International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.1, Issue1, pp-074-083 ISSN: 2249-6645

Compression of Initial Ranging Scheme in WIMAX Using Markova Models

[1]Dr.Balakrishna R , [2]Vijayanand S, [3]Rajesh K.S.
[1]Professor & HOD, Dept of ISE, Rajarajeswari College of Engineering, Bangalore.
[2,3]Asst.. Professor, Dept of ISE/CSE, Rajarajeswari College of Engineering, Bangalore

Abstract :

In this paper to analyze the Initial Ranging (IR) scheme in terms of the delay incurred and then proposes and implements an improvement to the same scheme. Initial Ranging is the process of obtaining the correct timing offsets and power adjustments such that the Subscriber Station (SS) is co-located with the Base Station (BS) and is transmitting at a suitable power level. During this process, which is a vital part of the network entry procedure in WiMAX, a series of request and response packets are exchanged between the Subscriber Stations and the Base Station. Due to the presence of multiple Subscriber Stations the request packets collide with each other thereby increasing the time required to complete the Initial Ranging process. Therefore, in our paper we analyze and obtain a formula for the delay incurred in Initial Ranging, using Markov models. In order to reduce the delay, we propose an enhancement to the existing Initial Ranging scheme. The concept of circularity is introduced, resulting in a reduced probability of collisions among the ranging request packets and compare the improved scheme with the original, and validate the delay formula obtained, using simulations.

Key words: Base station, Initial Ranging, WiMax,

Related work

Initial Ranging (IR) is the process of obtaining the correct timing offsets and power adjustments such that the Subscriber Station (SS) is co-located with the Base Station (BS) and is transmitting at a suitable power level. During this process, which is a vital part of the network entry procedure in WiMAX[1], a series of request and response packets are exchanged between the Subscriber stations and the Base Station. Due to the presence of multiple Subscriber stations the request packets collide with each other thereby increasing the time required to complete the Initial Ranging process. Therefore, in our project we analyze and obtain a formula for the delay incurred in Initial Ranging, using Markov models. Also, in order to reduce the delay involved, we propose an enhancement to the existing Initial Ranging scheme. The concept of circularity[2] is introduced to make the Subscriber stations less self-seeking, resulting in a reduced probability of collisions among the ranging request packets. We then compare the improved scheme with the original, and validate the delay formula obtained, using simulations.

Introduction

Pervasive computing describes an environment where a wide variety of devices carry out information processing tasks on behalf of users by utilizing connectivity to wide variety of networks. Pervasive computing does not just mean "computers everywhere"; it means "computers, networks, applications, and services everywhere." Mark Weiser[3] was a pioneer of this field.

Pervasive computing is roughly the opposite of virtual reality. Where virtual reality puts people inside a computer-generated world, pervasive computing forces the computer to live out here in the world with people.

Pervasive computing creates an augmented reality. It enriches objects in the real world and makes them "smart." This allows these devices to better assist people. With additional information about the environment and the context, these devices become better tools for the people using them.

Pervasive Computing is also known as *Ubiquitous* or *Nomadic* computing

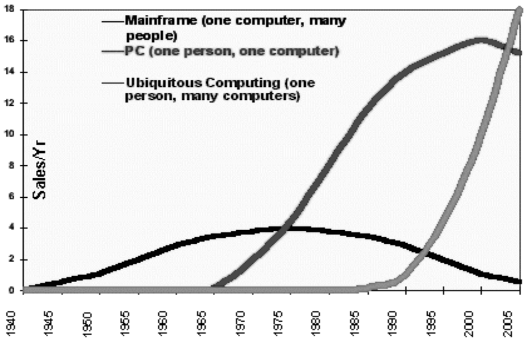

Figure 2.1: Major trends in Computing.

Ubiquitous[4] computing names the third wave in computing, just now beginning. First were mainframes, each shared by lots of people. Now we are in the personal computing era, person and machine staring uneasily at each other across the desktop. Next comes ubiquitous computing, when technology recedes into the background of our lives.

1.Evolution of Pervasive Computing

Mobile computing and pervasive computing represent major evolutionary steps in a line of research dating back to the mid-1970s. Figure 2.2 illustrates this evolution from a systems-centric viewpoint. New problems are encountered as one move from left to right in this figure. In addition, the solutions of many previously-encountered problems become more complex - as the modulation symbols suggest, this increase in complexity is multiplicative rather than additive[4]. It is much more difficult to design and implement a mobile computing system than a distributed system of comparable robustness and maturity; a pervasive computing system is even more challenging. As Figure 2.3 indicates, the conceptual framework and algorithmic base of distributed systems provides a solid foundation for mobile and pervasive computing.

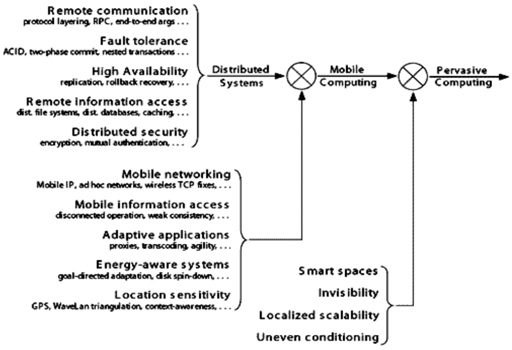

Figure 2.2: Evolution of Pervasive Computing
Characteristics of pervasive computing

Small: Ongoing miniaturization of components, moving to invisibility.

Embedded: Components are placed on or within other devices, objects or living beings.

Networked: Flexible capacity to exchange data and software components with other devices and platforms.

Context sensitive: Collect and exchange data on their environment and the host object via sensors.

Adaptive: Implement changes and modifications at the software and object level.

Collaborative: Ability to discover other objects and interact with them to establish cooperation on the software or information level

Network Volume: Sufficient in number and regularity of interaction to create network behaviors

This list of characteristics is interesting in how unremarkable it is. It indicates the extent to which the trajectory of the technology used in pervasive computing has already been determined, even if the implementation remains problematic. In this sense, pervasive computing is not about the introduction of a single technology, but about a potential qualitative change that may arise through an increasingly integrated technological environment

In brief Pervasive Computing Model can be summarized as follows:

- Access control.
- Anytime.
- Anywhere.
- Any Device.
- Any Network.
- Any Data.

Coupled with intelligent applications the above model can be realized.

Access Control

Information available in any the Pervasive System should be protected and should be accessible only to authorized personal. Hence access control is a very important aspect of any Pervasive System.

Architecture of an access control mechanism:

(1) Extract pieces of information in raw data streams early[5]

(2) Define policies controlling access at the information level

(3) Exploit information relationships for access control.

Anytime and Anywhere.

Pervasive Computing makes it possible users to access its services anytime and anywhere, round the clock.

Any Device

The power of Internet coupled with mobile devices such as PDAs, laptops envision the above goal.

The main features of pervasive devices are size, style, physical characteristics, content and services offered by them.

Based on size, there are three basic types of pervasive devices.

- Very Small Screen Devices (up to 4 inches)
- Small Screen Devices (up to 8 inches)
- Medium Screen Devices (up to 15 inches)

Any Network

Pervasive computing allows connectivity between different kinds of networks. Regardless of the network used at two ends, there can be a connection or information exchange between them.

Any Data

Data is classified as:

Categorical data- the objects being studied are grouped into some categories based on qualitative trait[6].
Measurement data- the objects being studied are grouped into some categories based on quantitative trait.

1. MARKOV MODELS

Markov processes provide very flexible, powerful, and efficient means for the description and analysis of dynamic (computer) system properties. Performance and dependability measures can be easily derived. Moreover, Markov processes constitute the fundamental theory underlying the concept of queuing systems. In fact, the notation of queuing systems has been viewed sometimes as a high-level specification technique for (a sub-class of) Markov processes[6].

Each queuing system can, in principle, be mapped onto an instance of a Markov process and then mathematically evaluated in terms of this process. But besides highlighting the computational relation between Markov processes and queuing systems, it is worthwhile pointing out also that fundamental properties of queuing systems are commonly proved in terms of the underlying Markov processes. This type of use of Markov processes is also possible even when queuing systems exhibit properties such as non exponential distributions that cannot be represented directly by discrete-state Markov models. Markovizing methods, such as embedding techniques or supplementary variables, can be used in such cases. Here Markov processes serve as a mere theoretical framework to prove the correctness of computational methods applied directly to the analysis of queuing systems. For the sake of efficiency, an explicit creation of the Markov process is preferably avoided.

Stochastic Process

A stochastic process is defined as a family of random variables {Xt : t E T} where each random variable Xt is indexed by parameter t belonging to T, which is usually called the time parameter if T is a subset of R+ = [0, infinity). The set of all possible values of Xt (for each t belonging to T) is known as the state space S of the stochastic process[7].

If a countable, discrete-parameter set T is encountered, the stochastic process is called a discrete-parameter process and T is commonly represented by (a subset of) No = {0,1,. . . }; otherwise we call it a continuous-parameter process. The state space of the stochastic process may also be continuous or discrete. Generally, we restrict ourselves here to the investigation of discrete state spaces and in that case refer to the stochastic processes as chains, but both continuous- and discrete-parameter processes are considered[8].

A large number of stochastic processes belong to the important class of Markov processes. The theory of Markov chains and Markov processes is well established and furnishes powerful tools to solve practical problems. This chapter will be mainly devoted to the theory of discrete-time Markov chains, while the next chapter concentrates on continuous time Markov chains.

Definition of Markov process

A stochastic process { X (t), t Є T} is a Markov process if the future state of the process only depends on the current state of the process and not on its past history. Formally, a stochastic process{ X (t), t Є T} is a continuous time Markov process if for all t0 < t1 < · · · < tn+1 of the index set T and for any set {x0, x1, x2, …,xn+1} of the state space it holds that

$P_r[X_{k+1}=x_{k+1}|\ X_0=x_0,\ldots,X_k=x_k|=P_r[X_{k+1}=x_{k+1}[X_k=x_k]$

A Markov process is called a Markov chain if its state space is discrete. The conditional probabilities Pr [Xk+1 = j | Xk = i] are called the transition probabilities of the Markov chain. In general, these transition probabilities can depend on the (discrete) time k. A Markov chain is entirely defined by the transition probabilities and the initial distribution of the Markov chain Pr [X0 = x0]. Thus we obtain the following relations,

$$P_r[X_0=x_0,\ldots,X_k=x_k\}=P_r[X_k=x_k[X_0=x_0,\ldots X_{k-1}=x_{k-1}] * P_r[X_0=x_0,\ldots X_{k-1}=x_{k-1}]$$

By the definition of Markov model, we get the following relation,

$$P_r[X_0=x_0,\ldots,X_k=x_k\}=P_r[X_k=x_{k\,|}\ X_{k-1}=x_{k-1]} * P_r[X_0=x_0,\ldots X_{k-1}=x_{k-1}]$$

Discrete Time Markov Chain

If the transition probabilities are independent of time k,

Pij = Pr[Xk+1 = j | Xk = i]

the Markov chain is called stationary. In the sequel, we will confine ourselves to stationary Markov chains. Since the discrete-time Markov chain is conceptually simpler than the continuous counterpart, we discuss with the discrete case.

4 .ANALYSIS AND DESIGNOF INITIAL RANGING

4.1 ANALYSIS OF IR SCHEME

After analyzing the Initial Ranging procedure, we enumerate the following states as well as transitions needed for modeling the procedure.

4.1.1 States involved

State 1: Waiting for UL-MAP. This is also the start state.

State 2: SS is performing Backoff procedure.

State 3: Waiting for an RNG-RSP message from BS.
State 4: Continue
State 5: Success State – Wait for CDMA Allocation IE.
State 6: Abort – Start network entry procedure at a different DL channel
State 7: Waits for RNG-RSP again.
State 8: Proceed to next phase of network entry
State 9: Commence Periodic Ranging

4.1.2 Transitions involved

In State 1, the SS waits for a UL-MAP. After receiving this message it makes a transition to State 2. Transmission of CDMA code occurs at end of State 2. Also a timer is set for waiting for RNG-RSP message. This transition leaves the system in State 3.

When in State 3, if the timer for RNG-RSP expires then SS increments the power level and goes back to State 1.

When in State 3, if RNG – RSP is obtained with Ranging code as well as the Ranging slot, then it makes a transition to State 4. Here the necessary adjustments specified in RNG-RSP are made and system moves to State 1.

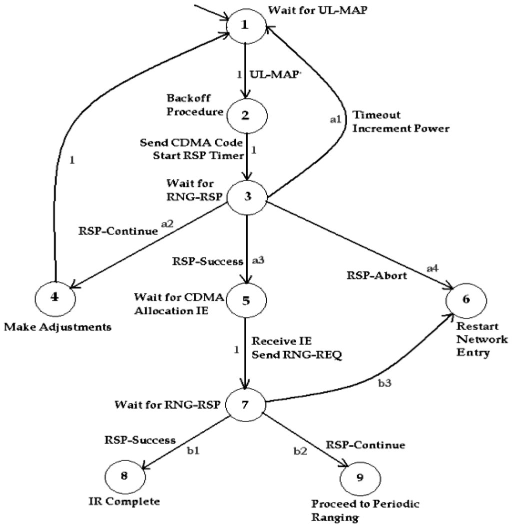

Figure 7.1 Markov Model of IR Scheme

When in State 3, if RNG-RSP is obtained with success status, then the system transits to State 5. Here it waits for CDMA Allocation IE. After reception it sends RNG-REQ message on the allocated bandwidth and moves to State 7[9].

When in State 7, on reception of RNG-RSP with success status it moves to State 8. On reception of RNG-RSP with continue status it moves to State 9. Else on reception of RNG-RSP with abort status, it goes to State 6 and SS starts the network entry procedure again.

When in State 3, if RNG-RSP is obtained with abort status then the system goes to State 6 and SS starts the network entry procedure again. The following diagram shows the Markov model that represents the Initial Ranging procedure of IEEE 802.16 network standard.

The states 6, 8 and 9 lead out of the IR model and are the absorbing states.

Next, we use the transition matrix obtained above to obtain the overall delay formula. For this, we first need to tabulate the delays involved in the individual states[10].

4.1.3 Probabilities of Transition

The Transition Probability Matrix corresponding to the Markov model is as follows:

	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0
3	a1	0	0	a2	a3	a4	0	0	0
4	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	1	0	0
6	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	b3	0	b1	b2
8	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	1

Table 7.1 show transition probability matrix to the Markov model

The states 6, 8 and 9 are absorbing states. P (i, i) = 1.

Delay Involved	Probabilities
UL-MAP Reception (1 to 2)	1
Backoff Delay + Sending CDMA (2 to 3)	1
RNG-RSP Timeout (3 to 1)	a1
RNG-RSP Reception + Processing (3 to 4, 5 or 6)	a2,a3,a4
IE Allocation Delay + Sending RNG-REQ (5 to 7)	1
RNG-RSP Reception + Processing (7 to 8, 9 or 6)	b1,b2,b3

Table 7.2 details of delay involved and probability

4.1.4 Details of Delays involved (Other than Backoff delay)

1. UL-MAP Reception = 5ms (Maximum of one complete frame length)
2. CDMA Sending Time = Transmission Time = 5ms/2 = 2.5ms
   [Frame Length/2 (Length of UL subframe) with frame length = 5ms]
3. RNG-RSP Timeout (T3) = 200 milliseconds.

4. RNG-RSP Reception + Processing (average value)
   = T3/2 + Max. RNG-RSP Processing Time/2 = 100 ms + 10ms/2 = 105 ms
5. CDMA Allocation IE delay = 5s (same as 1)
6. Sending RNG-REQ (Same as 2) = 2.5ms
7. RNG-RSP Reception + Processing (average value) = 105ms

   We assume that the delay involved for making changes at SS is negligible compared to the other delays involved.

4.1.5 Back-Off Delay Derivation

Consider the first time an SS enters Backoff procedure. Let the Initial Contention window be w_0. The random number will be picked in the range [0, w_0-1]. Let this random number be called k. The SS has to defer a total of k contention slots. Let the number of CS's in a frame be n_{cs}. The number of frames that have to be deferred is k/ n_{cs}. The delay involved here will be (k/ n_{cs})*frame length. After k/ncs frames have passed the SS defers a further k modulus ncs CS's. The delay involved here is equal to (k% n_{cs})*T_{cs}, where T_{cs} is the length of one CS[11].

Total delay so far = (k/ n_{cs})*frame length + (k% n_{cs})* T_{cs}

Here the value of k can vary from 0 to w_0-1. Thus, we take an average of the delay over the random number.

AD_0 = (1/w_0) *Sum of [(k/n_{cs})*frame length + (k%n_{cs})*T_{cs}]
as k varies from 0 to w_0-1.

Next we make an assumption that the probability of a successful transmission in a CS is 'p'. Thus, probability of failure will be '1-p'. In case of a failure the contention window is doubled in size. Let the new window be [0, w_1-1]. Similar to previous derivation the delay involved will be

AD_1 = (1/w_1) *Sum of [(k/n_{cs})*frame length + (k% n_{cs})*T_{cs}]
as k varies from 0 to w_1-1. Here $w_1 = 2 * w_0$.

Again there could be success or failure. So, it will enter the third Backoff window phase [0, w_2-1]. Continuing in this fashion, we get the following delays for the next three phases.

AD_2 = (1/w_2) *Sum of [(k/ n_{cs})*frame length + (k% n_{cs})* T_{cs}]
as k varies from 0 to w_2-1.

AD_3 = (1/w_3) *Sum of [(k/ n_{cs})*frame length + (k% n_{cs})* T_{cs}]
as k varies from 0 to w_3-1.

AD_4 = (1/w_4) *Sum of [(k/ n_{cs})*frame length + (k% n_{cs})* T_{cs}]
as k varies from 0 to w_4-1.

Here $w_2 = 2 * w_1$, $w_3 = 2 * w_2$, $w_4 = 2 * w_3$.

We make another assumption at this point. The SS is assumed to complete successful transmission of its CDMA code, in a maximum of 5 Backoff phases. Thus, the worst case of transmission will be four failures followed be a success. The final formula for the delay will be as follows.

Backoff Delay (BD)

$$\begin{aligned}&= p*\{AD_0 + t/2\}\\&+ ((1-p))*p*\{[AD_0 + t] + [AD_1 + t/2]\}\\&+ ((1-p)^2)*p*\{[AD0 + AD_1 + 2t] + [AD_2 + t/2]\}\\&+ ((1-p)^3)*p*\{[AD0 + AD_1 + AD_2 + 3t] + [AD_3 + t/2]\}\\&+ ((1-p)^4)*p*\{[AD0 + AD_1 + AD_2 + AD_3 + 4t] + [AD_4 + t/2]\}\end{aligned}$$

→ Equation 7.1: Backoff Delay incurred in IR scheme

Here t is the time-out after which failure is assumed. So, we take half that value for success i.e. t/2.

4.1.6 Overall Delay Formula

By traversing the transition diagram and multiplying the probabilities with the corresponding delays, the total delay can be calculated. The resulting formula is as follows.

The first part of the delay is in the loops 1-2-3-1 and 1-2-3-4-1. We call this D-loop. Then either success or abort occurs which is added to this part to get the final formula.

$$\begin{aligned}D_{loop} &= 1*\text{UL-MAP} + 1*(\text{BD} + \text{CDMA sending})\\&+ a_1 * (\text{Timeout T3} + \text{D-loop})\\&+ a_2 * (\text{RSP} + \text{D-loop})\end{aligned}$$

Simplifying we get,

$$D_{loop} = \frac{\text{UL} + \text{BD} + \text{CDMA sending} + a_1*\text{T3} + a_2*\text{RSP}}{1-(a_1+a_2)}$$

→ Equation 7.2: The delay in the loops of IR Markov model

Now, the total delay involved can be represented using the formula given below.

$$\begin{aligned}D_{total} &= D_{loop}\\&+ a_3 * (\text{RSP} + \text{CDMA\_IE} + \text{RNG-REQ} + (b_1+ b_2 + b_3)*\text{RSP})\\&+ a_4 * \text{RSP} \text{ (here } b_1 + b_2 + b_3 = 1)\end{aligned}$$

→ Equation 7.3: Total delay in terms of loop delay

Substituting equation 2 in equation 3, we arrive at the final formula.

$$\boxed{D_{total} = \frac{\text{UL} + \text{BD} + \text{CDMA sending} + a_1*\text{T3} + a_2*\text{RSP}}{1-(a_1+ a_2)} + a_3* (\text{RSP} + \text{CDMA\_IE} + \text{RNG-REQ} + \text{RSP})}$$

$$+ a_4 * RSP$$

→ Equation 7.4: Total delay incurred in the IR scheme

In equation 7.4 the Backoff Delay (BD) is given by equation 7.1

4.2 ENHANCEMENT OF IR SCHEME

During the IR procedure, the Subscriber Stations attempt to synchronize with the Base Stations. Since connections are not yet established, the process is contention based. As a result, there is always a finite probability that the request packets sent from different SS collide with each other. These packet collisions decrease the net throughput of the network, increase the Initial Ranging delay of SSs and ultimately lead to inefficient bandwidth utilization[12].

In our enhancement of the IR scheme, we attempt to reduce the probability of collisions between packets. We make use of the concept of circularity. Circularity is defined as a number which enables the identification of specific groups of events or packets. Each event or packet under consideration is numbered in a sequential manner. An event or packet with a number which is a multiple of the circularity value is said to be circularity-satisfied. It can be represented mathematically as follows

(Packet/Event number) = 0 modulo circularity

A finite delay is introduced before the occurrence of circularity satisfied events or the sending of circularity satisfied packets. This additional delay reduces the probability of packet collisions. The selfless behavior of certain SSs may increase the individual IR delays but on the whole the delay incurred in the entire network will be reduced.

In the IR procedure, whenever a ranging response packet is timed out, the backoff procedure is called and the window index is incremented. We apply the concept of circularity on this event. In case a time out event is circularity satisfied, the window index is increment an additional time. This, yields higher values for the backoff counter and hence reduces the packet collisions[5].

In addition, circularity is applied to ranging request packets in order to selectively increase the delay in their transmission. A finite delay is introduced before the circularity-satisfied request packets are sent. This further reduces request packet collisions.

5 . IMPLEMENTATION

The implementation consists of two parts:

- Simulation of existing scheme using ns-2
- Changing the backend and re-simulating

5.1 Simulation Setup

Tcl scripting is used to design and simulate the WiMAX[7] networks with varying architectures. Tcl gives us a lot of options that allow us to have a great degree of control over the simulation of networks. Some of the important features of the Tcl script we have written are shown below.

5.1.1 Parameters Used

The following parameters are used for the simulation of the existing Initial Ranging scheme.

- **General Parameters:**
    - Channel Type – WirelessChannel
    - Radio Propagation Model– TwoRayGround
    - Network Interface Type - Phy/WirelessPhy/OFDM
    - MAC Type – 802_16
    - Interface Queue Type – DropTail Priority Queue
    - Link Layer Type – LL
    - Antenna Model – Omni Antenna
    - Maximum Packets in Interface Queue – 50
    - Routing Protocol – DSDV (Routing is done through the Base Station)

- **Network Architecture Parameters:**
    - Number of Base Stations – 1
    - Number of Sink Nodes – 1
    - Number of Subscriber Stations – Varied from 6 to 54
    - Base Station Coverage – 20 meters
    - Traffic Start Time – 20
    - Traffic Stop Time – 40
    - Simulation Stop Time – 50

5.1.2 Simulation of Existing Initial Ranging Scheme

The parameters mentioned are used in the Tool Command Language (TCL) script that we have written. This script also uses the WiMAX Control Agent in order to produce a detailed account of the activities going on during the simulations. In the resulting output file, we search for the timing details of specific events in order to extract the Initial Ranging delay.

We search for two events:

1) Found Ranging Opportunity – This marks the starting point of the IR procedure.

2) Ranging Response obtained with Success status – This marks the ending of the IR procedure.

The corresponding start and stop times of the IR procedure for all the Subscriber Stations (SS) in the scenario are stored in files. Using a C program, we find the average IR delay per node, after calculating the total time taken by all the nodes to complete their respective IR processes.

```
void RangingRequest::expire () {
            count1++;
            mac_->debug ("Ranging request expires\n");
            if(nb_retry_==(int)mac_-
>macmib_.contention_rng_retry) {
//max retries reached, inform the scheduler
mac_->expire (type_);
} else {
if (window_ < s_->getBackoff_stop())
window_++;
nb_retry_++;
if (window_ < s_->getBackoff_stop() && count1 % 5
== 0)
window_++;
int result = Random::random() % ((int)(pow (2,
window_)+1));
mac_->debug ("Start Ranging contention in
%f(backoff=%d, size=%d, ps=%f)\n", result*s_-
>getSize()*mac_->getPhy()->getPS(),result,s_-
>getSize(),mac_->getPhy()->getPS());
    backoff_timer_->start (result*s_-    >getSize()*mac_-
>getPhy()->getPS());
                        backoff_timer_->pause();
             }
}
```

Such simulations can be carried out for different numbers of SS each time by the use of shell scripts. Then the average IR delay is recorded along with number of SS involved in each such simulation[8].

5.2 Enhancement of Initial Ranging

In order to enhance the Initial Ranging scheme, we make some modifications to the backend of ns-2, which is implemented in C++ language. The files that are of interest to us are the following:

contentionrequest.cc
contentionslot.cc

During the IR procedure, there will be many SS contending to send their requests to join the network. The packets sent by different SS may collide at some instants and they will have to be resent. We try to reduce the collisions between packets of different SS by making the SSs less selfish. We have made two such changes, one in each of the files mentioned above. Let us see each on in detail[9].

5.2.1 Modification of 'contentionrequest.cc'

In this file, apart from many functions, the expire () function of the Ranging Request class is implemented. It contains the actions to be taken in case the ranging response from the BS is not received within a certain time limit. In the existing scheme, upon timing out the Backoff window index is incremented to the next valid number.

For the purpose of enhancement, we introduce a counter that is incremented every time the expire() function is called. Now we introduce a concept called circularity. Circularity is defined as a number which enables the identification of specific groups of events or packets. It is implemented in terms of the modulo (%) operator. Consider the counter to be 'count1'. If 'count$_1$' is a multiple of the circularity value, then the event is said to circularity satisfied. It can be represented mathematically as follows.

$$\mathbf{count_1 = 0 \text{ modulo circularity}}$$

In case an 'expire' event is circularity satisfied, in order to reduce the probability of packet collisions, we increase the backoff window index one additional time. When we increase the backoff window an additional time, the backoff counter will assume larger values when chosen randomly. This reduces the chances of packet collisions from different SSs. After considering different circularity values, we observed that a value of 5 gives the maximum improvement in the delay. This is the first enhancement in our scheme. The code snippet below represents the improved 'expire()' function. The highlighted parts are the ones corresponding to the enhancement.

5.2.2 Modification of 'contentionslot.cc'

In this file, the addrequest() function of the RngContentionSlot class is implemented. This is a simple function in which, a new request packet is made ready in order to be sent during the ranging opportunity. The code snippet below shows the existing implementation.

As in the previous case, there is a certain finite probability of collision of these packets. In order to reduce the collision probability further, we keep track of the number of request packets being added. A global counter 'count2' is used that is incremented every time the addrequest() function is called at any SS. Now the principle of circularity is applied to these packets to identify the circularity-satisfied packets. The equation used is as follows.

$$\mathbf{count_2 \ \% \ circularity = 0}$$

Those packets which have their counter value as multiples of the circularity value are delayed by a finite amount of time before they are sent out. This is achieved by using a call to the pauseTimers() function followed by a call to the resumeTimers() function. After considering different circularity values, we observed that a value of 3 gives the maximum improvement in the delay. The modified code is shown below with the added part highlighted.

```
void RngContentionSlot::addRequest (Packet *p)
{
            assert (request_ == NULL);
            count2++;
            request_ = new RangingRequest (this, p);
            if ( count2 % 3 == 0 ) {
                        pauseTimers();
                        resumeTimers();
```

```
            }
}
```

5.2.3 Simulation of Enhanced Initial Ranging Scheme

Since the backend of the ns-2 code is changed, we need to recompile the entire backend. After this is achieved, simulations are carried out in identical fashion to the simulations of original IR scheme.

6. RESULTS AND COMPARISON

CASE 1: For one base station and many subscribers stations (64) in which all the subscribers stations are placed at same positions.

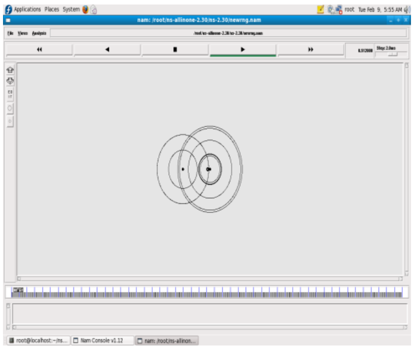

Fig 6.1 shows all mobile nodes access from same point

The delay values obtained for the above test case from the simulation are tabulated below.

No of subscribers stations	Delay in milliseconds
8	0.268365
16	0.454860
32	0.701238
64	0.845573

Table 6.1 Delay values for case1

The graphical representation for the above tabulated delay values is shown below

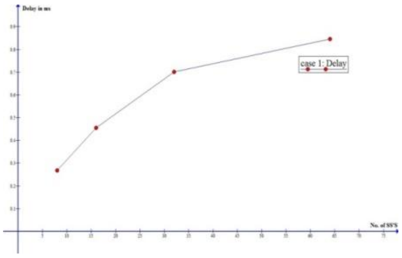

Fig 6.2 Graphical representation for delay values of case1

The throughput values obtained for the above test case from the simulation are tabulated below.

No of subscribers stations	Throughput in Mbps
8	0.48693
16	0.31706
32	0.23467
64	0.19960

Table 6.2 Throughput values for case1

The graphical representation for the above tabulated throughput values is shown below

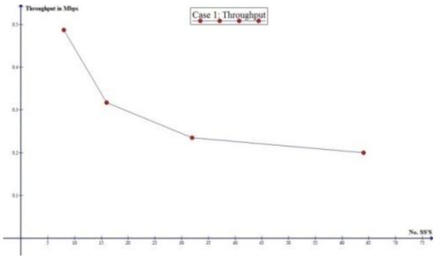

Fig 6.3 Graphical representation for throughput values of case1

CASE 2: For one base station and many subscribers stations (64) in which all the subscribers stations are placed at circular positions around the base station.

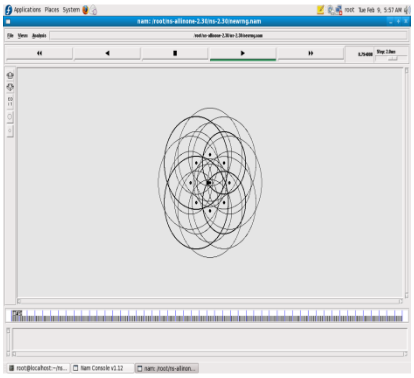

Fig 6.4 Shows mobile nodes access from circular position

The delay values obtained for the above test case from the simulation are tabulated below.

No of subscribers stations	Delay in milliseconds
8	0.2443542
16	0.4179602
32	0.5408570
64	0.6917536

Table 6.3 Delay values for case2

The graphical representation for the above tabulated delay values is shown below

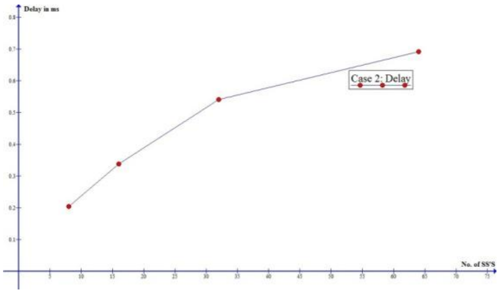

Fig 6.5 Graphical representation for delay values of case2

The throughput values obtained for the above test case from the simulation are tabulated below.

No of subscribers stations	Throughput in Mbps
8	0.53665
16	0.38782
32	0.28228
64	0.23638

Table 6.4 Throughput values for case2

The graphical representation for the above tabulated throughput values is shown below

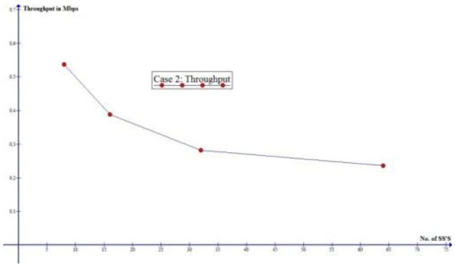

Fig 6.6 Graphical representation for throughput values of case2

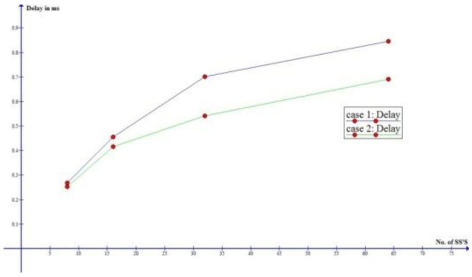

Fig 6.7 Comparison between case1 and case2 delay values

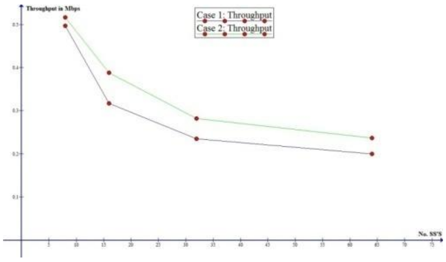

Fig 6.8 Comparison between case1 and case2 throughput values

CASE 3: For two base station and many subscriber stations (64) in which both the base stations have equal number of subscriber stations.

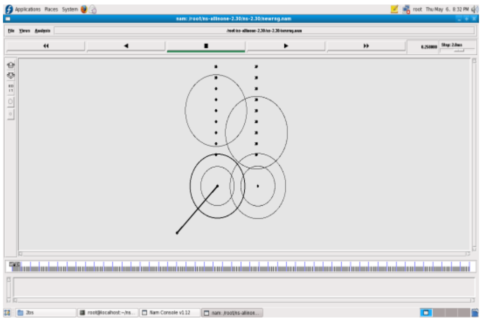

Fig 6.9 Shows 2BS having equal number of nodes

The delay values obtained for the above test case from the simulation are tabulated below.

No of subscribers stations	Delay in milliseconds
8	0.210553
16	0.404562
32	0.581445
64	0.765878

Table 6.5 Delay values for case3

The graphical representation for the above tabulated delay values is shown below

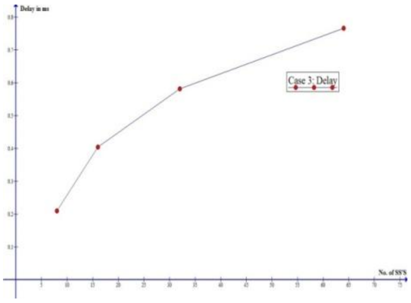

Fig 6.10 Graphical representation for delay values of case3

The throughput values obtained for the above test case from the simulation are tabulated below.

No of subscribers stations	Throughput in Mbps
8	0.569927
16	0.296617
32	0.206382
64	0.156682

Table 6.6 Throughput values for case3

The graphical representation for the above tabulated throughput values is shown below

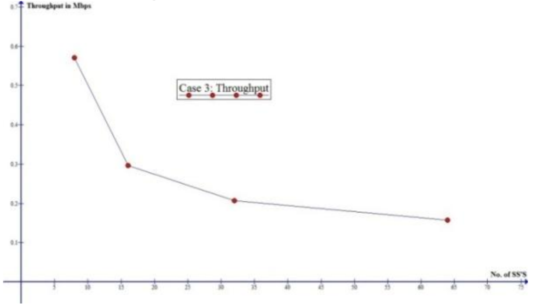

Fig 6.11 Graphical representation for throughput values of case3

These are the different cases on which we had successfully carried out simulation using NS-2. We had analyzed and evaluated delay and throughput for these above cases for IR mechanism. We had used various WiMAX patch files for its compatibility with NS-2. These patch files are backend files written in C++.

For the same purpose of analyzing and evaluating the performance of IR mechanism we had developed an generic code (shown in Appendix-B). This generic code written in visual C++ can be used to evaluate delay and throughput for further various cases.

CONCLUSION

In this paper, we have successfully analyzed and obtained a mathematical formula to calculate the delay involved in the Initial Ranging scheme. We have also enhanced this scheme using circularity, achieving about 25.10% reduction in the IR delay.

From the results and comparison we can also conclude that

1. With increasing number of SS, the circularity value controlling the delay must be increased.
2. With increasing number of SS, the circularity value controlling the window size must be decreased.

References

[1] *IEEE Standard for Local and metropolitan area networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1*, IEEE Std 802.16e-2005 and IEEE Std 802.16-2004/ Cor 1-2005 (Amendment and Corrigendum to IEEE Std 802.16-2004) Std., 2006.

[2] M. Morelli, “Timing and frequency synchronization for the uplink of an OFDMA system,” *IEEE Trans. Commun.*, vol. 52, no. 2, pp. 296–306, Feb. 2004.

[3] X. Fu and H. Minn, “Initial uplink synchronization and power control (ranging process) for OFDMA systems,” in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, vol. 6, Nov. 29- Dec. 3, 2004, pp. 3999–4003.

[4] Jenhui Chen and Wei-Kuang Tan, “Predictive Dynamic Channel Allocation Scheme for Improving Power Saving and Mobility in BWA Networks,” *ACM/Springer Mobile Networks and Applications (MONET)*, 2006.

[5] C. Eklund *et al.*, “IEEE Standard 802.16: A Technical Overview of the Wireless MAN Air Interface for Broadband Wireless Access,” *IEEE Commun. Mag.*, 40(6):98–107, June 2002.

[6] B. Fong, N. Ansari, A.C.M. Fong, G.Y. Hong, and P.B. Rapajic, “On the Scalability of Fixed Broadband Wireless Access Network Deployment,” *IEEE Commun. Mag.*, 42(9):12–21, September 2004.

[7] I. Koffman and V. Roman, “Broadband Wireless Access Solutions Based on OFDM Access in IEEE 802.16,” *IEEE Commun. Mag.*, 40(4):96–103, April 2002.

[8] T. Kwon, et al., “Design and Implementation of a Simulator Based on a Cross-Layer Protocol between MAC and PHY Layers in a WiBro Compatible IEEE 802.16e OFDMA System,” *IEEE Commun. Mag* 43(12):136–146, December 2005.

[9] Kyung-ah Kim, Chong-Kwon Kim, Tongsok Kim, A Seamless Handover Mechanism for IEEE802.16e Broadband Wireless Access, International Conference on Computational Science, vol. 2, pp. 527-534, 2005.

[10] Doo Hwan Lee, etc, “Fast Handover Algorithm for IEEE802.16e Broadband Wireless Access System”, ISWPC, 2006

[11] Sik Choi, Gyung-Ho Hwang, Taesoo Kwon, ”Fast Handover Scheme for Real-Time Downlink Services in IEEE 802.16e BWA System”, Vol. 3 (2005), pp. 2028-2032, IEEE VTC 2005 spring, Sweden, May 2005

[12] Minsik Shim, Hwasung Kim, Sangho Lee, "A Fast Handover Mechanism For IPv6 Based WiBro System", Feb.20-22, 2006 ICA0T2006.