Wireless Bonding for Maximizing Throughput in Multi-Radio Mesh Networks

Sung-Ho Kim
Graduate school of Information and Communication
Ajou University, Suwon, Republic of Korea
likeleon@ajou.ac.kr

Young-Bae Ko
Graduate school of Information and Communication
Ajou University, Suwon, Republic of Korea
youngko@ajou.ac.kr

***Abstract*—To enhance the per node throughput, mesh nodes in wireless mesh networks can be equipped with multiple network interfaces (NIC). In this paper, we propose a new multi-interface equipped architecture, named "Wireless Bonding (WBond)", which enables each node to fully utilize all its available interfaces, resulting in increased throughput. The proposed WBond provides a method for exposing a single virtual interface by bonding multiple NICs. When a node needs to transfer a packet, WBond selects one among the multiple interfaces depending upon the channel quality assigned to each interface. A novel method for measuring channel quality, called as "ChanQual", is presented and used by the WBond with high accuracy and no additional overhead. Comparison studies using *ns-2* simulations show that our WBond scheme can achieve the most reliable and highest performance (i.e., network throughput) compared to traditional multi-interface architectures.**

Keywords-wireless mesh networks; multi-interface bonding

I. Introduction

A wireless mesh network is a distributed and reliable wireless communication infrastructure consisting of mesh nodes equipping a single or multiple radio transceiver(s) to connect with each other. The uprising interest in this cooperative mesh networking sets the stage for the new generation of high speed, low cost and extensible wireless networks. As an example, a multi-hop wireless mesh network that shall support either as a wireless backbone or last-hop access networks is looming around, taking its place in campus, community and enterprise networks [1]. In this context, the IEEE 802.11 standard has set up the new task group "s", that is 802.11s [2], for adopting the concept of mesh networking.

Since a wireless mesh network mainly aims to serve as a wireless backbone by providing high-speed connectivity for generic mobile clients, how to improve its overall throughput is a very critical problem. One of the possible solutions and maybe the simplest one would be to have fixed mesh nodes equip more than one network interface card (NIC)[1]. It can be easily expected that utilizing multiple NICs can extend the maximum throughput that a particular node can provide. However, we observe that the current IP-based network architecture has a fundamental limitation with the idea of using multi-interfaces. Note that, with the IP-based mechanism like the Internet, any particular network entity becomes identified by an IP address, which is uniquely assigned to a network interface attached to a node. Hence, each network entity identified by other nodes is not the node itself but the interface of that node. Keeping this in mind, let us assume that some node having multiple interfaces wants to send packets generated by its FTP application to one of its neighbors. Although the multiple interfaces are available to use at once and hence the per node throughput can be maximized, this is not possible because only one link identified by a pair of source and destination IP addresses can be used by one particular application at a certain time. That is, the application must select only one link among many available ones, causing the throughput degradation.

In this paper, we propose an efficient way of improving network throughput by enabling each node to fully utilize all its available interfaces. The proposed scheme, called as, "Wireless Bonding (WBond)", positions itself between the IP layer and the layer 2. Being motivated by the existing link aggregation technique so called Ethernet Bonding for wired networking environments [3], our scheme provides a method for exposing a single virtual interface by bonding multiple NICs. The overall operations of our WBond technique can be summarized as follows: First, just like the Ethernet bonding, the proposed WBond layer trunks multiple interfaces together and makes it hidden from the IP layer. Secondly, it distributes incoming packets from the IP layer into any available interfaces. Unlike the traditional Ethernet bonding, the amount of packets to be assigned to one particular NIC will be dependent upon its estimated channel quality. A newly proposed method, named *ChanQual (Channel Quality)*, is used for measuring channel status.

The rest of paper is organized as follows. In Section II, a brief explanation of the Ethernet Bonding, and related works about multi-interface architectures will be provided. Section III describes the proposed WBond and ChanQual techniques. Simulation results in section IV will show that WBond gives the highest network performance, even in the worst cases, compared to single interface case and other multi-interface cases. Section V then concludes the paper.

[1] We use the terms *network interface*, *network interface card*, and *network interface controller (NIC)* interchangeably.

This work was in part supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2006-C1090-0602-0011), and by grant No. R01-2006-000-10556-0 from the Basic Research Program of the Korea Science & Engineering Foundation.

II. RELATED WORKS

The Linux Ethernet Bonding driver [3] is the software for aggregating multiple Ethernet interfaces into a single logical interface. The behavior of the bonded interfaces depends upon the mode. For instance, with the "balance-rr (round robin)" mode, a number of packets generated by one application is distributed to the bonded multiple interfaces in a round robin fashion. When employed in this fashion, the balance-rr mode allows individual connections between two hosts to effectively utilize greater than one interface's bandwidth. That is, if both a sending node and a receiving node are equipped with two interfaces, odd-ordered incoming packets from the IP layer will be assigned to the one interface while even-ordered packets will be assigned to the other. As a result, the receiving node may receive packets two times faster than the case when it has only one interface.

Although the performance enhancement by using the Linux Ethernet Bonding scheme is limited by the number of network interfaces Linux supports, it is known to be the best method for maximizing the network throughput. However, it may not work well in wireless environments because it is mainly proposed and used for wired networks. Unlike wired links, the status of wireless medium varies dynamically. Therefore in this paper, we argue that quality information of wireless channel should be taken into account when only one active interface (among bonded multi-interfaces) is chosen for an actual packet transfer via wireless links. Remind that, the current Linux bonding with the balance-rr mode transmits packets in a round robin fashion over the available interfaces, without any notion of channel status. Our argument will be proven later in Section IV, with the simulation results of the balance-rr mode based Linux Ethernet bonding scheme in wireless networking environments.

In recent years, several researchers have studied on utilizing multiple interfaces and multiple channels [4-9]. The closest work to ours is the MUP (Multi-radio Unification Protocol) presented in [9]. Similar to our WBond, MUP also coordinates the operation of multiple network interface cards for the purpose of improving performance. In order to find the best quality interface, RTT (round trip time) is designed here. That is, for measuring channel quality of each link, a MUP-enabled node issues a *probe* message to all its neighbors, expecting each neighbor to respond with an *acknowledgement* packet where the timestamp is written. Upon being acknowledged, the node previously sending the probe message can calculate the RTT value, treating it as the quality of that interface towards the responding neighbor. Of course, RTT is likely to provide relatively precise information of the current channel in use (at least in sparse network). However, RTT seems to require each node to probing with all its neighbors, resulting in significant overhead to the network.

Recent research has shown that using the minimum hop-count metric is not sufficient to decide the quality of a route. Hence, to figure out a better routing metric, some work has been done with the use of link quality and proposed a way of estimating it more accurately [10, 11, 12].

III. WIRELESS BONDING

The main goal of the proposed Wireless Bonding (WBond) technique is to maximize throughput when a node has multiple interfaces. Assuming the two nodes that have a set of interfaces whose channel currently in use is same, our WBond layer forwards incoming packets from the upper layer like IP into one of all available interfaces. An interface with the better channel quality can be selected with higher probability. The operations of the WBond layer can be summarized; **neighbor and available interface list maintenance, channel quality measurement, and packet distribution.** Each operation will be explained more deeply in each of the following subsections.

Figure 1 compares a traditional multi-interface architecture and the WBond inserted architecture. As seen in the figure, the proposed WBond module essentially sits below the IP layer and above the NIC layer. As explained earlier, it exposes one virtual interface by bonding and managing multiple network interfaces. Thus, a WBond-enabled node is configured with only one IP address for multiple interfaces and applications.

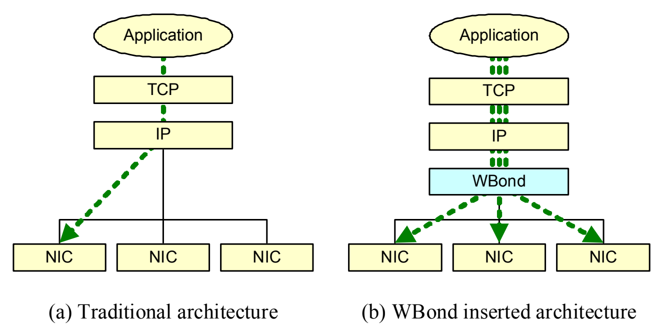

Figure 1. *A comparison between a traditional multi-interface architecture and our WBonding layer inserted architecture. (Packets represented by a dotted line are generated by some application and pass through a network stack. The proposed WBond layer distributes these packets to all three interfaces so the maximum throughput becomes three times higher than the one achiened in the traditional architecture.)*

In the current IP-based network architecture, the maximum throughput of one application is limited to the throughput of one interface and a busy interface can be selected to transmit traffic even if other interfaces are not busy. Moreover, this situation can go on unless the connection is lost. MUP intelligently selects a best-quality interface every certain period but it also has a limitation in the sense that its maximum throughput is bounded by the maximum throughput of that interface. Instead, our WBond's maximum throughput is equal to that node's maximum throughput, i.e.., the throughput achieved by using all available interfaces.

A. *Neighbor and Available Interface List Maintenance*

The WBond layer of each node maintains a neighbor table consisting of its neighbor entries to identify the neighbor and to establish the available interface list which can be used to communicate with the neighbor. An entry in the table contains four different fields: the neighbor node's IP address, the list of available interfaces, the update timer, and retry count. See Table I, representing the structure of neighbor entries.

TABLE I. NEIGHBOR ENTRY STRUCTURE

Field	Description
Address	IP address of this neighbor
Interface list	Available interface list for communicating with this neighbor
Update timer	Timer for updating this entry
Retry count	Number of failure while updating this entry

Each neighbor entry is created when a node receives a packet from its previously unknown neighbor. The node registers the interface which the packet comes from on the *available interface list*. Then the node generates an address resolution protocol (ARP) [13] request packet with a target IP address set to this unknown neighbor's IP and broadcasts it using all its interfaces. The destined neighbor will eventually respond by unicasting an ARP reply packet using all the interfaces the previous ARP request came from. Such operations of the modified ARP protocol guarantee that a pair of two nodes can make a list of available interfaces for each other. Also, it can allow WBond layer-enabled nodes to be backward compatible with legacy nodes.

As seen in Table 1, the *update timer* for each entry is used to update the neighbor entry. The timer starts when a new entry is created and becomes reset whenever it receives a packet from the neighbor. If it expires with no packet reception from the neighbor for a while, the node will re-perform the ARP operations described above to verify whether the neighbor is still alive or not. Another field of *retry count* in the table is designed to increase by one when the node fails to hear from the neighbor. If it exceeds a predefined threshold, the node presumes that the neighbor is not available anymore so the entry is removed from the table.

B. *Channel Quality Measurement*

ChanQual is a newly proposed method for a channel quality measurement, which utilizes device statistics information. Most of recent IEEE 802.11 WLAN interface cards support the function providing some informative communication statistics. Although the actual values of the statistics may differ based on the driver and hardware specifics, they mostly have the similar meanings. We have analyzed a few well-known devices [14-17] and selected some common statistics as follows.

- rxpkt: Number of packets received; To be updated just before passing the received frame to the network stack.
- rxerr: Number of errors occurred while receiving packets; To be updated when fail to read a frame from the card, discover bad CRC on received frame, or receive an oversize frame.
- retry: Number of packets the hardware failed to deliver; To be updated when the retry count becomes greater than threshold while attempting to send RTS or DATA frame.

The WBond layer enables a promiscuous mode of all bonded interfaces and gathers the statistics information every T seconds. We will denote these statistics values by adding T to the end of each statistical notation, like $rxpkt_T$. Now, we can measure the channel quality using the following equation. As the channel becomes busy, the values become higher, meaning the quality of that channel is low. As the errors occur more frequently, the value may increase as well.

$$QUAL = K_1 \times rxpkt_T{}^S + K_2 \times rxerr_T{}^S + K_3 \times retry_T{}^S \qquad (1)$$

In (1), the values of K are constant giving different weight to each correspondent value. S is also the constant value to increase the difference of the calculated channel quality values. The calculated result is a value representing the channel status of the previous T seconds. To prevent the value from being fluctuated we incorporate it to the smoothed value, *SQUAL*, using (2) where α is a real number between 0 and 1. The result *SQUAL* is used as the channel quality.

$$SQUAL = \alpha \times QUAL + (1 - \alpha) \times SQUAL \qquad (2)$$

The proposed ChanQual has three advantages compared to the prior channel measurement methods. First, it does not add any overhead to the network. Second, it is totally independent from its neighbors' status. The ChanQual operates only with local information so it may not be affected by other nodes' failure. Third, it can reflect the quality changes of small duration. The only delay caused by ChanQual is a processing delay for calculating equations and for gathering device statistics. Therefore, in ChanQual, it is possible to set the channel measurement period T with extremely small time to reflect the quickly changing network environment. Fourth, it is not affected by the size of packet queue so the measurement accuracy can be improved. In a RTT-like method requiring a packet exchange, the queue size is likely to make more impact on the measured value.

C. *Packet Distribution*

When a packet arrives from the upper layer, our WBond selects one interface and forwards the packet to it. WBond looks at the next hop address, searches the neighbor entry having that address, and then selects one interface from the interface list of the entry. The criterion for selecting one interface is the quality of channel the interface uses. The basic idea is to let an interface with better quality channel handle more packets. Again, it can be measured using the ChanQual method, by which any network-related effects like traffic level is never disturbed. Therefore, when an interface selection is needed, the channel quality values are in available state so they can be utilized without any extra delay.

In WBond, the probability that one interface is selected for a packet transfer is proportional to the channel quality of that interface. However, if the quality of one interface is more than three times worse than the best interface quality, that interface is excluded from the interface selection phase. This is because the probability that packets arrive in out-of-order at the receiver will increase as the quality difference between network interfaces becomes higher.

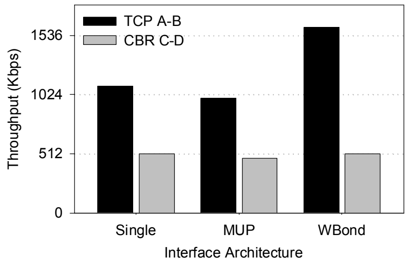

Figure 2. *Throughput of TCP and CBR*

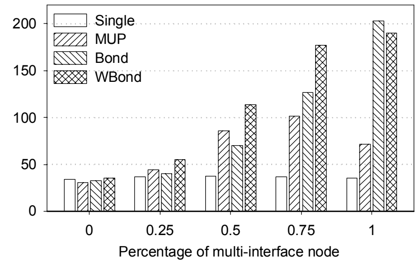

Figure 3. *TCP throughput varying the proportion of multi-interface nodes*

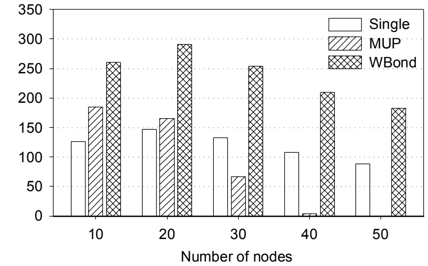

Figure 4. *TCP Throughput varying the number of nodes in random topology*

IV. PERFORMANCE EVALUATIONS

We performed a simulation study using ns-2 [18]. ARP is modified to give our WBond an ability to handle ARP functions but there is no modification in IP and IEEE 802.11 MAC source codes. The configurable parameters of WBond are set to the appropriate values; the update timer of each neighbor entry expires after 5 seconds from the last update and the threshold value of retry count is set to 3. To find the recommendable values of ChanQual parameters, we performed a number of experiments and the results are listed in Table II. The retry parameter increases by one when RTS or DATA retransmission count reaches the threshold so NIC gives up its trial for a packet transfer. It represents how poor the channel quality is. In our experiments, K3 is set to a higher value than those of K1 and K2 to give more weight to the retry parameter.

TABLE II. CHANQUAL PARAMETER SETTINGS

Parameter	***K1***	***K2***	***K3***	***S***	***α***	***T***
Value	1	1	10	2	0.1	0.1sec

As a comparison target, we have also implemented the MUP protocol [9]. The primary ns-2 parameters are the followings. The MAC data rate is 2Mbps, the radio propagation model is two-ray ground reflection model, the antenna type is omnidirectional and the ad hoc on-demand distance vector routing (AODV) [19] is used as a routing protocol. The transmission range of each node is about 250m.

A. *Preliminary Results with a Simple Topology*

In order to see how WBond utilizes all available network bandwidth, a simple scenario is designed with four nodes which are within the communication range of each other; nodes A and B are assumed to equip two interfaces (with two different channels numbered 1 and 2, respectively), whereas nodes C and D are to a single interface (with the channel numbered 2). We assume that node A transmits FTP traffic to node B, while node C transmits CBR traffic whose sending rate is 512 Kbps to node D.

Figure 2 shows the results of various interface architectures. Note that, in single interface architecture, nodes A and B are also assumed to use only one interface (with channel no. 1). As shown in the figure, the TCP throughput between nodes A and B using channel numbered 1 is about 1100 Kbps, the maximum throughput of one link with the current ns-2 settings. The CBR throughput between nodes C and D using channel numbered 2 is 512 Kbps. With the MUP, both the TCP and CBR throughput are slightly decreased compared to the single interface case. This is mainly due to the fact that RTT results in more overhead to the network. Another reason worth to denote is related to the RTT channel selection algorithm presented in [9]. Sometimes, channel 2 can be selected as a preferred channel. In this case, both FTP and CBR traffics are sent via channel 2 while channel 1 is free.

On the contrary, WBond utilizes the available bandwidth of both channel 1 and channel 2. The *SQUAL* value of channel 2 is larger than that of channel 1 because four nodes compete for channel 2 but only two nodes compete for channel 1. With the measured *SQUAL* values, node A transmits more FTP packets using channel 1. As a result, TCP throughput increases to over 1610 Kbps while CBR throughput remains to 512 Kbps.

B. *The Effect of NIC Heterogenity with a Grid Topology*

In the heterogeneous network where nodes have different number of interfaces, the number of out-of-order packets can be increased, resulting in the performance degradation. To investigate the impact of such heterogeneity in number of NICs, we design a bit complicated scenario with 4x4 grid topology having 16 nodes, spaced by 200m from each other. We vary the proportion of the legacy nodes equipping a single NIC among 16 nodes. Clearly, no schemes for utilizing multi-interfaces such as Linux Bond, MUP and our WBond can be applied for these single-interface equipped nodes. All other nodes in the topology are assumed with multiple interfaces (2 NICs, here).

The Linux Bonding is newly added to the comparison target with named Bond. In Bond case, neighbor classification and maintenance procedure are remained but ChanQual does not work. The operation mode is balance-rr. One FTP connection is placed where the source is the top-left node and the destination is the bottom-right node in the topology. Four Pareto traffics provided by ns-2 are also placed to add some background traffic to the network.

Results with TCP throughput are shown in Figure 3. If the ratio of multi-interface nodes is equal to zero (i.e., all nodes have a single NIC), the results are almost same for all the cases. However, as the proportion of two-interface equipped nodes increases the performance becomes differ with each other, even though all the schemes except for the single case have the increasing shape in their graphs. The throughput of MUP also increases but the increasing degree is relatively low compared to the Bond and WBond. Especially, its performance decreases drastically when the ratio is equal to one (i.e., all nodes have two NICs). We believe that this is because the overhead of RTT method increases as the number of MUP-enabled neighbors increase.

The throughput of WBond is always the best, except when the ratio of multi-interface nodes is equal to one. The simple Bond shows the highest TCP throughput in this case when the occurrence of out-of-order packets might not be so frequent. However, as the ratio becomes smaller and therefore the degree of heterogeneity becomes larger, the Bond's performance degradation is more significant than that of our WBond.

C. *The Effect of Node Density with a Random Topology*

To investigate the performance in more realistic network, a random topology is considered in this scenario. We vary the number of nodes from 10 to 50 in 1000m x 1000m area. Every node is assumed to have two interfaces. There is one FTP and no background traffics. The source and destination nodes are selected randomly, but are forced to be apart long enough so an average length of a route between them is at least more than one hop.

Figure 4 proves the fact that, with increasing node density, the throughput generally decreases in all cases. Nevertheless, the WBond shows the best performance irrespective of node density. The performance of MUP seems to be better than the single case when the number of nodes is relatively small -- 10 or 20 nodes in our case. However, as the number of nodes increases more and more, its performance becomes worse and even lower than the single case. Based on the analysis of the trace files, we could conclude that such a degradation of MUP is due to the overhead of RTT probing. As can be seen in the figure with 40 and 50 nodes, it is even hard to find a route due to the reiteration of broadcast flood initiated by route discovery and the RTT probing packets. We found that even the source node succeeds to find a route the data packets hardly reach to the destinations due to massive RTT overhead.

V. CONCLUSIONS

In this paper we proposed a new multi-interface architecture, named "WBond". It distributes incoming packets from upper layers to each of interfaces based on the channel qualities of those interfaces. This operation enables an application to utilize all the available interfaces so the overall network throughput can be maximized. To measure the channel quality, we also proposed a new channel quality measurement method, *ChanQual*. It measures the channel quality by utilizing the statistics the network device provides.

Using various scenarios with different topologies, we have shown that the proposed WBond with ChanQual achieves the highest throughput in most scenario settings. Even in the heterogeneous networking environments with nodes having different number of interfaces and burst background traffics, the WBond shows about 50~450% performance improvement compared to the single interface case and about 20~150% improvement compared to the MUP. In the last simulation with more realistic random topology, our WBond still achieves the maximum throughput regardless of the node density.

In this paper, due to the implementation limits, we could not investigate the performance of MUP over testbed. As the future work, we will implement WBond and ChanQual in the testbed and investigate the performances. The integration of routing protocol and WBond inserted architecture for utilizing the bonded virtual interface in routing layer will also be researched.

REFERENCES

[1] Wireless Communities. http://www.personaltelco.net/index.cgi/WirelessCommunities

[2] Amendment to the current 802.11 standard to provide Extended Service Set Mesh Networking. http://standards.ieee.org/board/nes/projects/802-11s.pdf.

[3] Linux Ethernet Bonding Driver mini-howto http://www.kernel.org/pub/linux/kernel/people/marcelo/linux-2.4/Documentation/networking/bonding.txt.

[4] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," *ACM Mobihoc*, 2004.

[5] A. Raniwala and T. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," *IEEE INFOCOM*, 2005.

[6] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," *ACM MobiCom*, 2005.

[7] R. Draves, J. Padhye, and B. Zill, "Routing in Multi-Radio, Multi-Hop Wireless Mesh Networks," in *ACM MobiCom*, 2004.

[8] P. Bahl, A. Adya, J. Padhye, and A. Wolman, "Reconsidering Wireless Systems with Multiple Radios," *ACM Computing Communications Review*, July 2004.

[9] A. Adya, P. Bahl, J. Padhye, A. Wolman, and L. Zhou, "A Multi-Radio Unification Protocol for IEEE 802.11 Wireless Networks," in *IEEE International Conference on Broadband Networks (Broadnets)*, 2004.

[10] R. Draves, J. Padhye, and B. Zill, "Comparisons of Routing Metrics for Static Multi-Hop Wireless Networks," *ACM Annual Conf. Special Interest Group on Data Communication (SIGCOMM)*, Aug.2004, pp. 133-44.

[11] S. Keshav, "A Control-theoretic approach to flow control," In *SIGCOMM*, September, 1991.

[12] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "High-throughput path metric for multi-hop wireless routing," in *MobiCom*, September, 2003.

[13] D. C. Plummer, "Ethernet address resolution protocol: Or converting network protocol addresses to 48 bit Ethernet address for transmission on Ethernet hardware," IETF, Request for comments (Standard) RFC826, 1982.

[14] Atheros, http://www.atheros.com.

[15] MadWifi, http://madwifi.org.

[16] Orinoco, http://www.orinoco.org.

[17] Atmel, http://www.atmel.com.

[18] UCB/LBNL/VINT Network simulator (ns) – Version 2. http://www.isi.edu/nsnam/ns.

[19] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (aodv) routing", IETF, Request for comments (Standard) RFC3561, 2003.