CHAPTER 3

The Proposed Call Admission Control Scheme

To introduce our proposed Call Admission Control Scheme, there are two main parts to discuss. One is path bandwidth calculation, and the other is bandwidth routing. Because the wireless medium resource is scarce, so we want to calculate the present bandwidth and know about if there is enough bandwidth to allow a new flow to achieve the QoS guarantee. And path bandwidth calculation is also the most important part we want to stress in this thesis. According to the information about the bandwidth, we can implement the call admission control scheme, and consider about drop policy and fairness problem. Our research object is to provide the substantial bandwidth to support QoS in multihop wireless network. We proposed a path bandwidth calculation method to estimate current bandwidth easily and quickly and which can be used whatever MAC protocol is.

3.1 Architecture

The infrastructure is as followed, in Figure 3-1.

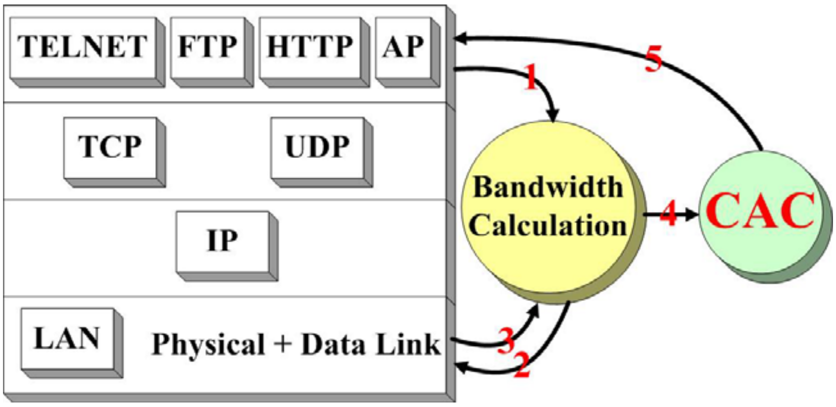

Figure 3-1: Architecture.

We plan to add a CAC (Call Admission Control) control procedure into the framework above TCP/UDP layer. Before admitting a new flow entering the network, we have to decide if the current network remaining bandwidth is enough to handle the new request. So when a new application (AP) requests to enter the network (Procedure 1 in Figure 3-1), we have to do “path bandwidth calculation” first. We inquire physical layer about the current channel usage statistics (Procedure 2), and employ the path bandwidth calculation method that we proposed to calculate the current bandwidth usage “percentage” (Procedure 3). After calculating, we pass the result to CAC (Procedure 4), and then CAC decide if we are going to accept or reject the new flow (Procedure 5).

We suppose to use the simulator, NCTUns, to simulate in the circumstance of IEEE 802.11 wireless unslotted, contention-based environment. At first, we will focus our simulation on path bandwidth calculation. The purpose is to know that the error percentage between our formula (idle percentage) and real bandwidth (according to log file), and see if the error percentage is acceptable.

The, we add the CAC module to the simulator and measure the throughput/goodput and delay as the architecture shown above. We will simulate the two situations, fixed and dynamic network topology, individually. Then we can evaluate the performance and efficiency using our proposed scheme.

3.2 Path Bandwidth Calculation

In path bandwidth calculation, we don’t want to assemble detail information such like every slot idle condition. We want the information of the idle percentage of the frame; each node senses its idle percentage respectively, then they can calculate link bandwidth by common idle percentage of the two nodes on the link, that is, the bandwidth condition we want to know about.

With our scheme, even though that there are lots of slots, the bit count we’re going to transmit about the channel condition is fixed and never increase.

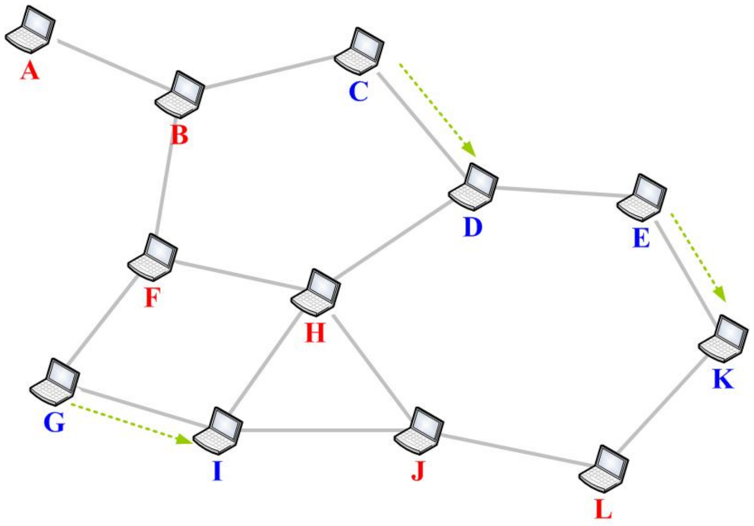

Figure 3-2: Illustration of Hops, Observers, and Interferers.

Figure 3-2 is an illustration of hops, observers, and interferers (given background traffic) If two nodes are adjacent, we call it one-hop distance, that is, single-hop; in this example, A to B and B to F are single-hop, etc. And if A wants to transmit to C or F, but the two nodes are out of the transmitting range of A, then A has accomplish through one intermediate node B; we call A to C two-hop. When the intermediate nodes from source to destination are more than one, we call it multi-hop, like A to L in this example. Here we define "observer(s)" as all the nodes along the path from source to destination (whatever how many hops they are); in this example, A is going to transmit data to L through B, F, H, and J, and we can call A, B, F, H, J and L observers. Lastly we define "interferer(s) or interfering pair(s)" as our given background traffics; {C, D}, {G, I}, and {E, K} here are the interferers which are always transmitting packets between them.

We will discuss the following situations, single-hop, two-hop and multi-hop respectively in the following sections.

3.2.1 Single Hop

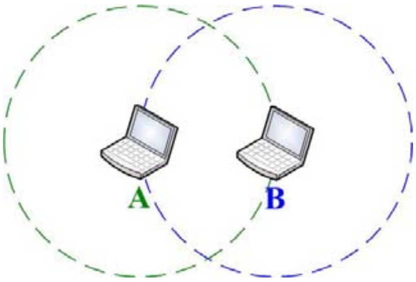

Figure 3-3: Single Hop.

As Figure 3-3 shown, if the mobile node is transmitting or receiving packets, that slot is indicated to be busy. If A is idle, that means that B is not transmitting data, but B is possible receiving data. On condition of one-hop distance, the common idle slots of A and B is the intersection of A's idle slots and B's idle slots, then A and B can transmit and receive packets through these slots.

First of all, we try to find a mathematical formula to apply to this situation. Although our path bandwidth calculation scheme could be used in unslotted, contention-based protocol, but during our deduction, we still take it as time-slotted for simplicity.

As the following Figure 3-4 shown, suppose that every slot is independent, x indicates the idle percentage/probability of Node 1, and y indicates the idle percentage/probability of Node 2. n is the virtual number of time slots.

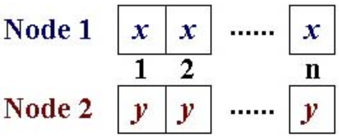

Figure 3-4: Induction of Single Hop.

n = 1: If there's only one slot, the common idle percentage is xy.

n = 2: If there are two slots, there are 3 common idle situations as followed.

The probability of 1 common idle slot: $C_1^2 xy(1-xy)$

The probability of 2 common idle slots: $(xy)^2$

The probability of 0 common idle slots: $(1-xy)^2$

⇨ generalize: If there are n slots

The probability of 0 common idle slots: $(1-xy)^n$

The probability of 1 common idle slot: $C_1^n xy(1-xy)^{n-1}$

The probability of 2 common idle slots: $C_2^n (xy)^2 (1-xy)^{n-2}$

……

The probability of i common idle slots: $C_i^n (xy)^i (1-xy)^{n-i}$

……

The probability of n common idle slots: $(xy)^n$

We can generalize from probability theorem that this is binomial distribution $b(i;n,p) = C_i^n p^i q^{n-i}, i = 0,1,2,...,n$. n represents the total number of time slots; and i indicates the number of common idle slots. Assume that common idle probability $p = xy$, busy probability (Node 1 and Node 2 are not idle at the same time) $q = 1 - p$, and the mean of the distribution is that $\mu = np$, the variance $\sigma^2 = npq$. So in mean number of n slots, common idle slots are nxy, and variance is $nxy(1 - xy)$. That is, common idle percentage is $\frac{nxy}{n}$, which equals to *xy*.

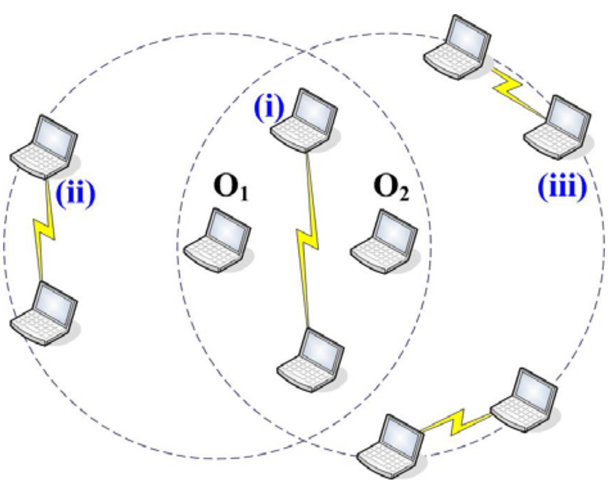

Figure 3-5: Illustration of available bandwidth analysis.

Here we discuss the different topologies around the observers which would affect the available bandwidth. We use Figure 3-5 to illustrate the available bandwidth analysis, and (i), (ii), and (iii) are interferences. First one is the worse case, there are hidden terminals to one of the observers, for instance, (i), (ii), and (iii) all exist, and the idle percentage is *xy*.

Second one is the best case, the two observers sense all the same transmitting pairs around them, for example, only (i) exists in this topology, and the idle percentage in this case is *min(x,y)*. Third is the in-between case, that is, there exist (i)(ii) or (i)(iii), and the idle percentage is between *xy* and *min(x,y)*.

We can also conclude *min(x,y)* as the upper bound of single hop idle percentage, and *xy* is the lower bound. The available bandwidth depends on whether routing protocols can provide the information to choose the appropriate calculation. If there is no information about the topology, then we just choose the lower bound, namely *xy*.

3.2.2 Two Hop

As Figure 3-6 Shown below, in the situation is two-hop, the bandwidth between arbitrarily two mobile nodes is between half of slot intersection of two nodes and slot intersection of them. Thus, we can calculate that the intersection bandwidth of A's idle slot and B's idle slot, and half of slot intersection bandwidth; then the same of B, C as well. The most conservative bandwidth between A and C is the intersection of half of slot intersection of A, B and B, C.

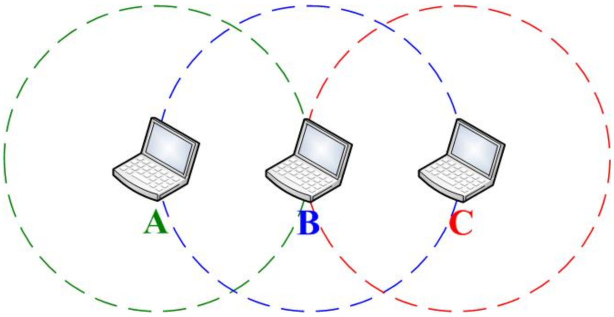

Figure 3-6: Two-Hop.

As Figure 3-7 depicted below, *x*, *y*, and *z* indicates channel idle probability of Node 1, Node 2 and Node 3 respectively. The difference between two-hop and single-hop is that it's not necessary to transmit data when the three nodes have the common idle slots simultaneously. A indicates average common idle probability of Node 1 and Node 2, B indicates that of Node 2 and Node 3, and C indicates the three nodes all common idle probability. Because Node 2 cannot receive and transfer packets in the same time (C shares the same uplink and downlink channel), we have to take half slots of the intersection set. Thereby on condition of two-hop, the average idle probability is $\frac{C}{2} + \min(A - C, B - C)$.

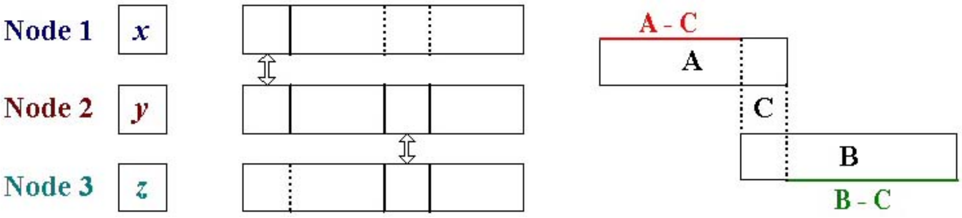

Figure 3-7: Induction of Two-Hop.

Mean $A = xy$, $B = yz$, $C = xyz$

The mean idle percentage of Node 1, Node 2 and Node 3 is $\frac{xyz}{2} + \min[xy(1-z), yz(1-x)]$.

Variance $A = xy(1-xy)$, $B = yz(1-yz)$, $C = xyz(1-xyz)$

The variance idle percentage of Node 1, Node 2, and Node 3 is $\frac{xyz(1-xyz)}{2} + \min[xy(1-z)(1-xy-xyz), yz(1-x)(1-yz-xyz)]$.

Similar to single-hop, the two-hop available bandwidth analysis is as follows. If the transmitting pairs around the observers are not all the same to them, then the idle percentage of this two-hop link is $\frac{xyz}{2}+\min[xy(1-z), yz(1-x)]$. If Node 1 and Node 2 sense the same traffic flows around them, or Node 2 and Node 3 sense the same traffic flows, then the idle percentage of the link is *min(x,y,z)*. And we also have the in-between case here. The first one $\frac{xyz}{2}+\min[xy(1-z), yz(1-x)]$ is the lower bound, and the last one *min(x,y,z)* is the upper bound in two-hop case. The available bandwidth depends on whether routing protocols can provide the information to choose the appropriate calculation. If there is no information about the topology, then we just choose the lower bound, that is, $\frac{xyz}{2}+\min[xy(1-z), yz(1-x)]$.

3.2.3 Multi Hop

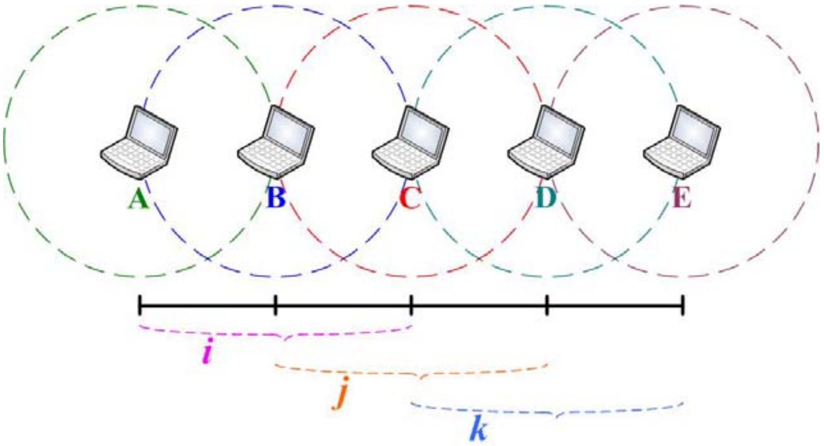

Figure 3-8: Induction of Multi Hop.

Generally, there are several hops to transmit datagram from source to destination. And our proposed solution to multi-hop is the expansion of two-hop solution. As Figure 3-8 shown above, *i*, *j*, *k* indicate the idle percentage of A to C, B to D, C to E (all two-hop distance) respectively. And the idle percentage of the whole path from A to E is *min(i,j,k)*. We acquire the multi-hop path idle percentage effortlessly.

3.3 Bandwidth Routing

There are many routing protocols for ad hoc mobile wireless networks, and here we choose the modified Bellmen-Ford routing protocol, DSDV, as our starting point since it is simple. We combine path bandwidth calculation and DSDV routing algorithm into a new one; that is, we extend the DSDV routing protocol to further include the information of bandwidth in the routing table. Because of “before-demand” path bandwidth calculation, a host can decide either to accept or to reject a new call immediately without any delay. The mobile node can do path bandwidth calculation and then build a routing table for any destined mobile node.

Our path bandwidth calculation is used over the loop free DSDV routing algorithm and the MAC layer is implemented using IEEE 802.11 mechanism. We combine bandwidth information and routing path into one message, which has four important fields, destination, source, distance and bandwidth. First, a node broadcast its bandwidth usage message will fill the destination field with its own id number and the distance field with zero as well as bandwidth equal to node idle percentage. When the neighbors receive this message and finish path bandwidth calculation, they will transfer the message by filling their id number into the source field and increase the distance field by one. Then the receivers will know that there is a path from itself to node X and passed by node Y, where node X and Y are recorded in destination field and source filed respectively. So they can fetch required routing information from the message and record in routing table. If the calculation result is useless, the node will discard the message; otherwise, it will transfer the message by replacing the source field with its id number, increasing the value in distance field by one, and updating the path bandwidth calculation results (bandwidth idle percentage). At last, each node builds a routing table for every destination node as shown in Table 3.1, which is a routing table for destination node A according to the topology in Figure 3-9. The last column, namely Bandwidth, which is the path bandwidth calculation result, is kept in each node locally but doesn't need to exchange to other nodes. For example, the third row in the table shows that node A can send packets through node B to node C, with distance = 2, and idle percentage = C's, etc.

Table 3.1: Routing Table at Node A.

Destination	**Next Hop**	**Distance**	**Idle Percentage**	**Bandwidth**
A	n/a	0	A's	
B	B	1	B's	
C	B	2	C's	
D	B	2	D's	

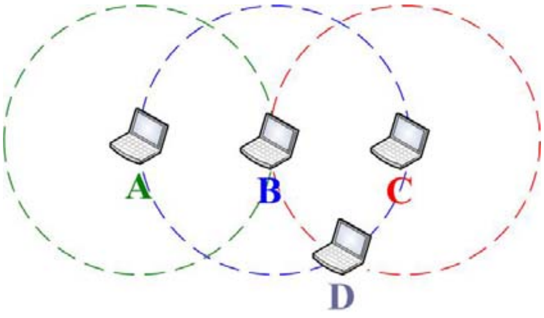

Figure 3-9: Topology for Table 3.1.

The algorithm would find some candidate paths to any destination with associated available bandwidth. Each node will choose the shortest path with enough bandwidth when it wants to transmit data to the desired destination. If no proper path is available, the request would be rejected.

3.3.1 Call Admission Control Policy

At present, our call admission control policy doesn't take many factors into account. When a new flow demands to enter the network and our path calculation result is enough for the request, then we admit the new flow, but we don't do the follow-up admission control for the time being. And we still don't implement a stand-alone and complete call admission control scheme; we just perform the call admission control policy at the source node now.

3.3.2 Contribution Discussion

Our proposed path bandwidth calculation scheme is to calculate the "idle percentage" of the available remaining bandwidth of the link. The advantage of this policy is that it is simple. And what's more, as we mentioned before, the decrement in bandwidth not only affects the link, but also influences the rage around it. If we just deduct the bandwidth from its present available one, while each node could only sense its own bandwidth condition, we couldn't calculate the path bandwidth precisely. Through our proposed path bandwidth calculation, we could easily calculate the path idle percentage of the link.

The main goal of this thesis is to how the knowledge of network bandwidth can be efficiently collected and how the applications can exploit this bandwidth information in order to enhance performance and to support QoS in a wireless multi-hop network. Compared to the related work, all the bandwidth routing papers we referenced were using TDMA mechanism which is time-slotted. However, they are restricted in TDMA systems and somehow complicated in path bandwidth calculation. We would like to propose another path bandwidth calculation solution that can be used not only in TDMA (time-slotted) but also in IEEE 802.11 (unslotted, contention-based) networks. Our method doesn't need any other supported hardware; for instance, CDMA, etc., all we need is CSMA mechanism. And there is little additional overhead (little information exchange and computation overhead) to do bandwidth routing with our path bandwidth calculation scheme. Applying our path bandwidth calculation results, it is easy to implement call admission control, and is also easy to combine path bandwidth calculation and ad hoc routing protocols into a new bandwidth routing algorithm according to our requirements. It is used in order to guarantee required QoS for real time traffic. Thus, the notion of "bandwidth" for a certain link here is equal to the "idle percentage" on the link.