed in this paper.

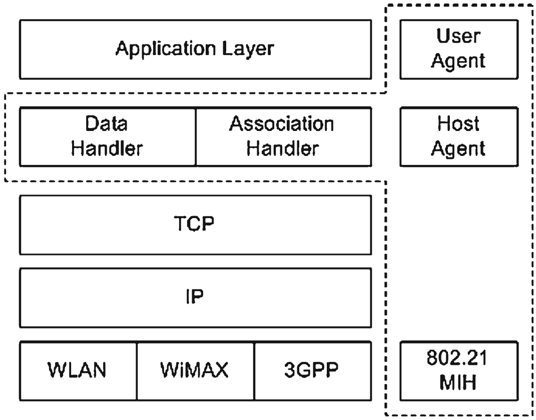


Fig. 1 Proposed architecture showing session layer & cross layer components

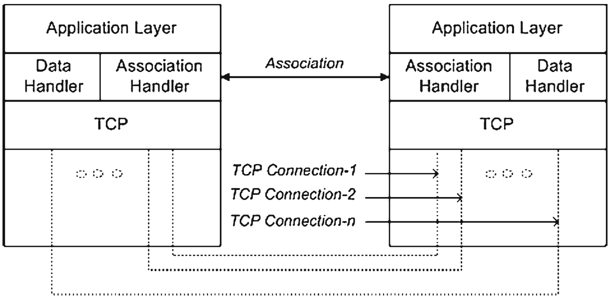


Fig. 2 Multiple connections under single association

3 Proposed Architecture for Exploiting Multihoming

As depicted in Fig. 1, the proposed architecture is a combination of session layer and cross layer components. Session layer components of the architecture manage the association between two communicating nodes and handle data exchange during handover and bandwidth aggregation states. While, cross layer components gather network conditions from network interfaces, get user preferences from the user and manage handover and bandwidth aggregation decisions.

In the proposed architecture, an association is established between two communicating nodes. This association is maintained on top of the transport layer. Multiple TCP connections can be established under a single association as shown in Fig. 2. This allows to transmit/receive data of a single application flow over multiple TCP connections. This we refer to as *connection diversity*. For association establishment, it is necessary for both communicating nodes to have support of proposed architecture. Before establishing the association, it is checked whether the support is available at both communicating nodes or not. If the support is available then the association is established, otherwise a normal TCP connection is established between the two nodes.

Vertical handover service is suitable for long lived connections, whereas bandwidth aggregation benefits the applications with high throughput requirements. These services may not make much sense for short lived or request/response type of applications. Therefore, it may not be suitable to establish every TCP connection under the association. Users may give preference of application types or protocols for which associations may be established. Whenever, there is a request for connection establishment, decision to establish the association for this connection request is made by taking into consideration the user preferences.

At the sending side, application data received under the association is scheduled for the transmission over multiple TCP connections. To ensure in-order delivery, association sequence numbers are included in session layer packets. At the receiving side, the packets are brought in order using these association sequence numbers.

The proposed architecture has been designed to provide support for both legacy applications as well as new multihoming-aware applications. A multihoming-aware library has been developed. New applications can call this library to provide the vertical handover and bandwidth aggregation services. To provide support for legacy applications, we used library function interception technique using *ld_preload* [12,13]. For this, we created a shared library that overloaded the standard socket calls to provide the vertical handover and bandwidth aggregation services to the legacy applications.

3.1 Session Layer Components

Session layer components include the *Association Handler* and *Data Handler* modules. Association Handler manages the association establishment process while Data Handler schedules and reassembles the data over multiple connections during handover and bandwidth aggregation states. These components also interact with cross-layer components to receive handover and/or bandwidth aggregation commands.

3.1.1 Association Handler

In the proposed architecture, whenever a connection request is received from an application, decision regarding the establishment of an association is made taking into consideration the user preferences. This is called *local-compliance check*. If connection request is not locally compliant then association is not established and normal TCP connection is established with the remote node. However, if it is locally compliant then association establishment process is started and remote node is checked for the *remote-compliance*. If the support of proposed architecture is not available on the remote node or a not-compliant response is received from remote node, in both cases association is not established and normal TCP connection is established. However, if remote node sends the association establishment response message, then an association is established between two communicating nodes. Two important tasks that are performed during association establishment process are (i) generation of unique association identifier (AID), and (ii) exchange of shared key for the association.

Each association is identified by a 32-bit Association Identifier (AID). AID must be unique on two communicating nodes. It needs not be globally unique. In our implementation, to ensure the AID uniqueness, we created the 32-bit AID as two concatenated 16-bit identifiers. Initiator of association sends 16-bit *Initiator-AID* that is unique at the initiator side. Responder sends 16-bit *Responder-AID* that is unique at the responder side. Both the nodes concatenate these two 16-bit numbers to generate the 32-bit AID that is unique at two communicating nodes. This AID is used in vertical handover and bandwidth aggregation control messages.

The initiator sends its public key in Association Establishment Request message, and responder sends its public key in Association Establishment Response message. Both the nodes process these keys to generate a shared secret key. It is recommended to use public key certificates to avoid the Man-in-the-Middle attack during public key exchange [14]. To avoid security attacks such as redirection attack, AID in vertical handover and bandwidth aggregation control messages is encrypted with the shared key between two communicating nodes. To generate the shared key, any shared secret key exchange mechanism can be used. In our implementation, we used Elliptic Curve Diffie Hellman (ECDH) for shared key exchange [15]. Further, Advanced Encryption Standard (AES) [16] is used to encrypt the AID along with a nonce. This nonce is used to avoid the possible replay attacks [14].

3.1.2 Data Handler

After the association is established, whenever application delivers data for the transmission, Data Handler schedules the data according to the state in which it is at that time. There are three different states in Data Handler that are (i) *Normal State*, (ii) *Handover State*, and (iii) *Bandwidth Aggregation State*.

When Data Handler is in normal data transmission state, it simply forwards the data to the TCP. It means that there is only one TCP connection in this association. It is almost like data transmission over traditional TCP connection.

Whenever, there is a handover, a new connection with the peer node is established. If this is vertical handover then it means this new connection is established on different network interface. It is possible that previous interface is still up, in which case there can be two connections in handover state. In this state, data sent on one connection is also sent on the other connection, thus providing the *connection diversity*. Peer node receiving data on either of the connections, delivers data in-order to the application. Duplicate data that is likely to be received on the other connection is simply discarded. This simultaneous transmission of data on two connections in handover state facilitates the smooth handover. Old connection is closed when new connection reaches the stable state or handover time-out occurs. Handover time-out is introduced to cater to the situation when new connection takes a long time in reaching the stable state. This may happen because of frequent packet losses either due to poor wireless link conditions or network congestion.

For bandwidth aggregation, Data Handler can exploit the existence of multiple network interfaces as they become active. The communicating side having multiple network interfaces can initiate bandwidth aggregation irrespective of the fact that peer node may or may not possess multiple interfaces. Proposed architecture decides about the initiation of bandwidth aggregation on the basis of user preferences. User may configure the particular applications/protocols for which bandwidth aggregation should be used. In bandwidth aggregation state, data handler transmits data on multiple connections. In contrast to the handover state, in bandwidth aggregation state, different data is transmitted on multiple connections. More specifically, in handover state, packets with same association sequence numbers are scheduled on two connections, whereas in bandwidth aggregation, packets with different sequence numbers are scheduled on multiple available connections. This enables Data Handler to transmit data at higher rate that can help the multihomed device to overcome throughput bottleneck on its side. However, other end's bottleneck is not eliminated unless that end also has multiple network interfaces up. This is beneficial especially in cases when a mobile node with multiple network interfaces is communicating with a high end server. In this way, mobile node, using multiple network interfaces can eliminate the throughput bottleneck of its side. As different network interfaces may not have same bandwidth, thus in order to reduce the possible head-of-line blocking, data is scheduled on multiple connections with the rate at which a connection is transferring the data to the peer node. Data transmission performance is measured for each connection. While scheduling the data, priority is given to the connections taking into consideration their throughput performance.

At the peer node, data received on multiple connections is brought in-order with the help of association sequence numbers. If out-of-sequence data exceeds the head-of-line blocking threshold, then Data Handler at receiving side can request the Data Handler at sending side for retransmission of packets. This retransmission request includes the expected association sequence number at which the blocking is occurring.

3.2 Cross-Layer Components

There are three cross-layer components in the proposed architecture: (i) User Agent, (ii) IEEE 802.21 MIH and (iii) Host Agent. User Agent provides an interface to the user that allows to configure preferences about applications, protocols and network interfaces. User Agent can also be used to give manual willful handover and bandwidth aggregation commands. IEEE 802.21 MIH is used to gather link information from heterogeneous network interfaces in a unified way. MIH can provide link triggers that indicate the change of network interface conditions. These triggers are passed to the Host Agent. Host Agent makes decisions about whether handover or bandwidth aggregation is to be initiated or not.

3.2.1 User Agent

For multihomed mobile devices, handover decision based only on link conditions is not appropriate. For heterogeneous access networks, there are several other parameters that might affect the handover decisions. For example, monetary cost of using some access network is an important parameter that usually affects the user's choice to switch from one network to the other. Moreover, achievable data rate is also an important parameter that affects the user's choice to connect to a particular access network. This is primarily due to significant difference in available data rates for different access networks. For example, achievable data rate in GPRS is in 100s of Kbps, whereas data rate achievable in IEEE 802.16 WiMAX is in Mbps. Therefore, if a user has equal cost and proportionally equal link quality then achievable data rate on a particular access network is a decisive parameter. On the basis of these parameters, user can have higher preference for a particular access network as compared to the other access networks.

Other than handover preferences, user might also be interested in giving his preferences about the network interfaces to be used for bandwidth aggregation. More often users in offices have free network access through office's Ethernet and WLAN interfaces, whereas they may have their own subscriptions of WiMAX and GPRS. For example, consider a user who has a multihomed device with four network interfaces i.e. Ethernet, WLAN, WiMAX and GPRS. User may be interested in using Ethernet and WLAN for bandwidth aggregation for high data rate applications and may not be willing to use WiMAX and GPRS for bandwidth aggregation due to their higher cost. This is totally a matter of user's choice and has little to do with the link quality.

Another aspect of user preferences is the type of applications for which he/she might be interested in using the services of handover and bandwidth aggregation. For example, it makes better sense to use handover and bandwidth aggregation services only for long lived applications like FTP, whereas for applications that use short lived connections, it does not make much sense to preserve the connection or to use bandwidth aggregation for small data exchanges. For example, it does not make sense to use bandwidth aggregation for simple DNS query. Due to these reasons, User Agent has been designed so that users may configure their preference of application types for which handover and bandwidth services should be used.

3.2.2 IEEE 802.21 Media Independent Handover

IEEE 802.21 MIH is a recent standard that assists the handover decision making for the mobile devices [17]. MIH provides the link information to the upper layer decision making entity called *MIH User* in a media independent way. It provides this information in different forms e.g. it can provide the link events in the form of triggers: such as *link up, link down, link going down,* etc. Although MIH was basically proposed for facilitating the vertical handover decision making [18], however in the proposed architecture, we also used the MIH triggers for facilitating the bandwidth aggregation decision making. Host Agent may initiate handover or enter bandwidth aggregation state on receiving *link up* trigger, and handover and exit from bandwidth aggregation state on receiving *link down* and *link going down* triggers.

```
input  : MIH Link Layer Trigger Received
output: Handover/Bandwidth Aggregation/DDNS Update Decision Command
trigger ⇐ receivedMIHtrigger
if trigger == LinkUP then ;                          // link up trigger received
    if isLocationUpdateEnabled() then
        sendDNSupdate(addRecord, UPlink)
    end
    if isHOenabled(UPlink) then
        k ⇐ getPref(UPlink)
        l ⇐ k − 1
        foreach l < k do ; // possbile HO from low preference link
            if isUP(link_l) then
                HO(from_l, to_k);
                break;
            end
        end
    end
    if isBAenabled(UPlink) then
        addToBA(UPlink)
    end
end
if trigger == (LinkDown OR LGD) then ;              // LD or LGD trigger received
    if isLocationUpdateEnabled() then
        sendDNSupdate(deleteRecord, LDLGDlink)
    end
    if isThereNonBAtrafficOnLink(LDLGDlink) then
        k ⇐ getPref(LDLGDlink)
        l ⇐ k − 1
        foreach l < k do ; // possbile HO to low preference link
            if isUP(link_l) & isHOenabled(link_l) then
                HO(from_k, to_l);
                break;
            end
        end
    else
        removeFromBA(LDLGDlink)
    end
end
```

Algorithm 1: Handover/Bandwidth Aggregation Decision Algorithm

3.2.3 Host Agent

Host Agent is a cross layer decision module that obtains user preferences from User Agent and MIH triggers from the IEEE 802.21 MIH layer. From the perspective of 802.21 MIH, Host Agent acts as MIH User. On the basis of link layer triggers from MIH and user preferences from User Agent, Host Agent makes handover and/or bandwidth aggregation decisions and passes the handover/bandwidth aggregation commands to the Data Handler. In addition to these decisions, Host Agent can also send dynamic DNS updates for mobile servers. When a trigger from 802.21 MIH is received, Algorithm-1, described in the following, is executed in order to decide about the initiation of handover and bandwidth aggregation processes.

As also shown in Fig. 3, when a link layer trigger is received from 802.21 MIH module, first it is checked whether it is a Link UP trigger or Link Down/Link Going Down Trigger. In both cases, it is checked that if user has given the preferences for sending location updates in the form of Dynamic DNS Updates. If location updates are enabled then on receiving Link UP trigger, DNS Add Record message is sent to the corresponding name server. Then, it is checked from the user preferences that if the link that recently has become UP is defined as Handover_Enabled or not. If it is HO_enabled then the traffic on lower priority link is handed over to the recently UP link. Similarly, if it is defined in user preferences that whenever this link becomes UP, it can be used for bandwidth aggregation, then recently UP link is also added in the list of links that are being used for the bandwidth aggregation.

Similarly, when Link Down or Link Going Down (LGD) trigger is received then for location updates, DNS Delete Record message is sent to the corresponding name server. In the proposed architecture, we differentiate the user traffic flows as BA or Non-BA flows. Traffic of BA flows is sent over multiple network interfaces to get bandwidth aggregation benefit. While Non-BA traffic is sent only on a single network interface. Therefore, when a Link Down or Link Going Down (LGD) trigger is received for some link, it is checked that which type of traffic flows where being sent on this link. If there has been BA traffic then this link is simply removed from the BA list. For Non-BA traffic, it is checked that if any lower priority link is UP. If a lower priority link is available then traffic flows are handed over from the link being down to the lower priority link.

For both Link Down (LD) and Link Going Down (LGD) triggers, we are doing the same processing. The reason for this is the availability of intelligent algorithms [19] that can predict with very high probability that the link is actually going down or not.

4 Achieving Vertical Handovers & Bandwidth Aggregation with Proposed Architecture

In this section, we discuss how vertical handover and bandwidth aggregation services are provided with the proposed architecture. The architecture supports both forced as well as willful handovers. Willful handover scenario can occur, when a higher preference link is activated and becomes available for communication, then connections on lower preference link are handed over to the higher preference link. This can also happen, when due to any reason, user manually switches his connectivity from one link to another link. We refer it as *manual willful handover* whereas, former is referred as *automatic willful handover*.

For supporting the bandwidth aggregation service, users can configure their preference for the type of applications and/or protocols for which bandwidth aggregation should be used.

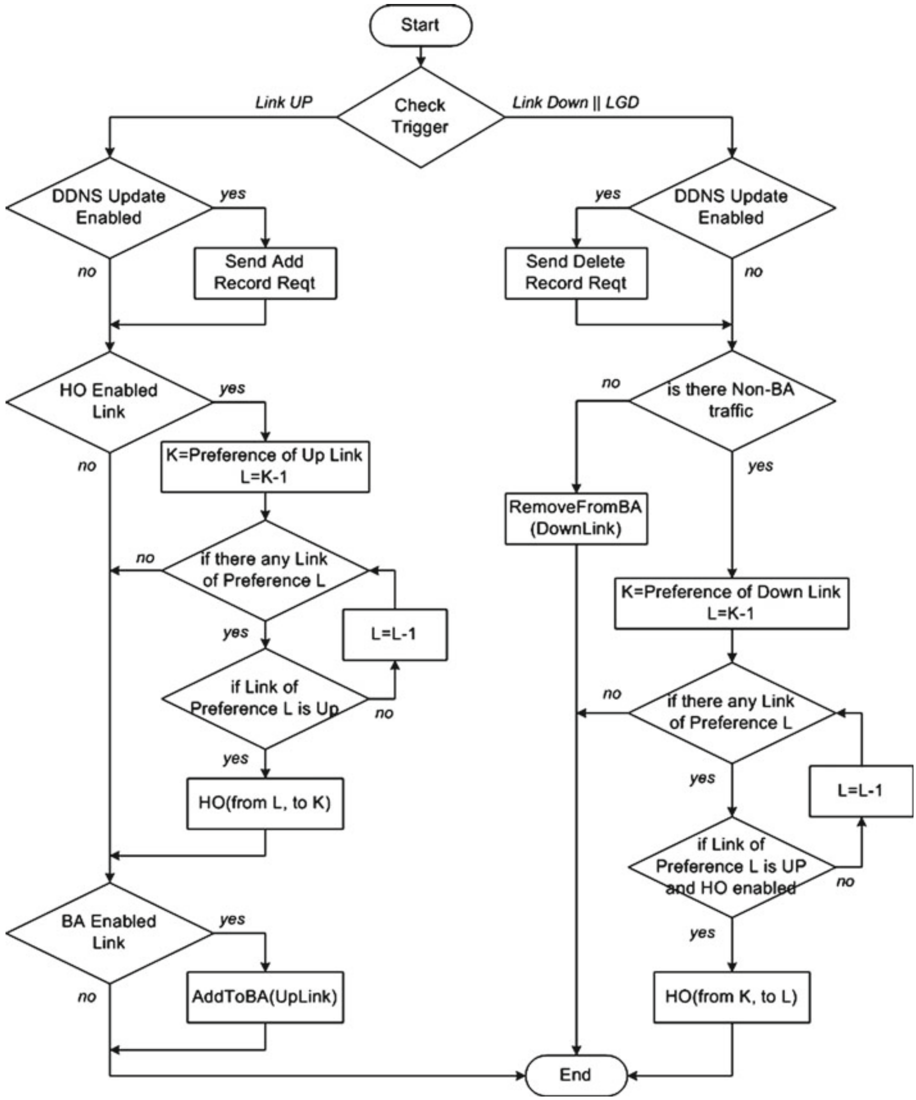


Fig. 3 Flow chart of handover/bandwidth aggregation decision algorithm

It is desirable that bandwidth aggregation should not be used for all types of applications. Specifically, short lived request/response type of applications are not suitable for this service. Bandwidth aggregation makes more sense for high data rate applications or bulk data transfer applications.

4.1 Handover Management

Whenever a link goes down and an alternate link is available or a link with higher preference than the existing active link comes up or user manually issues a command to switch the inter-

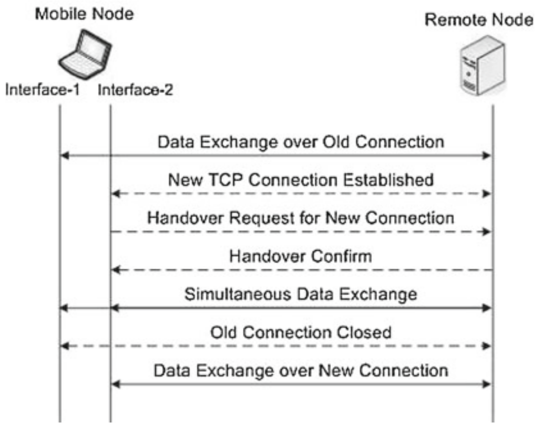


Fig. 4 Message exchange during handover

face due to any reason, Host Agent checks the active associations on the link from where the handover is to be made. Host Agent issues the handover command to Data Handler. As there may be multiple associations on a mobile node, so Data Handler exchanges the handover messages for each of the associations with the peer Data Handler.

Data Handler encrypts the AID and nonce and sends *handover request* message to the remote node. Data Handler at the remote node decrypts and validates the AID and nonce. If these are not validated then peer Data Handler discards handover request and sends the *handover reject* message. If AID and new nonce are validated then Data Handler at remote node again encrypts the AID and new nonce and sends back a *handover confirm* message. On reception of *handover confirm* message, handover initiating node decrypts and validates the AID and new nonce. If these are not validated or Data Handler has not sent the handover message, Data Handler discards the message. However, if AID and new nonce are validated, then Data Handler enters the handover state.

Simple message exchange during handover process is shown in Fig. 4. However, besides this simple handover, some complex handover scenarios e.g. simultaneous handover is also possible. Figure 5 shows different such scenarios. In these diagrams there are two nodes A and B each having two network interfaces (a_1, a_2) and (b_1, b_2) respectively. Scenario (i) is simple HO as described in Fig. 4 also. Scenario (ii) shows the situation in which, at the time when node B is wishing to send HO Request, before sending HO Request to node A, node B itself receives HO Request from node A. In this case, node will send the HO Request with updated IP address of new interface of node A. This is also depicted in Fig. 6.

In scenario (iii), both nodes A and B send HO Request messages simultaneously to each other from their respective new interfaces. However, old interfaces at both the nodes are also reachable. Thus, both the nodes receive respective HO Requests at their old interfaces. Now both the nodes will again send HO Requests to each other with new IP addresses of new interfaces learned from old HO Requests. This scenario is also depicted in Fig. 7.

Scenario (iv) is Hard HO scenario in which both nodes A and B send HO Requests simultaneously to each other from their respective new interfaces and their old interfaces are no longer reachable. Thus, both HO Requests are not reached at the respective nodes. Both the nodes will query the name server to get updated IP addresses of each other. After getting response from the name server, both the nodes will send HO Requests to each other with new IP addresses. This scenario is depicted in detail in Fig. 8.

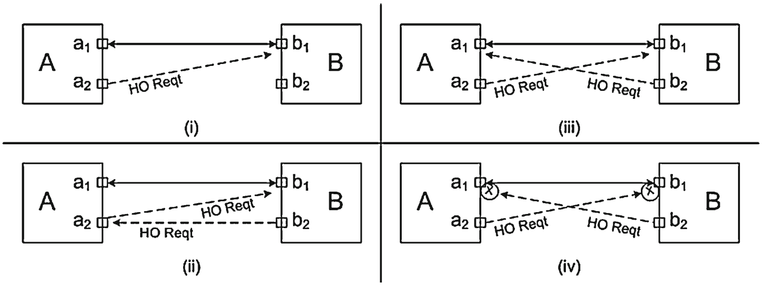


Fig. 5 Simultaneous handover scenarios

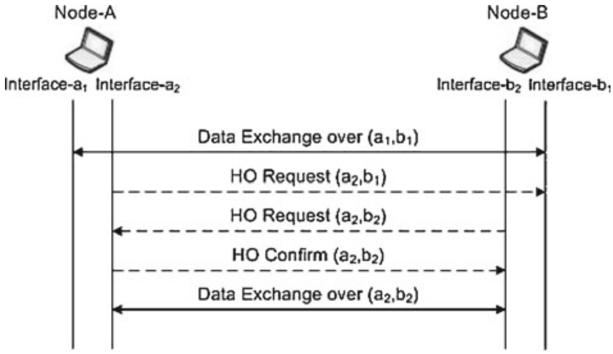


Fig. 6 Simultaneous handover scenario-2

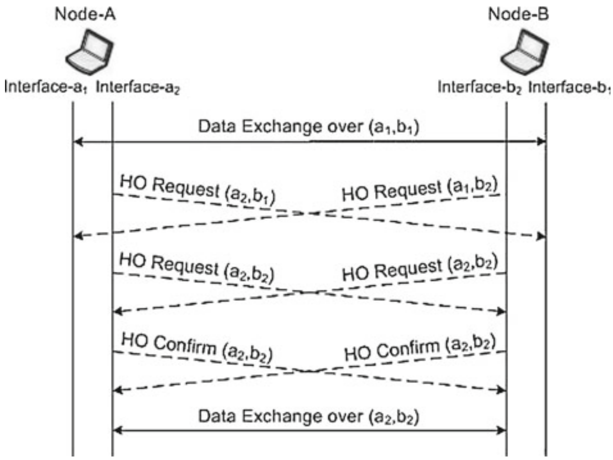


Fig. 7 Simultaneous handover scenario-3

4.2 Bandwidth Aggregation Management

Whenever a new network interface goes up, Host Agent checks whether to use this link for bandwidth aggregation or not. Host Agent informs Data Handler to initiate bandwidth aggregation process for certain associations. Data Handler establishes new connection with

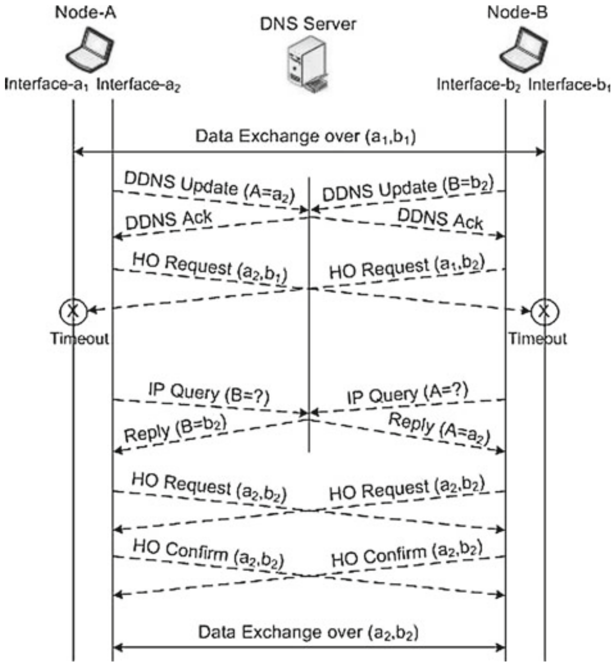
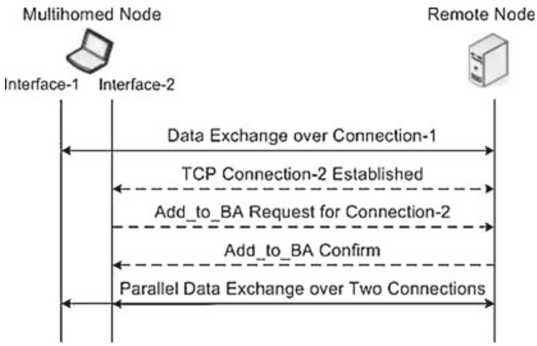


Fig. 8 Simultaneous handover scenario-4

Fig. 9 Message exchange during bandwidth aggregation



the remote Data Handler. After this, initiating node sends the *add_to_BA request* message to the remote Data Handler. This message contains the encrypted AID and nonce. Remote Data Handler, after authentication, sends back the *add_to_BA response* message. Upon reception of this response message, Data Handler enters in the bandwidth aggregation state. In this state, Data Handler has multiple TCP connections under a single association. In bandwidth aggregation state, the sender transmits data over all connections under an association. While transmitting data, Data Handler also computes the performance measure of each connection. Data on individual connections is scheduled taking into consideration the computed performance measure of each connection. Message exchange to start the bandwidth aggregation is depicted in Fig. 9.

During bandwidth aggregation, if an interface goes up or down, then it is not necessary to perform handover in all the scenarios. Decision regarding handover depends on whether

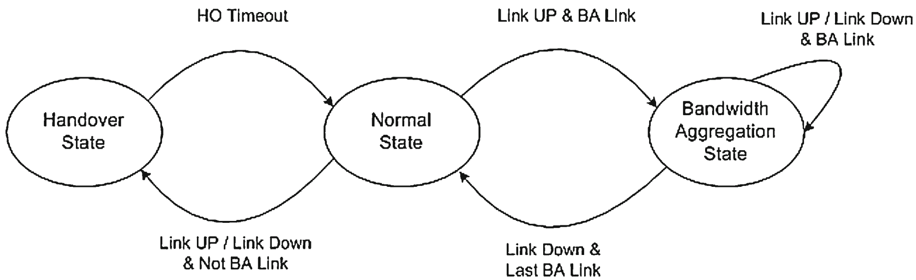


Fig. 10 State diagram for handover and bandwidth aggregation state changes

the aggregation is being performed on multiple interfaces or only on a single interface. If bandwidth aggregation is being performed on multiple interfaces and one of these interfaces goes down then no handover procedure is initiated. Only *remove_from_BA request* message is sent to the peer node and after authentication, that interface is removed from the bandwidth aggregation. Data already scheduled on the connection of that link as well as further data from the application is scheduled on the connections of remaining active interfaces. Similarly if some new network interface goes up then again no handover procedure is initiated. Rather *add_to_BA request* message is sent to the peer node and this active interface is included in the bandwidth aggregation.

However, if a node has only one up interface and peer node has multiple interfaces, the peer node can initiate the bandwidth aggregation process. In this case, node with single active interface will have multiple connections for bandwidth aggregation over a single network interface. This scenario can also occur if preferences to include the network interfaces for bandwidth aggregation are not defined on the node. In this case, if this single interface goes down and some lower preference link is available then handover process can be initiated during the bandwidth aggregation state. This is a composite scenario and we propose that bandwidth aggregation connections should not be directly handed over to the new connections. Rather, we suggest that bandwidth aggregation state should first be shifted to the normal state i.e. if there are multiple connections for the aggregation then first close all the connections except the one with the highest performance measure. Then, this single connection is handed over to the connection at lower preference available interface. Once this handover is completed then, node having multiple interfaces can again initiate bandwidth aggregation.Thus handover during bandwidth aggregation will undergo following state changes: $BAstate - > NormalState - > HOstate - > NormalState - > BAstate$. These state changes are also depicted in Fig. 10.

5 Performance Analysis of Proposed Architecture

For analyzing the performance of proposed architecture, we implemented and tested it on Linux Fedora Core 9 and Windows XP platforms. We used laptops with multiple network interfaces as multihomed devices. Implementation design of proposed architecture is shown in Fig. 11. As shown in figure, we developed a bandwidth aggregation and handover aware library that we termed as *BAHO library*. This library includes the overloaded socket calls for providing the bandwidth aggregation and vertical handover services to the applications running over multihomed mobile devices. In order to provide support for legacy applications, we developed a *System Call Translator (SCT)* module that intercepts and translates

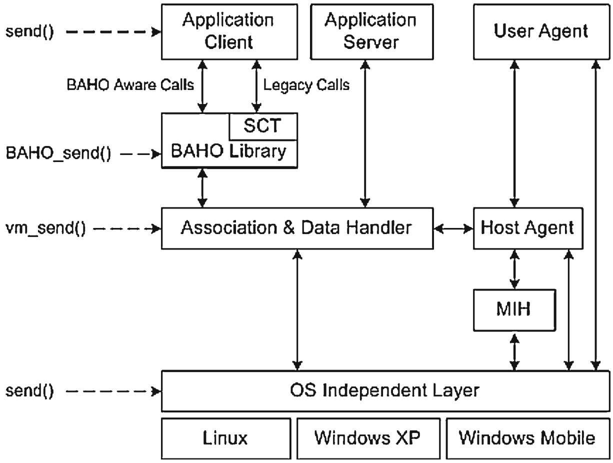


Fig. 11 Implementation design of proposed architecture

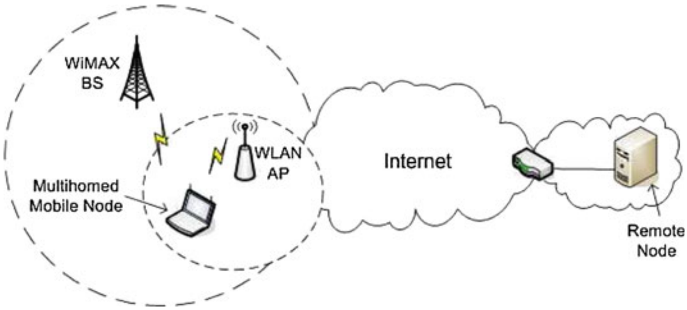


Fig. 12 WiMAX-WLAN topology for performance evaluation

the socket calls from legacy applications into the multihoming-aware *BAHO library* calls. In our implementation, we also developed an operating system independent virtual layer. This virtual layer provides the abstraction from the underlying operating systems. In this way, we were able to execute same code over multiple operating system platforms. Also, we implemented a subset of IEEE 802.21 MIH functionality. This subset provides the triggers and information about the locally attached network interfaces only. For simplicity, we did not implement the network services of IEEE 802.21 MIH.

Variety of tests were performed in both WAN and LAN environments. Experimental setup for WAN is depicted in Fig. 12. For this, we used multihomed laptops equipped with WLAN and WiMAX interfaces. For WLAN, we used Atheros AR242x 802.11abg WLAN interface and for WiMAX, we used Motorola WiMAX USBw35100 interface.

5.1 Throughput & Latency During Handovers

Proposed architecture uses the idea of connection diversity for getting seamless handover. Whenever, mobile node is in overlapping coverage area of two access networks, and it needs to be handed over from one access network to the other, mobile node is able to establish connection using both the access networks. For the time, until new connection gets stable, Data

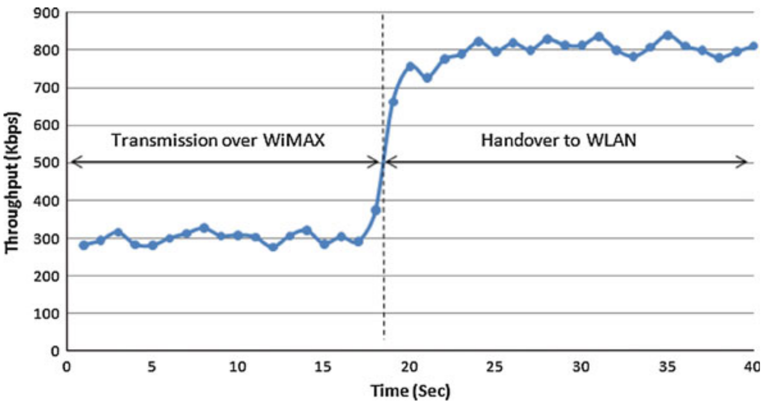


Fig. 13 Throughput performance during handover from WiMAX to WLAN

Handler continues to exchange data over the previous connection, as well. As soon as new connection gets stable, previous connection is closed. Due to this simultaneous transmission, there is no throughput degradation during the handover and thus applications experience a seamless handover. Figure 13 shows the vertical handover of a multihomed mobile device from WiMAX interface to WLAN interface. Figure 14 shows the vertical handover from WLAN interface to WiMAX interface. In both these vertical handovers, we can observe the change of throughput that applications receive from the network. This change of throughput is due the difference in the data rates provided by the underlying heterogeneous access networks. However, if data rate supported by the underlying network interfaces is same then applications will experience a completely seamless handover as shown in the Fig. 15. It can be seen that there is no throughput degradation during the handover period. This seamless handover is due the simultaneous transmission of same data over two connections. Application get data received on either of the connections. In this way, throughput experienced by applications during handover process is the maximum of the throughput achieved on either of the two connections.

$$TP_{\mathrm{HO}} \leq Max(TP_{c1}, TP_{c2})$$

where TP_{c1} = Throughput experienced over connection 1, TP_{c2} = Throughput experienced over connection 2.

Although for the regions with overlapping coverage area of multiple access networks, the throughput degradation time during handover is minimal and thus is the handover latency. However, for the areas where overlapping coverage is not available or when link goes abruptly down and MIH is not able to generate link going down event, the handover latency may be larger. First, a new TCP connection with three-way handshake is established and then two handover control messages i.e. handover request message and the handover confirm message, are exchanged between communicating nodes. Therefore, it can take at least 2.5RTTs to complete the handover to a new connection.

$$\text{HO Latency} = \begin{cases} 0 & \text{(For overlaping regions)} \\ D_{3whs} + D_{cm} + D_{1wd} & \text{(For non overlaping regions)} \end{cases}$$

where D_{3whs} = TCP 3 way handshake delay, D_{cm} = Handover control messages exchange delay, D_{1wd} = 1 way delay of data exchange after handover.

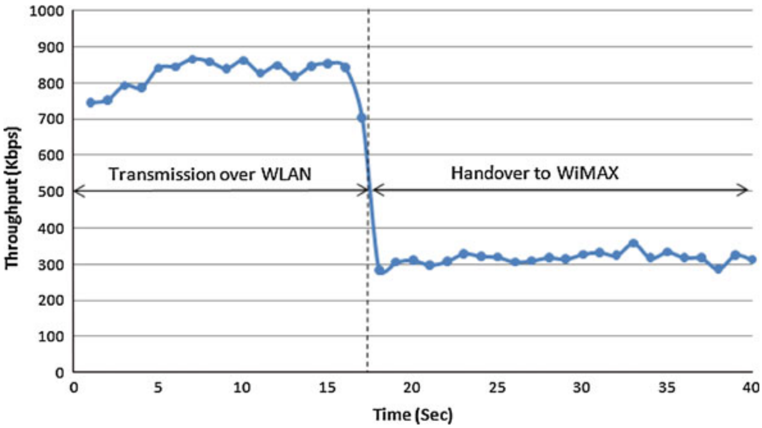


Fig. 14 Throughput performance during handover from WLAN to WiMAX

D_{1wd} will be experienced if node initiating handover is the sender of the data. This is due to the reason that node initiating handover will not send data until it receives the *handover confirm* message from peer node. However, if peer node is sending data, then it can send it immediately after sending the *handover confirm* message. In this case D_{1wd} will be negligible. It should be noted that above mentioned handover delay in non-overlapping region is the delay from the time when MIH issues a trigger to the handover completion time. This delay does not include the link layer association delay and IP address acquisition delay.

5.2 Throughput Gain During Bandwidth Aggregation

When multiple network interfaces are available then multiple connections can be established to get benefit of bandwidth aggregation. Throughput experienced by the applications during bandwidth aggregation (TP_{BA}) is less than or equals to the sum of throughputs achieved on the individual connection on each interface.

$$TP_{BA} \leq \sum_{i=1}^{n} (TP_i)$$

where n = Number of connections in bandwidth aggregation, TP_i = Throughput achieved on ith connection.

Figure 16 shows the throughput gain achieved during the bandwidth aggregation over WiMAX and WLAN interfaces. Topology used for this test is same as depicted in Fig. 12. During the experiments, it was noted that getting benefit from multiple network interfaces depends upon number of factors. For example one such factor is, whether multiple network interfaces belong to the same network or to different networks. If multiple network interfaces belong to different networks then throughput gain due to bandwidth aggregation is evident, however if multiple network interfaces belong to the same network, then throughput gain depends on the fact that the network is contention-based or contentionless. If network is contention-based, then it is not possible to transmit simultaneously over the contention-based channel, due to this reason, throughput gain with multiple network interfaces of same network may not be achieved.

Throughput gain due to multiple network interfaces also depends on node density in the shared network. Although throughput due to bandwidth aggregation is increased, however as

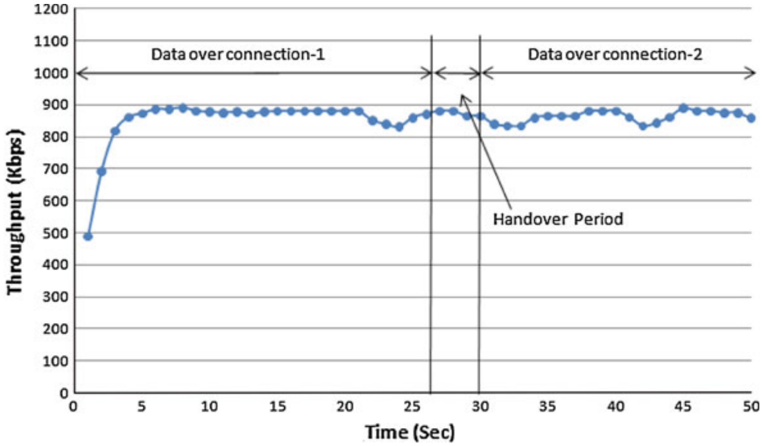


Fig. 15 Throughput performance during seamless handover in same technology (WLAN)

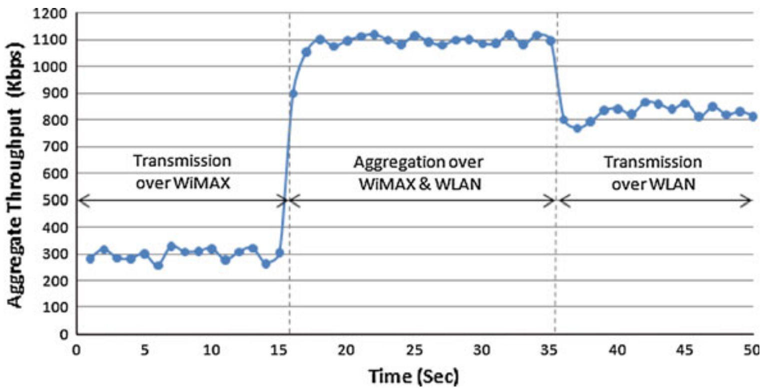


Fig. 16 Throughput gain during bandwidth aggregation over WiMAX & WLAN

with the increase in number of contending nodes in share network, bandwidth share achieved on individual interface is reduced, therefore throughput gain due to bandwidth aggregation is not so evident. This is also shown in Fig. 17. Bandwidth share achieved by a multihomed device having a variety of network interfaces can be presented by the following equation.

$$\text{Bandwidth Share } \leq \sum_{i=1}^{c} \frac{mB_i}{n} + \sum_{j=1}^{f} mB_j$$

where c = number of distinct access networks with contention based MAC, m = number of network interfaces of one network technology at multihomed device, n = total number of interfaces contending in an access network, B_i = total bandwidth of ith contention based access network, B_j = reveived bandwidth in jth contention free access network, f = number of distinct access networks with contention free MAC.

To further elaborate this equation, let's have a WLAN BSS in which a node with one WLAN interface is attached with an Access Point (AP) and it is the only node in BSS. Let this node gets bandwidth B in WLAN. Now, if we add another WLAN interface to the same

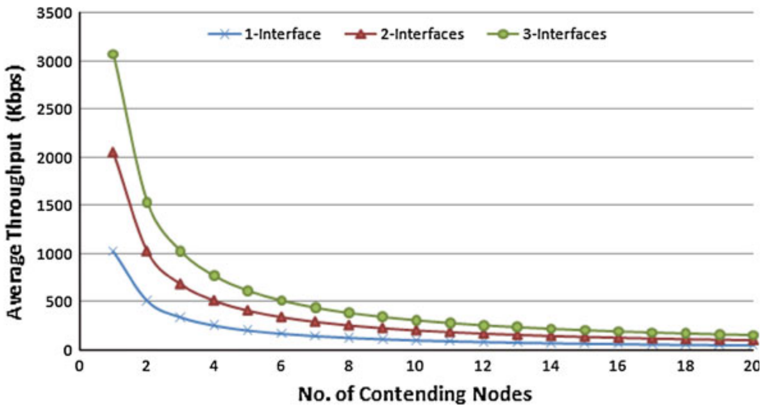


Fig. 17 Throughput gain & degradation with increasing node density

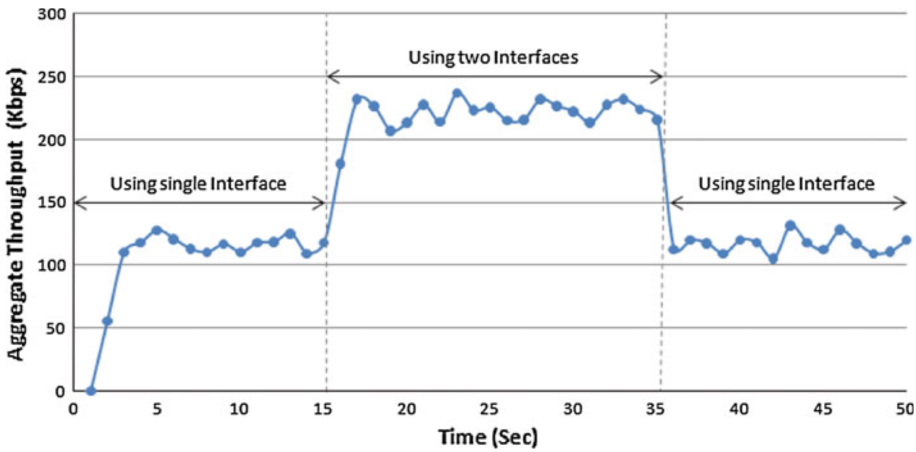


Fig. 18 Throughput gain during bandwidth aggregation over two WLAN interfaces

node attached to the same AP then by equation we have a node with two WLAN interfaces i.e. $m = 2$ and these are the only two interfaces contending in BSS thus $n = 2$ therefore bandwidth share remains $2B/2 \Rightarrow B$. Hence, in this scenario, there is no advantage of bandwidth aggregation. However, if node density in BSS is increased, let's say, to 9 with all the nodes having only one WLAN interface, in this case for every individual WLAN node $m = 1$ and $n = 9$, therefore every individual node will on-average experience the bandwidth share of $B/9$. Now, if we add one additional WLAN interface to one of WLAN nodes making it multihomed then in this case, for the multihomed node $m = 2$ and now total number of contending interfaces are $n = 10$. In this scenario, the bandwidth share of multihomed node will be $2B/10 \Rightarrow B/5$ that is surely $> B/9$. In this case, although total bandwidth of the network remains same however bandwidth experienced by the multihomed node is greater as compared to the single homed node. This fact is also depicted in Fig. 18.

5.3 Scalability of Proposed Architecture

As proposed archicture allows multiple TCP connections under single association for one application flow therefore, increased number of connections may pose the scalability issue. In this section, we discuss the scalability issue in both handover and bandwidth aggregation cases. Let there are two multihomed nodes with number of network interfaces at first node is n and number of network interfaces at other node is m. Then, theoretically, there can be $n \times m$ number of connections for a single application flow. However, in *Normal State*, the proposed architecture allows only *one* connection under an association. In *Handover State*, two connections are established for getting the smooth handover and in *Simultaneous Handover State*, $three$ connections are established. In these case, establishing two or three connections, for a short period of handover state, does not pose the scalability issue.

In *Bandwidth Aggregation State*, not all the network interfaces are used for bandwidth aggregation. Only those interfaces for which, user has given the preferences for including such interfaces, are used for bandwidth aggregation. As such links are limited in number normally 2 or 3 thus, scalability is not a big issue as, these connections are being established in accordance to the user preferences. For resource limited nodes, user can give the preferences for not including extra network interfaces for bandwidth aggregation. Moreover, for limiting the number of connections from $n \times m$ to $\max(n, m)$, capabilities in terms of number of available network interfaces and user preferences at both sender and receiver, can be exchanged between two communicating nodes. However, in current implementation of the proposed architecture, such optimization is not included.

5.4 Overhead of Proposed Architecture

In order to provide vertical handover and bandwidth aggregation services, some computation and transmission overhead is incurred. In this section, we evaluate the impact of this overhead on the performance of applications using the proposed architecture. There are two types of overheads that proposed architecture may introduce. (i) processing overhead, and (ii) transmission overhead in terms of additional header. These overheads may introduce some additional delay that may impact on the end-to-end delay experienced by the application. In the proposed architecture, first computational delay is caused by the scheduling process in the Data Handler module at the sender side (D_{sch}). Secondly, some computational delay is also caused by the reassembly process in the Data Handler module at the receiver side (D_{rs}). Whereas, transmission overhead in terms of additional header causes to increase the transmission delay that in turn increases the end-to-end delay of TCP flow (D_{e2eTCP}). End-to-end delay experienced by the applications (D_{e2eAp}) using proposed architecture can be expressed by the following equation:

$$D_{e2eAp} = D_{sch} + D_{e2eTCP} + D_{rs}$$

where D_{sch} = Scheduling delay caused by Data Handler at sender side, D_{e2eTCP} = End to end delay of TCP flow, D_{rs} = Reassembly delay caused by Data Handler at receiver side.

If we schedule application data at the rate that is higher than the transmission rate, then delay caused by the scheduling $D_{sch} \cong 0$. We achieved this by always filling the TCP buffer with data so that TCP always have data to transmit. However, processing delay during reassembly D_{rs} cannot be avoided. But, this is not of much concern because, $D_{rs} \ll D_{e2eTCP}$. This is why from the application's perspective, the $D_{e2eAp} \approx D_{e2eTCP}$. This is also depicted by Fig. 19. Here, we compared the performance of application using proposed architecture with that not using the proposed architecture and we see no obvious difference in the

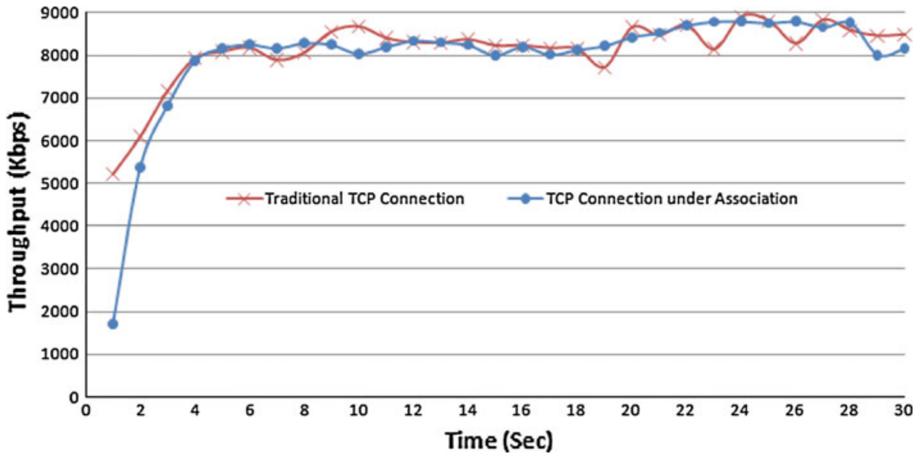


Fig. 19 Throughput comparison with and without proposed architecture

performance of two applications. However, it should be noted that the initial response time is higher when proposed architecture is used. The *initial response time (IRT)* of the applications using proposed architecture is always greater than the IRT of applications using traditional TCP by one RTT. The reason of this is the initial delay in establishing the association between two communicating nodes. But once the association is established, then the effect of delay due to additional computation and additional header is negligible.

6 Conclusion

This paper presented an architecture that exploits the existence of multiple network interfaces in multihomed devices. Proposed architecture provides vertical handover and bandwidth aggregation services for TCP applications. Moreover, architecture also provides the willful handover service. Architecture has built in security mechanism for authentication of handover and bandwidth aggregation control messages. We have implemented and tested the proposed architecture on Linux and Windows platforms. Several scenarios for vertical handover and bandwidth aggregation have been cosidered. Performance analysis results exhibit the capability of proposed architecture to provide vertical handover service seamlessly, as well as the bandwidth aggregation service with prominent throughput gains while using the multiple network interfaces in multihomed devices.

Acknowledgments We would like to thank *National ICTR&D Fund* for supporting this research (http://www.ictrdf.org.pk/). We would also like to thank to the *emfproject team* for their dedicated contribution.

References

1. Nikander, P., Henderson, T., & Arkko, J. (April, 2008). *End-host mobility and multihoming with the host identity protocol*. IETF RFC 5206.
2. Iyengar, J. R., Amer, P. D., & Stewart, R. (2006). Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths. *IEEE/ACM Transactions on Networking, 14*(5), 951–964.

3. Dreibholz, T., & Becke, M. (November 08, 2010). *SCTP socket API extensions for concurrent multipath transfer*. IETF Internet Draft, draft-dreibholz-tsvwg-sctpsocket-multipath-00 (Work in Progress).
4. Koh, S. J., Chang, M. J., & Lee, M. (2004). mSCTP for soft handover in transport layer. *IEEE Communications Letters, 8*(3), 189–191.
5. Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., & Kozuka, M. (September, 2007). *Stream control transmission protocol (SCTP) dynamic address reconfiguration*. IETF RFC 5061.
6. Hsieh, H. Y., & Sivakumar, R. (2005). A transport layer approach for achieving aggregate bandwidths on multihomed mobile hosts. *Wireless Networks, 11*(1–2), 99–114.
7. Hsieh, H. Y., Kim, K. H., & Sivakumar, R. (2004). An end-to-end approach for transparent mobility across heterogeneous wireless networks. *Mobile Networks and Applications, 9*(4), 363–378.
8. Ford, A., Raiciu, C., Handley, M., Barre, S., & Iyengar, J. (March, 2011). *Architectural guidelines for multipath TCP development*. IETF RFC 6182.
9. Blanchet, M., Viagenie, & Seite, P. (March 28, 2011). *Multiple interfaces and provisioning domains problem statement*. IETF Internet-Draft, draft-ietf-mif-problem-statement-11 (Work in Progress).
10. Wasserman, M., & Seite, P. (March 28, 2011). *Current practices for multiple interface hosts*. IETF Internet-Draft, draft-ietf-mif-current-practices-09 (Work in Progress).
11. Yousaf, M., & Qayyum, A. (May, 2008). On end-to-end mobility management in 4G heterogeneous wireless networks. In *Proceedings of IEEE 2nd international networking and communication conference (INCC) 2008*, pp. 118–123.
12. Goglin, B., & Prylli, L. (April, 2003). *Performance analysis of remote file system access over high bandwidth local network*. Research Report RR2003-22, LIP, ENS Lyon, France.
13. Henderson, T., & Gurtov, A. (March 14, 2011). *HIP experiment report*. IRTF Internet Draft, draft-irtf-hip-experiment-11 (Work in Progress).
14. Stallings, W. (November 16, 2005). *Cryptography and network security—principles and practices* (4th ed.). Pearson Education Inc., ISBN 0-13-187316-4.
15. Gupta, V., Gupta, S., Chang, S., & Stebila, D. (September, 2002). Performance analysis of elliptic curve cryptography for SSL. In *Proceedings of 1st ACM workshop on wireless security*, pp. 87–94.
16. Daemen, J., & Rijmen, V. (2002). *The design of Rijndael: AES—the advanced encryption standard*. Berlin: Springer. ISBN: 3-540-42580-2.
17. IEEE 802.21-2008. (January, 2009). *Media independent handover services*. IEEE Std. 802.21-2008.
18. Yoo, S. J., Cypher, D., & Golmie, N. (2010). Timely effective handover mechanism in heterogeneous wireless networks. *Wireless Personal Communications, 52*(3), 449–475.
19. Yousaf, M., Bhatti, S., Rehan, M., Qayyum, A., & Malik, S. A. (September 16–19, 2009). An intelligent prediction model for generating LGD trigger of IEEE 802.21 MIH. In *Proceedings of 2009 international conference on intelligent computing (ICIC 2009)*. Springer Verlag Lecture Notes in Computer Science (LNCS).

Author Biographies

Muhammad Yousaf completed M.Sc. Computer Science from the University of Karachi, Karachi, Pakistan in 2003. In 2006, he completed MS Computer Engineering with area of specialization as Computer Networks from Center for Advanced Studies in Engineering (CASE), Islamabad, Pakistan. During his studies at CASE, Islamabad, he won the Talented Student Award, twice, on his academic performance in 2004 and 2006. He is a Ph.D. scholar in Computer Engineering at CASE, Islamabad, Pakistan. He completed a research project titled as "End-to-End Mobility Management Framework (EMF) for Multihomed Mobile Devices" funded by National ICT R&D Fund, Ministry of Information Technology, Government of Pakistan from 2008 to 2010. As ISOC Fellow, he participated in 76th IETF Meeting held in Hiroshima, Japan in November 2009. He joined Riphah International University (RIU), Islamabad, Pakistan as Assistant Professor in 2010. His research interests include mobility management and security issues in heterogeneous mobile networks.

Amir Qayyum graduated in electrical engineering, in 1991 from the University of Engineering and Technology, Lahore, Pakistan. In 1995, he obtained Masters in Computer Engineering from E.S.I.M., France and in 1996, he obtained D.E.A. in Parallel Computer Architectures from University of Paris-Sud, France. He received the Ph.D. degree also from University of Paris-Sud, France in 2000. From 1996 to 2000, he was also working as research student at INRIA Rocquencourt, France. From 2001 to 2003, he was the team lead at Avaz Networks Inc. for the development of complete TCP/IP stack for RISC based packet processor. He served as a Project Director of many R&D projects. Currently, he is a professor and head of Electronic Engineering Department at M. A. Jinnah University, Islamabad, Pakistan. His research activities are focused on MANETs/VANETs, WMN, QoS and mobility management protocols.

Shahzad A. Malik is a Professor at the Department of Electrical Engineering, COMSATS Institute of Information Technology, Islamabad, Pakistan. He holds a B.Sc. in Electrical Engineering from University of Engineering and Technology (UET), Lahore, Pakistan and M.S. and Ph.D. in Telecommunications from ENST, Paris, France. His research interests include wireless and mobile networks, integration of heterogeneous wireless networks, MAC and transport layer protocols for wireless networks, QoS and performance analysis of networks. He has published papers in several areas including radio resource allocation, mobility management, performance and capacity evaluation and quality of service provisioning in wireless networks. He has taught courses on wireless communication, mobile networks, computer networks, digital communications and performance analysis of computer networks.