DESIGN OF 802.16 WIMAX BASED RADIO ACCESS NETWORK

Hung-Yu Wei
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
hywei@ntu.edu.tw

Samrat Ganguly
IP Network and Distributed Systems
NEC Laboratories America
Princeton, NJ, USA
samrat@nec-labs.com

ABSTRACT

In this paper, we present the design of the WiMAX based radio access network to provide data coverage to 802.16 enabled devices. The design provides a multihop solution to provide wide area coverage to users and targets maximizing the radio resource utilization. We propose efficient route construction, data scheduling for increased utilization of the backhaul network and base station selection algorithm for end users for efficient load balancing. Through extensive solution, we should attain the benefits by each of the proposed design methods.

I. INTRODUCTION

Recently, WiMAX based on IEEE 802.16 standard is emerging as a cost-effective solution to providing last mile broadband access to residential and business community. The ongoing evolution of WiMAX technology might expand to provide the wide area wireless data coverage to roaming users as an alternative to current HSDPA/EV-DO based 3G solution. This is possible by having 802.16e based WiMAX-enabled portable devices such smartphones, PDAs and notebooks. In this paper, we consider such an alternative radio access network (RAN) architecture as shown in Figure 1.

In this architecture, operators can deploy WiMAX base stations (BS) with both Line of Sight (LOS) and Non-Line of Sight (NLOS) capabilities. The NLOS link is used in point-to-multinode (PMP) mode by the BS to provide data access to the roaming client or subscriber stations (SS). The high bandwidth LOS links in point-to-point (PtP) mode can create the wireless backhaul to connect the WiMAX base station to the gateway. Such an architecture provides a flat IP-based RAN [2] resulting in higher cost saving, easier to upgrade and reduced complexities in terms of management. For users, the RAN architecture can provide higher bandwidth and support of multiple applications including data, VoIP and media streaming [4].

In designing the WiMAX based RAN, we propose efficient mechanisms for maximizing the supported data throughput to end users. We assume that the placement of the WiMAX base stations and the inter-BS links are already provided. Authors in [3] considers the planning of inter-BS links for supporting 3G traffic. The supported throughput depends upon the capacity of the backhaul network. In order to efficiently utilize the backhaul network, we propose creation of interference-aware multiple path from the gateway to each WiMAX BS. Use of multiple paths based on demand from each BS achieves better spatial diversity compared to single path solution as proposed in [7]. Given, the set of paths connecting the WiMAX BS, we propose efficient scheduling of data along paths which interferes among each other. The scheduling scheme is based on proper pacing of the data packets at the gateway node. Our proposed scheduling algorithm operates at the gateway node and is centralized as all traffic finally emerges through the gateway node. This is as opposed to distributed scheduling algorithm as proposed in [6] for 802.16 mesh.

Apart from considering the backhaul inter-BS WiMAX mesh network we also consider the BS-to-SS PMP links and use the certain flexibilities that can be attained in the proposed RAN (Figure 1). In particular, we consider the scenario where each client or SS can be covered by more than one WiMAX BS. Such a scenario can not only improve the channel quality to the user but also allow efficient load balancing on the backhaul mesh network. Specifically, we determine which WiMAX base station and which multihop backhaul route should be selected to serve a new join or handoff node. The selection algorithm also results in better utilization of the PMP links from BS to the SS.

II. ROUTE CONSTRUCTION IN MESH BACKHAUL

The route construction algorithm computes multiple routes from the source node S toward the destination node D. The aim of this route construction algorithm is to create maximum disjoint routes that are capable of supporting multiple parallel flows. The route construction algorithm considers capacity constraints and interference constraints and strives to achieve high end-to-end throughput between S and D.

In Figure 2, we illustrate a route construction procedure with a simplified example. We try to establish multiple routes from A to D. The number shown next to a link is the bandwidth of such a link. The interference links are circled together. In Figure 3, we first construct the route A-C-D and updates the bandwidth usage. Not only the link
bandwidth values of selected links are updated, all the interfered routes are updated. Although A-B-D was originally a high-bandwidth route than A-E-F-G-D in Figure 2, it will not be a better route selection due to the interference. In Figure 3, the second route is constructed. After these two routes are constructed, the route selection algorithm will determine which route will be use for a give traffic flow in the next stage.

The following notations will be used in this section.

- \( L_{ij} \) - directional link from node \( i \) to node \( j \)
- \( r_{ij} \) - physical link rate of link \( L_{ij} \)
- \( t_{ij} \) - normalized residual time of link \( L_{ij} \)
- \( U_T \) - temporary radio resource usage on route \( T \).

The route construction algorithm first adds the source node \( S \), which is also the Gateway of wireless mesh backbone. Until there is not more resource to create a route, we compute a route \( \Gamma_D \) to destination node \( D \) in each route construction iteration, and deduct the wireless resource usage from the available wireless resource.

During an route construction iteration, we have several node addition iterations (line 8 – 22) that adds one node to a routing tree in each iteration. We will first initialize the tree \( T \) and add one node each time to the tree \( T \) and the set of already added nodes \( InRoute \). Gateway node, which is the source node \( S \), is the first node to be added.

One important metric \( U_T(n) \) is the temporary radio resource usage along the multihop route from \( S \) toward node \( n \) on \( T \). Since our wireless mesh network model considers interference and has different link rates at different wireless links, it is difficult to directly measure radio resource usage in bits/second across spatial locations. We opt for calculating radio resource in normalized time \( t_{ij} \) and \( \tau_{ij} \) while considering interfered links and then converting to link capacity in bits/second \( R_{ij} \) at different spatial locations.

The main route selection criteria to minimize the radio resource usage (in terms of normalized time) along the multihop route. The notation \( U_T(n) \) represents the multihop radio resource cost from \( S \) to node \( n \) using routing tree \( T \).

The notation \( \succ \) is used to indicate interference between two links. \( L_{ij} \succ L_{uw} \) implies that link \( L_{ij} \) interferes with link \( L_{uw} \) (when \( L_{uv} \) is active, \( L_{ij} \) cannot be active). We also define \( L_{ij} \succ L_{ij} \). The time of a bit transmitted over a link \( L_{ij} \) is \( 1/R_{ij} \). As \( L_{ij} \succ L_{uw} \), \( L_{uw} \) cannot be active during this \( 1/R_{ij} \) time duration. The wireless radio resource cost of transmitting a bit over \( L_{ij} \) will be \( 1/R_{ij} \) on \( L_{ij} \) plus \( 1/R_{ij} \) on the interfered link \( L_{uw} \).

At the beginning of node addition iteration, nodes already in the route examine the neighboring nodes and pick a candidate child node with the minimum one-hop transmission cost (line 9 – 12). The candidate child node, which has the minimum multihop radio resource transmission cost, is added to the routing tree. The multihop transmission cost values along the route will be updated (line 16 – 19). The node addition iterations continue until \( \Gamma_D \), a route from \( S \) to \( D \), is derived.

After the new route \( \Gamma_D \) is given, the cost at \( L_{uw} \) for transmitting one bit over route \( \Gamma_D \), \( \tau_{uv} \), is computed (line 24 – 26). The ratios of available link resource and transmission cost, \( \phi_{ij} \), are calculated to determine the bottleneck link, which is the link with minimum \( \phi_{ij} \) value. Then we will allocate the maximum possible wireless resource \( \Phi \) along the route. After we update the available resource and add current route \( \Gamma_D \) to the set of routes, we will continue calculating the next available route.

III. DATA SCHEDULING AT GATEWAY NODE

In the present architecture, we are dealing with two independent type of resources: multihop wireless mesh backbone and the Point-to-Multipoint access points. Both of the above types of resource have their own capacity constraints and have to be used accordingly. Therefore, the proposed scheduling policy has two schedulers: the Gateway scheduler and the Access Point scheduler. The Gateway scheduler is decides how to schedule and dispatch the received packets from the outside network domain into this mesh backbone networks, which acts as a feed-forward network connecting the MNs. The Access Point scheduler decides how to schedule the downlink data frames over the NLOS WiMAX channel. The Gateway scheduler has per AP buffer installed while the AP scheduler has per MN buffer installed.

We next describe the scheduling policy for both the schedulers.

A. Gateway scheduling

The scheduling mechanism at Gateway is employed to serve the per AP buffer as described above. The buffer is maintained as FIFO queue to ensure in order delivery of data.

At Gateway, we assume that the scheduler schedules a block of data to be sent toward a multihop route.

One should note that the scheduling criteria used should not lead to starving of any buffer since buffer buildup will inhibit Gateway for forwarding data for the given AP. The
Input: 
- \( N \): Set of Nodes \(- S \) is the source; \( D \) is the destination
- \( r_{ij} \): physical link rate of link \( L_{ij} \) \(- i, j \in N \)

Output: 
- \( T \): Set of Routes

Algorithm:
1. \( T \leftarrow \phi \)  
2. \( t_{ij} \leftarrow 1 \) \( \forall i, j \)  
3. Do  
4. \( \Psi \leftarrow \phi \) \( // \) Initialize tree \( \Psi \)  
5. \( \text{InRoute} \leftarrow \{S\} \)  
6. \( U \gamma(S) \leftarrow 0 \)  
7. While \( D \notin \text{InRoute} \) do  
8. For each \( n \in \text{InRoute} \) do  
  a. \( M_n \leftarrow \min_j \{1/r_{nj}\} \) \( s.t. j \notin \text{InRoute} \)  
  b. \( n \rightarrow \text{Node with } M_n \)  
9. End For  
10. End While  

\( \text{InRoute} \rightarrow \text{InRoute} \cup y \)  

End While  

\( \Gamma_D \rightarrow \text{Route from } S \) to \( D \) in tree \( \Psi \)  

For each link \( L_{uv} \in \Gamma_D \) do  
14. \( \tau_{uv} \leftarrow \sum_{l_{ij} \in L_{uv}} t_{ij}, l_{ij} \in \Gamma_D \{1/r_{ij}\} \)  
15. End For  

For each link \( L_{ij} \in \Gamma_D \) do  
16. \( \phi_{ij} \leftarrow t_{ij}/\tau_{ij} \)  
17. End For  

\( \Phi \leftarrow \min_{(i,j)} \{\phi_{ij}\} \)  
18. \( t_{ij} \leftarrow t_{ij} - \Phi \cdot \tau_{ij} \) \( \forall L_{ij} \in \Gamma_D \)  
19. \( T \leftarrow T \cup \Gamma_D \)  
20. End Do  
21. Return \( T \)

Figure 4: Algorithm to create multiple routes

admitted traffic load of access point \( y \) is \( L_y \), while the backlog to \( A_P \) is \( D_y \). Therefore, the scheduler selects the buffer with the maximum weighted buffer length \( (D_y/L_y) \), and schedules the dispatch for block of data from the buffer.

The goal of the scheduler design at Gateway is to maximize the data throughput through the mesh backbone. While the proposed scheduling policy will have equal weighted buffer levels, it will not use the Gateway resource efficiently. When serving multiple access points, there can be blocking at certain edges lowering the throughput. If two paths to two destinations are non-interfering, there is clearly no problem. However, for two interfering paths, the blocking can be resolved if two paths are served with a time gap \( \Delta \) between them. The time gap with ensure that data block on the first path has already been pushed to the first AP creating no blocking when the next packet is served.

Next we present the algorithm to find \( \Delta \) for any two paths in the mesh network. Based on finding the \( \Delta \), we subsequently propose a pipeline scheduling in dispatching data blocks from Gateway. Note that minimum value of \( \Delta \) is \( T \), which denotes the time to completely transmit a single block over a link, since the Gateway cannot transmit simultaneously to more than one destination.

B. Delay between two paths

Given two paths \( P_0 \) and \( P_1 \), our objective is to find the minimum delay value \( \Delta \) such that dispatching a block on \( P_1 \) does not lead to interference with the block sent on \( P_0 \). There are several intermediate relay hops along \( P_0 \) and \( P_1 \). We define a conflict node pair \( \xi(n_0, n_1) \) as one relay node \( n_0 \) on \( P_0 \) and the other relay node \( n_1 \), which will interfere with each other when they are both transmitting.

Consider a conflict node pair \( \xi(n_0, n_1) \) for path pair \( P_0 \) and \( P_1 \). For block \( b_0 \) sent on \( P_0 \), let the transmission from relay node \( n_0 \) starts at \( t_{b_0} \), and ends at \( t_{b_0} + \delta_{b_0} \). Similarly, on path \( P_1 \), the block \( b_1 \) will be transmitted between \( t_{b_1} \) and \( t_{b_1} + \delta_{b_1} \) at relay node \( n_1 \). Nodes \( n_0 \) and \( n_1 \) are located within the interference range. We say the conflict node pair \( \xi(n_0, n_1) \) is active when two time intervals \( t^k_{b_0}, t^k_{b_0} + \delta_{b_0} \) and \( t^k_{b_1}, t^k_{b_1} + \delta_{b_1} \) overlap, which implies the transmission at \( n_0 \) and \( n_1 \) are actually interfering with each other and we need to impose a time delay \( \delta_{b} \) to avoid this interference.

We will compute the \( \Delta \) value with iterative steps. The scheduling of path \( P_0 \) is fixed and the scheduling at path \( P_1 \) will be delayed with \( \Delta \) to avoid interference. The initial value of \( \Delta \) is set to zero. When an active conflict pair is found, we compute the necessary delay value \( \delta_{b} \) to resolve the conflict of \( \xi_{ni} \), and then update \( \Delta = \Delta + \delta_{b} \). After completion of iterative steps on all conflict pairs we will get the minimum \( \Delta \), which resolves all possible interference on path \( P_0 \) and \( P_1 \).

\[ \text{Input:} \]
- \( \xi \): the set of conflicting nodes along path \( P_0 \) and \( P_1 \)
- \( t^k_{b_0} \): time that \( n_0 \) start transmission; \( \xi(n_0, n_1) \in \xi; n_0 \in P_0 \)
- \( t^k_{b_0} \): time that \( n_0 \) start transmission; \( \xi(n_0, n_1) \in \xi; n_0 \in P_0 \)
- \( t^k_{b_1} \): time that \( n_1 \) start transmission; \( \xi(n_0, n_1) \in \xi; n_1 \in P_1 \)
- \( t^k_{b_1} \): time that \( n_1 \) start transmission; \( \xi(n_0, n_1) \in \xi; n_1 \in P_1 \)

\[ \text{Output:} \]
- \( \Delta \): Minimum delay to dispatch on conflicting path \( P_0, P_1 \)

Algorithm:
1. \( \Delta \leftarrow 0 \)
2. For each conflicting node pair \( \xi(n_0, n_1) \in \xi \)
   3. If \( (t^k_{b_0} + \Delta) \in (t^k_{b_1}, t^k_{b_1} + \delta_{b_1}) \) or \( (t^k_{b_0} + \Delta) \in (t^k_{b_1}, t^k_{b_1} + \delta_{b_1}) \)
   4. \( \delta \leftarrow t^k_{b_0} + \Delta \)
   5. \( \Delta \leftarrow \Delta + \delta \)
6. End If  
7. End For  
8. Return \( \Delta \)

Figure 5: Algorithm to compute \( \Delta \) for conflicting path pair

C. Pipelined dispatch at Gateway

In the pipelined dispatch scheduling at Gateway, after dispatching current block \( b_i \), the scheduler selects the next block \( b_{i+1} \) from the buffer that has the maximum level. It computes the waiting time, \( \Delta(b_i, b_{i+1}) \), and assigns the dispatch time as \( T_c + \Delta(b_i, b_{i+1}) \) to the block \( b_{i+1} \), where \( T_c \) is
the current time. However, this dispatching rule may result in void (period of no transmission from Gateway) between two dispatches leading to low utilization. To circumvent the above problem, we consider the following void filling approach.

The void is filled iteratively by first selecting block $b_k$ from the buffer $B_k$ which has minimum $\Delta(b_k)$ such that $\Delta(b_k) + \Delta(b_{k+1}) < \Delta(b_k)$, where $\Delta(b_k)$ and $\Delta(b_{k+1})$ is the waiting time required to avoid collision with the blocks $b_k$ and $b_{k+1}$ block respectively. If there are more than one buffer $B_k$ with minimum $\Delta(b_k)$ and satisfying the above constraint, one with more weighted buffer level is selected. This process is repeated until no more buffer can be served in the void.

The above dispatch process is outlined as follows. In the routine, a block from main schedule is denoted as $b^m$ and the block from the void filling schedule is denoted as $b^v$. A dispatch of a block refers to the event corresponding to initiating transmission of a block at the Gateway.

**Algorithm:**

1. On dispatch of a block $b^m$ at current time $T_e$
2. Select block $b_{n+1}^m$ from buffer with max weighted backlog
3. $\Delta = \Delta(b_n^m, b_{n+1}^m)$
4. Schedule $b_{n+1}^m$ at $T_e + \Delta$

/start void filling
5. $\delta = 0$
6. Last block $b_{n}^m$
7. While $\delta < \Delta$
8. For each non-empty buffer $B_k$
9. $b_k^v = $ data block from buffer $B_k$
10. End For
11. $b_k^v = \text{argmin}_{b_k}{\Delta(b_k, b_{k+1})}$
   where $\Delta(b_k, b_{k+1}) \leq \Delta - \delta$ for
12. $\delta = \delta + \Delta(b_k, b_{k+1})$
13. Schedule $b_k^v$ at $T_e + \delta$
14. $b_k^v$
15. End While

Figure 6: Algorithm to schedule data block dispatch

**D. Discussion**

To achieve scalability, the two-tier data dispatching process is applied. The first-tier dispatching decision at Gateway is made based on the destination access point. On the other hand, as data blocks arrive at an access point, the second-tier data dispatching is based on the destination mobile nodes.

Instead of maintaining a uncertain large number of queues, Gateway node only needs to maintain a small fixed number of queues that is equal to the number of access points in this mesh radio access network. In addition, user fairness is conserved with per-user data dispatching at access points.

**IV. ACCESS POINT AND ROUTE SELECTION**

The route construction algorithm computes multiple routes from Gateway to an access point. A mobile node $x$ may be connected to more than one access points. The access point selection algorithm determines which available access point $y$ should serve this mobile node. The route selection algorithm determines the route that packets are sent through Gateway to access point $y$. The access point and route selection algorithm will consider load condition and packet delivery in both mesh backbone and wireless access points.

As a mobile node could connect to multiple access points, the set of candidate serving access points for mobile node $x$ is denoted as $A(x)$. We denote the physical link rate between mobile node $x$ and access point $y$ as $r_y(x)$. The set of access points that a mobile node could attach to is denoted as $A(x)$. The arriving radio resource of the access point $y$ can be given in normalized time $\varphi_y$, which is a real number between 0 and 1. When no mobile node is served by access point $y$, $\varphi_y$ is set to 1. When no radio resource is available at access point $y$, $\varphi_y$ is 0. The residual capacity concept will be applied to access points. Since the link rates from an access points to different mobile nodes depends on distance and radio propagation condition, the residual capacity value is different for different mobile nodes and is given as $C_y(x) = \varphi_y \cdot r_y(x)$. The expected delay for a bit to be delivered from AP $y$ to MN $x$ is $1/C_y(x)$.

In the mesh backbone, the routing selection procedure is based on the concept of the estimated dispatching delay from Gateway to the serving access point. Given the fixed wireless network topology, $\Delta$ between a pair of routes could be computed by the algorithm in the previous section. When there is currently no active serving route and access point, we need to initialize the serving routes and access points (line 1 5). A serving route and serving access point with minimal expected delay time from AP to MN plus the estimated delay time from GW to AP. Without considering background traffic, the expected dispatching delay from GW to AP $y$ on route $\Gamma_y(k)$ is $\Delta(\Gamma_y(k), \Gamma_y(k))$. The expected delay from AP to MN is $1/(r_y(x)\varphi_y)$.

The second part of the selection process (line 6 10) will estimate the delay with coexisting traffic. In addition to the expected delay of transmitting two packets sequentially over the same route and the delay from AP to MN, the $\Delta(\Gamma_x, \Gamma_y(k))$ term represents the average delay of first transmitting a packet from an active route $\Gamma_x$ before transmitting on $\Gamma_y(k)$, while the $\Delta(\Gamma_y(k), \Gamma_y(k))$ term represents the average delay of first transmitting a packet from $\Gamma_y(k)$ before transmitting from an active route. Notice that the actual dispatching delay with pipeline dispatching is unknown during the stage of route and access point selection. The delay metric for selection is a heuristic estimation.

**V. PERFORMANCE EVALUATION**

A discrete time simulator is implemented to evaluate the proposed WiMax-based radio access network. The locations of WiMax base stations and subscriber stations are randomly generated over a 12 km by 5 km rectangular space. The default number of base stations is 15 and all of them act as relay node for the mesh radio access net-
work. One base station also acts as gateway that connects to the global Internet. The radio transmission range is set to 2.5 km. Omni-directional antennas are used for transmission, and nodes within the transmission radius are not allowed for simultaneous transmission. None of the WiMax base stations is connected via the mesh backbone, and each subscriber station is served by at least one base station. The radio link rates are randomly generated. The maximum link rate between and two mesh relay node and base station to subscriber station is $M_{\text{mesh}}$ and $M_{\text{user}}$ respectively. Due to the space limitation, we only show selected scenarios with $M_{\text{mesh}} = 20 \text{Mb/s}$ and $M_{\text{user}} = 4 \text{Mb/s}$. The constant-bit-rate traffic with fixed packet size is sent to each of the subscriber station. The number of received packets and the packet delivery latency are the two metrics for performance evaluation.

We compare 4 schemes in the following figures. The scheme that applies proposed scheduling and routing is denoted as $R + S$ in the figures. The scheme with proposed routing and base case scheduling is denoted as Route while the scheme with base scheduling and the proposed scheduling is denoted as Schedule. For comparison, the maximum rate route between gateway and the base $k$ station is selected in the base routing scheme. The serving base station $k$ for subscriber $x$ is selected with $\text{max}(R(GW, k), r_k(x))$, in which $R(GW, k)$ is the the route max-rate from gateway to $k$ and $r_k(x)$ is the link rate between base station $k$ and subscriber station $x$. Meanwhile, the base scheduling scheme applies FIFO discipline for packet dispatching.

As shown in the figures, the performance difference widened as the system loads increase. When the number of users increases to 100, the congestion becomes severe and the number of packet loss increases as shown in the figures. In terms of packet delivery delay, the proposed routing and scheduling scheme shows significant improvement over the base scheme. In the base scheme, the average packet delay increases to the range of 300 400 ms as the load increases. On contrary, the average packet delay is less than 50 ms in the proposed scheme. In terms of the total delivered packets, the proposed scheme also shows a 20% improvement in heavy-load situations.

VI. CONCLUSIONS

A IEEE 802.16-based RAN architecture is proposed to provide a flexible and low-cost deployment solution for the future metropolitan-area wireless network. Not only subscriber stations are connected via WiMax links, WiMax wireless mesh backhaul connects all of the base stations. Multiple mesh route construction, intelligent route selection and access point selection, and pipeline dispatch mechanism are investigated to attach the benefits of load balancing and user capacity improvement. The simulation results showed that the proposed design achieves significant lower packet delivery delay and also accommodates more users as the traffic load increases.

REFERENCES


