MICRMA: Modified Incremental Collision Resolution Multiple Access

A.Prof Imane Aly Saroit Ismail
Information technology department
Faculty of Computers and Information
Cairo University

Abstract

The Incremental Collision Resolution Multiple Access (ICRMA) protocol is one of the famous Medium access control (MAC) protocols based on collision resolution. It dynamically divides the medium into cycles of variable length; each cycle consists of a contention period and a queue-transmission period. During the contention period, stations with one or more frames to send compete for the right to be added to the data-transmission queue using a deterministic tree-splitting algorithm. The queue-transmission period is a variable-length train of frames, which are transmitted; each on a slot; by stations that have been added to the distributed transmission queue by successfully completing a collision-resolution round in a previous contention period. It has been proved that in ICRMA high number slots are wasted (collided or idle), it also has high average frame's delay especially with small number of sending stations; so it decreases the network's performance. In this paper, the Modified Incremental Collision Resolution multiple access (ICRMA) which is a development of the ICRMA protocol; is proposed to improve the network's performance; it reduces the number of wasted slots and the average frame's delay.

ملخص:

إن بروتوكول الوصول حل التصادم المتزايد هو أحد بروتوكولات التحكم المشهورة فى الوصول للوسط المعتمدة على حل التصادم. إنه يقسم الوسط ديناميكيا إلى دورات متغيرة الطول، كل دورة تتكون من فترة تنافس و فترة إرسال للصف. خلال فترة التنافس، المحطات التى لديها قالب أو أكثر للإرسال تتنافس على حق الإضافة لصف إرسال البيانات بإستخدام خوارزم تقسيم الشجرة المحدد. إن فترة إرسال للصف هى عبارة عن قطار متغير الطول من القوالب المرسلة –كل فى شق–من المحطات المضافة لصف الإرسال الموزع عن طريق إكمال دورة حل التصادم بنجاح فى فترة تنافس سابقة. لقد تم إثبات أن عدد الشقوق المفقودة (متداخلة أو خالية) عالى فى هذا البروتوكول، كما أن متوسط تأخير القالب عالى خاصة إذا كان عدد المحطات المرسلة قليل، و بالتالى فأنه يقلل من أداء الشبكة. إن بروتوكول الوصول

حل التصادم المتزايد المعدل، و الذى هو تطوير لبروتوكول الوصول حل التصادم المتزايد، قد تم إقتراحه فى هذا البحث، لتحسين أداء الشبكة حيث يقلل عدد الشقوق المفقودة و متوسط تأخير القالب.

I. Introduction

When a set of stations share a single-channel communication medium, they communicate with each other by sending messages over this shared channel. If only one station transmit a message, transmission is successful. But if more than one station transmits simultaneously, collision occurs and all their messages are lost, so a medium access protocol is needed to regulate the access to the channel. Medium access protocols [1-7] are classified into two categories; collision-free and collision-based protocols.

Collision free protocols work by preventing collision at all, so they do not require collision detection circuit, which can be difficult or costly to implement in some systems. Some collision free protocols are time division multiple (TDMA), bit map, binary countdown and token rings [1-3]. In TDMA, the channel is divided into time slots and every station is assigned a unique slot. In the bit map protocol, a control frame contains a bit for each host, each host transmits a 1 bit during its allocated time slot if it has a frame to send. After each time slot has passed by, every station knows all of the stations that want to send, so they will each send in order without contention. In the binary countdown protocol, each host transmits its address bits when it wants to send, the bits in each position from different stations are boolean OR together. As soon as a station sees that a high order bit position that is 0 is overwritten by a 1, it gives up. This continues until only one station (with the highest number) is left, this one is allowed to send. In token rings; a token is passed around the ring and only the host in possession of the token may transmit. Generally, a host captures the token and destroys it, sends its frame then generates a new token.

In collision based protocols collision is allowed, so collision detection and resolution are needed. ALOHA and Carrier sense multiple access (CSMA) families are examples of collision-based protocols [1-3]. In ALOHA; a station transmits whenever it has data, when there is a collision, the station waits for a random period and then retransmits. An improvement is known as Slotted ALOHA; where time is divided into uniform slots whose size equals to the frame transmission time, transmission is permitted to begin only at a slot boundary, so colliding frames overlap totally, this increase the throughput. As in ALOHA when there is a collision, the station waits for a random period and then retransmits, but also at the slot's boundary. Carrier sense multiple access (CSMA) protocols use the fact that the propagation delay between stations is usually very small compared to the frame transmission's time. A transmitting station senses the link before sending, and do not send until it sees nothing on the medium, so collisions would be rare because they would occur only when two or more stations begin to transmit almost simultaneously.

Note that collision-free protocols, work well under high loads but underutilize the channel bandwidth under low loads resulting in larger delays, while collision-based protocols utilize the channel efficiently under low loads while the throughput reduces dramatically at high loads due to the increase in the number of collisions [7].

An efficient way to achieve a good utilization both at low and high loads is to dynamically allocate the channel bandwidth to the contending stations by resolving collisions. Tree splitting was one of the first techniques proposed for collision resolution [5-8]; where a collision occurs, the colliding stations are split into groups. Stations present in group 0 are allowed to transmit, followed by stations in group 1, 2, and so on. If a collision occurs in a group, then second-level groups are created. The above procedure is used recursively until all the collisions are resolved. Two main approaches are used to allocate stations in the groups; probabilistic and deterministic [7].

- In the probabilistic approaches, stations in each group are chosen randomly. These approaches have two main problems; very high delay and very low throughput (especially in case of high load).
- In the deterministic approaches, the stations are assigned unique identifiers which are used in resolving collision. The incremental collision resolution multiple access (ICRMA) [5]; based on tree-splitting concept; is a famous deterministic approach that is employed to overcome the above two problems.

The ICRMA protocol operates with a collision-resolution stack for control frames and a distributed queue for data frames. ICRMA builds a distributed transmission queue dynamically using a deterministic tree-splitting algorithm; access time to the channel is divided into cycles of transmissions for all members of the transmission queue, which is called a queue-transmission period, followed by short contention periods during which stations attempt to join the queue. The queue-transmission period is a variable-length train of frames from stations that have been added to the transmission queue by successfully completing a collision-resolution round in a contention period. A single step of collision resolution (i.e., a success or an idle or a collision step of control frames) is allowed in each contention period. The control frames used in each contention period are much smaller than data frames [5].

It has been proved that ICRMA has low throughput; meaning that the number of wasted slots (collided or idle) is high, it also has high average frame's delay especially with small number of sending stations (light load). So using ICRMA protocol; leads to bad network's performance.

The goal of this paper is to develop the ICRMA protocol to improve the network performance, so we propose another medium access protocol; Modified Incremental Collision Resolution Multiple Access (MICRMA) protocol. The main difference between ICRMA and MICRMA protocols is that; MICRMA builds a distributed transmission queue dynamically using a deterministic non binary; instead of binary in ICRMA; tree-splitting. Upon collision occurrence at the beginning of each round; the proposed MICRMA protocol splits the first interval (whole number of stations in the network) into number of subintervals equal to the number of nodes wanting to send, instead of splitting it into two parts in ICRMA protocol.

The rest of the paper is organized as follows. Both the Incremental Collision Resolution Multiple Access (ICRMA) and the proposed Modified Incremental Collision Resolution Multiple Access (MICRMA) protocols are detailed in sections 2 and 3 respectively. Section 4 is devoted to simulation results for both protocols to prove the efficiency of the proposed protocol over the original one. Finally; final conclusions are illustrated and future works are proposed in section 5.

II. Incremental Collision Resolution Multiple Access Protocol (ICRMA)

The incremental Collision Resolution Multiple Access (ICRMA) protocol; is used with a system consisting of a number of stations connected through a single shared communication channel. The channel access is assumed to be slotted; where the slot is the basic unit of transmission and all the stations are synchronized to slot boundaries. The stations monitor the channel continuously and obtain a feedback about the channel status that indicates whether a slot is idle, has a collision or is used for a successful transmission. The feed-back is obtained within the slot duration; hence the duration of a slot is at least equal to the round-trip time in the network [5-7].

A message to be transmitted by a station is made up of several frames. All the frames are of fixed length and each requires one slot for transmission. A station starts transmission at the beginning of a slot boundary, and transmits one frame if no collision occurs. When a collision is detected, the station aborts transmission, such that; only one slot of information is lost. As every frame transmission needs to be followed by an idle period of duration at least equal to the maximum one-way propagation time in the network, the slot duration is fixed to 1.5 times the round-trip time in the network [5-7].

ICRMA first uses control frames for collision resolution. However, as the stations are assumed to have the capability to abort transmission upon collision detection, a recommended change was to use the data frames directly for channel access.

The ICRMA protocol and two detailed examples to explain how it really works are illustrated in the following two subsections respectively.

2.1 ICRMA Protocol

The ICRMA protocol divides the channel access into cycles. Each cycle has a short contention period (one slot) followed by a collision-free transmission period. Stations that are not already in the transmission queue but have a message, transmit a frame during the contention slot. If no collision occurs, the station continues with the frame transmission. The station enters the transmission queue if it has more frames to transmit. If a collision is detected, the contending stations abort transmission. The collision is resolved using a deterministic binary tree-splitting technique in the future contention slots. During the transmission period, the stations in the queue transmit a frame. When transmitting a frame, the station informs if it is going to leave the queue or not after the transmission. The ICRMA protocol can be summarized as follows:

- For an n-stations network; every station in the system has a unique physical identifier from 0 to n-1. It is a unique leaf node of a k-level binary tree; where $n=2^k$. The stations are arranged on the tree using their physical identifier.
- Each station has a stack that holds identifier intervals, denoted by (i,j). The top of the stack entry is the allowable interval. Stations whose identifier lies within this allowable interval are permitted to transmit in the next contention slot.
- Upon a collision, the allowable interval (i,j) is split into two subintervals (i, (i+j+1)/2-1) and ((i+j+1)/2, j), and they replace the top entry in the stack. Depending on which of the two subintervals is pushed last on the stack, the priority is decided. It is assumed that the right subtree is given priority over the left subtree. Therefore, the lower subinterval is pushed first on the stack followed by the higher subinterval.
- When a successful transmission or an idle slot is observed, the allowable interval is popped from the stack.
- When the stack becomes empty, the interval is pushed back onto the stack.
- It is clear that the time (in slots) between two successive insertions of interval (0, n-1) is defined as a binary tree-split cycle (BTS-cycle). Figure 1 shows the arrangement of the stations in a binary tree for a 16-stations network.

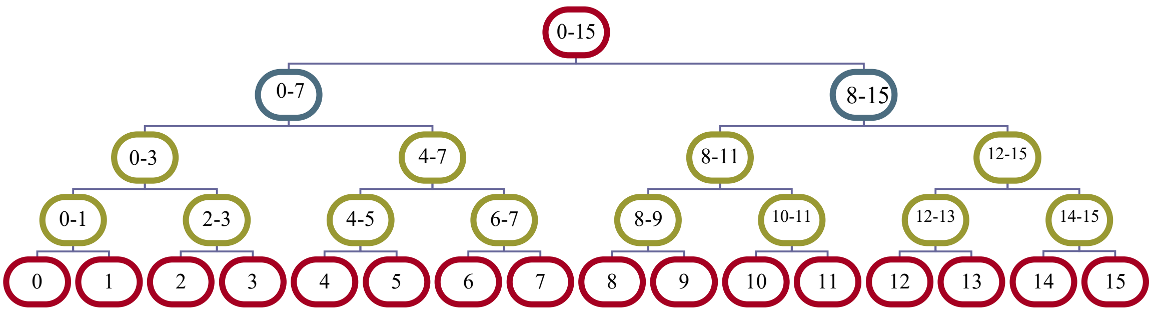


Figure 1   Four-level binary tree for 16 stations

2.2 Example

In this section, two examples are illustrated to explain in details how the ICRMA protocol works.

2.2.1 Example 1

Consider a 16-stations network, with virtual identifier from 0 to 15, stations 1, 3, 6, 7, 8 and 11 have frames to transmit. Figure 1 showed how the 16 stations are organized as a binary tree; where in each level each branch (representing an interval of nodes) is divided into two branches (two subintervals of nodes). Figure 2 explains in details how the ICRMA uses the stack for this example; it shows the stack content, the sending stations and the resulting state (succeeded or collided or idle) for each slot. Figure 3 shows the used part of the whole four-level binary tree (which is subset of figure 1).

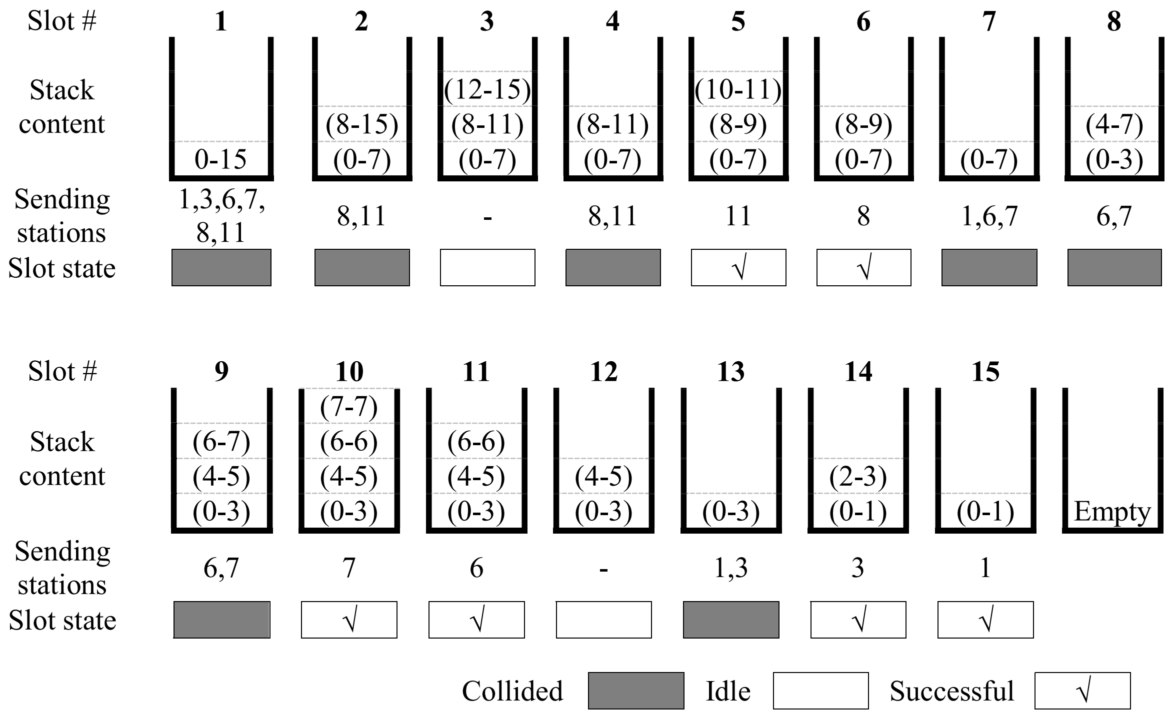


Figure 2 Detailed steps ICRMA takes for example 1

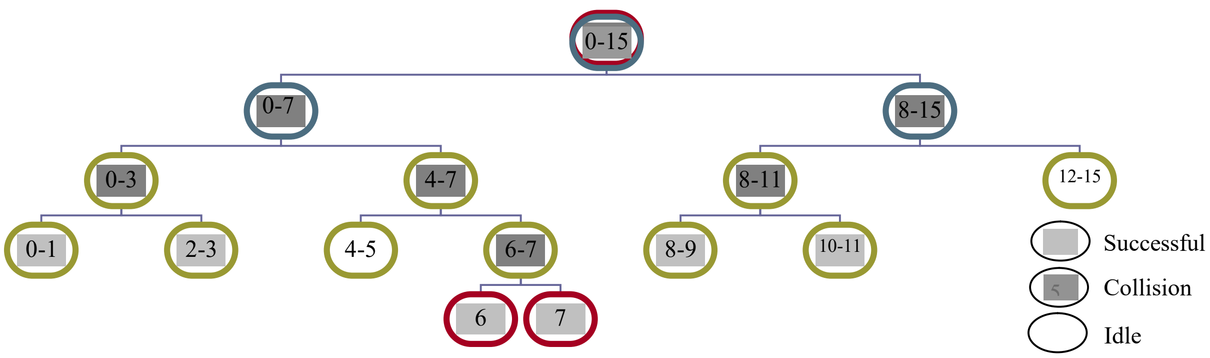


Figure 3 Used part of the 16-leaves binary tree

<u>2.2.2 Example 2</u>

Consider a 32-stations network, with virtual identifier from 0 to 31, stations 0, 6, 9, 14, 15, 18, 27 and 28 have frames to transmit. Figure 4 explains in details how the ICRMA uses the stack for this example; it shows the stack content, the sending stations and the resulting state (succeeded or collided or idle) for each slot. Figure 5 shows the used part of the whole five-level binary tree (representing a 32-leaves BTS).

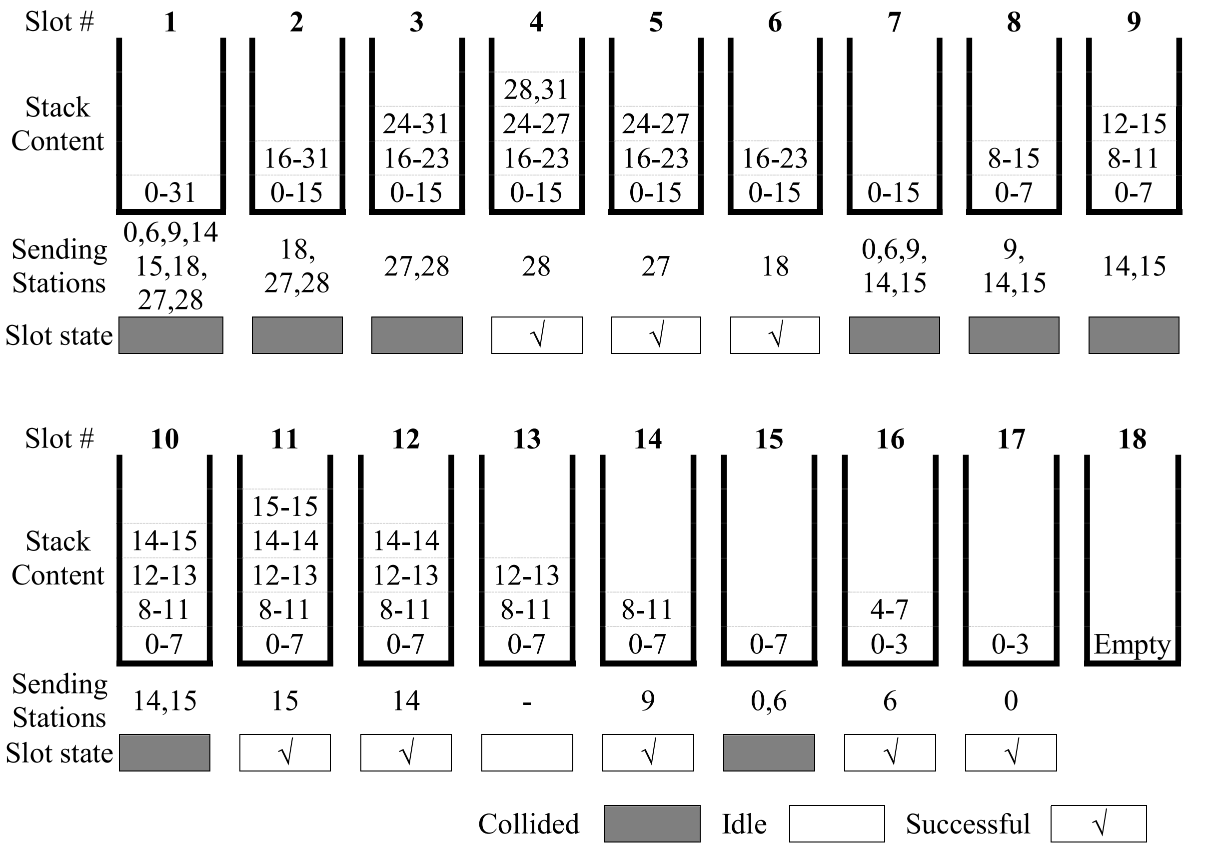


Figure 4 Detailed steps ICRMA takes for example 2

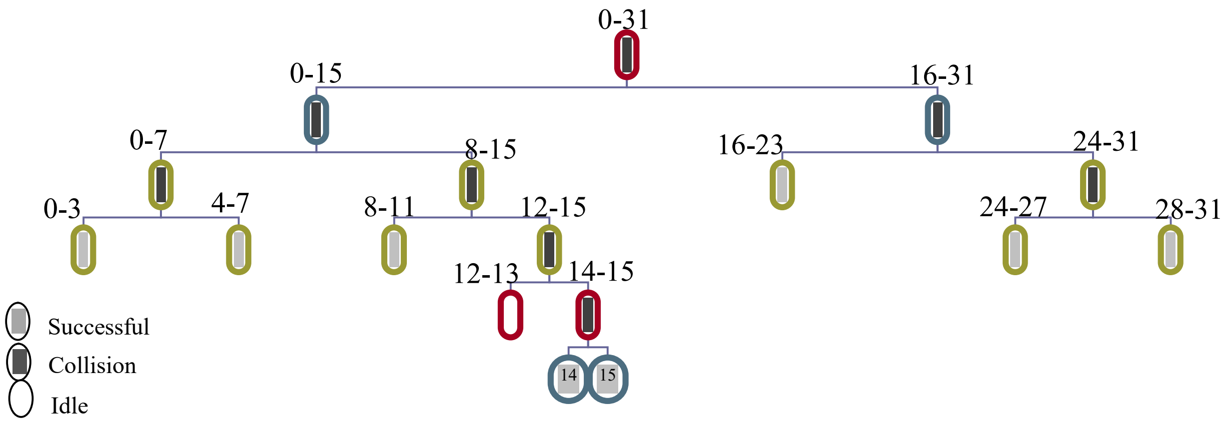


Figure 5 Used part of the 32-leaves binary tree

2.2.3 Comments on the examples

From the two above examples, it is clear that ICRMA protocol has the following drawbacks:

- The ratio between the slots used to send successfully a number of frames and this number is very high (>2), i.e. more than half the slots is wasted (collided or lost), so a lot of the medium's bandwidth is lost.
- Most of the wasted slots are collided slots, so wasting stations' energy beside the waste of the medium's bandwidth.
- Frames are too much delayed, so wasting time.

- **<u>III. Modified Incremental Collision Resolution Multiple Access protocol (MICRMA)</u>**

This protocol; Modified Incremental Collision Resolution Multiple Access protocol (MICRMA); which is a development of the ICRMA protocol; is proposed in order to overcome its drawbacks. It has the following goals:

- Saving the medium's bandwidth, by decreasing the number of wasted slots (i.e. decreasing the ratio between the slots needed to send successfully a number of frames and this number).
- Saving the stations' energy by decreasing the number of collided slots in the wasted slots.
- Saving time by decreasing the frames' delay.

From the ICRMA protocol's examples in section 2.2, we have noticed the following for an n-stations network with s sending nodes:

- The number of levels in the binary tree used is k; where $n=2^k$.
- At least one collision slot is guaranteed at each level, or there will be no need to reach the next level.
- In each level, the set of nodes is split into two sets. In the 2nd level, the probability of having m/2 sending nodes in each set is 50%. In the 3rd level, the probability of having m/4 sending nodes in each set is 25%, and so on.
- Number of needed subintervals (branches of the BTS) is between s and N-1.

Meaning that for example with 4 sending stations in a 16-stations network, collision is guaranteed at both the 1st and 2nd level, at least the 3rd level must be reached to obtain four intervals, so there will be a chance for each station to exist alone in the interval.

The main idea of the MICRMA protocol is; upon collision occurrence at the beginning of each round, split the n nodes into s intervals from the beginning so there will be a great probability that each (or at least some) interval may contain only one sending node, so decreasing the collided slots.

The MICRMA protocol and two detailed examples to explain how it really works are illustrated in the following two subsections respectively.

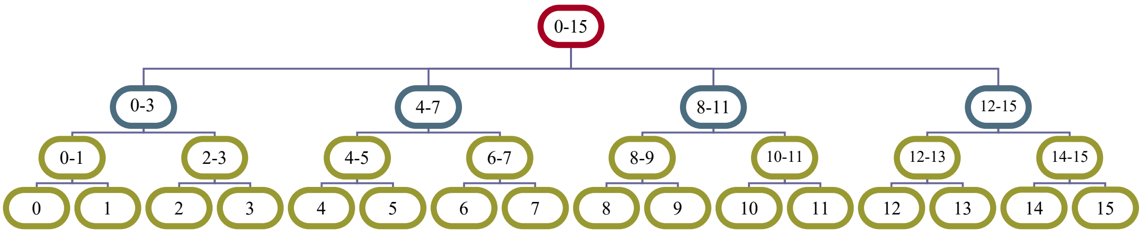
<u>3.1 MICRMA Protocol</u>

The MICRMA protocol divides the channel into rounds; each composed of one or more cycles; the 1st cycle of each round is the one with interval consisting of all stations in the network. As in ICRMA, each cycle is composed of a one slot contention period followed by a queue transmission period. Stations enter and leave the transmission queue in the same way as in the ICRMA protocol (section 2.1). Also, collision treatment is done in tree-splitting but using a non binary technique. The MICRMA protocol can be summarized as follows:

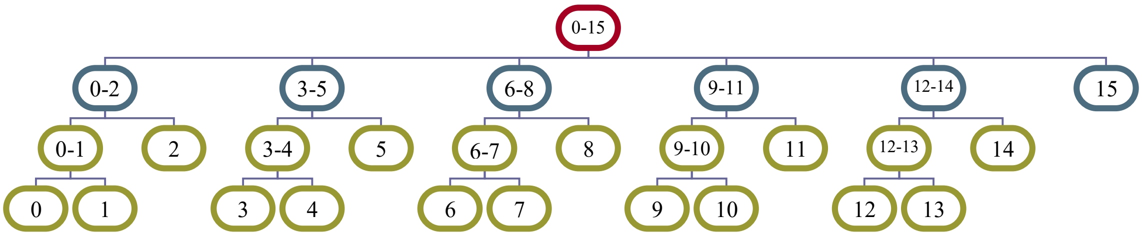
- For an n-stations network; every station in the system has a unique physical identifier from 0 to n-1. It is a unique leaf node of a k-level tree; but in contraire to ICRMA k changes according to the number of nodes wanting to send. The stations are arranged on the tree using their physical identifier.
- Each station has a stack that holds identifier intervals, denoted by (i,j). The top

of the stack entry is the allowable interval. Stations whose identifier lies within this allowable interval are permitted to transmit in the next contention slot.

- Upon first collision at the beginning of each round, the allowable interval (representing the whole number of stations in the network) is split into a number of subintervals; equals to the number of stations wanting to send.
- Upon next collisions in the round, the allowable interval is split into two subintervals. The subintervals replace the top entry in the stack.
- When a successful transmission or an idle slot is observed, the allowable interval is popped from the stack.
- When the stack becomes empty, the interval is pushed back onto the stack.
- The time (in slots) between two successive insertions of interval (0, n-1) is defined as a tree-split cycle. Figures 6.a and 6.b show the arrangement of the stations for a 16-stations network in a tree using MICRMA protocol for if the number of stations wanting to send is 4 and 6 respectively.



(a) Number of sending stations is equal to 4



(a) Number of sending stations is equal to 6

Figure 6 Organization of 16 stations into a tree using MICRMA

The common points between ICRMA and MICRMA protocols are:

- For an n-stations network; every station in the system has a unique physical identifier from 0 to n-1. It is a unique leaf node of a k-level tree. The stations are arranged on the tree using their physical identifier.
- Each station has a stack that holds identifier intervals, denoted by (i,j). The top of the stack entry is the allowable interval. Stations whose identifier lies within this allowable interval are permitted to transmit in the next contention slot.
- Upon collision, the allowable interval is split into a number of subintervals; which replace the top entry in the stack; the lower subinterval is pushed first on the stack followed by the higher subintervals.
- When a successful transmission or an idle slot is observed, the allowable interval is popped from the stack.
- When the stack becomes empty, the interval is pushed back onto the stack.

The different points between ICRMA and MICRMA protocols are:

- The stations are organized in a k-levels non-binary tree, where k changes according to the number of nodes wanting to send.
- Upon first collision at the beginning of each round, i.e. in the first level of the tree; the allowable interval (representing the whole number of stations) is divided into s subintervals (branches), where s is the number of stations wanting to send.
- Upon next collisions in the round; i.e. in all next levels of the tree; stations in each interval are divided into two subintervals (branches).
- Upon splitting an interval; the number of stations in the last sub-interval may be less than or equal to that's in the other sub-intervals.
- If the number of sending stations exceeds 70% of the whole number of stations, it reverts to traditional TDMA protocol; i.e. each station is allocated a slot in order; whether or not it has a frame to send or not.

3.2 Examples

In this section, the same examples used in 2.2 are illustrated to explain in details how the MICRMA protocol works.

3.2.1 Example 1

Consider the same 16-stations network in example 2.2.1, where same stations 1, 3, 6, 7, 8 and 11 have frames to transmit. Figure 6.b showed how these stations are organized as a tree using MICRMA protocol. In the first level; the main set (whole number of nodes) is split into number of branches (sub-intervals) equal to the number of stations wanting to transmit; i.e. 6. In all other levels each branch (interval of node) is split into two branches (subintervals).

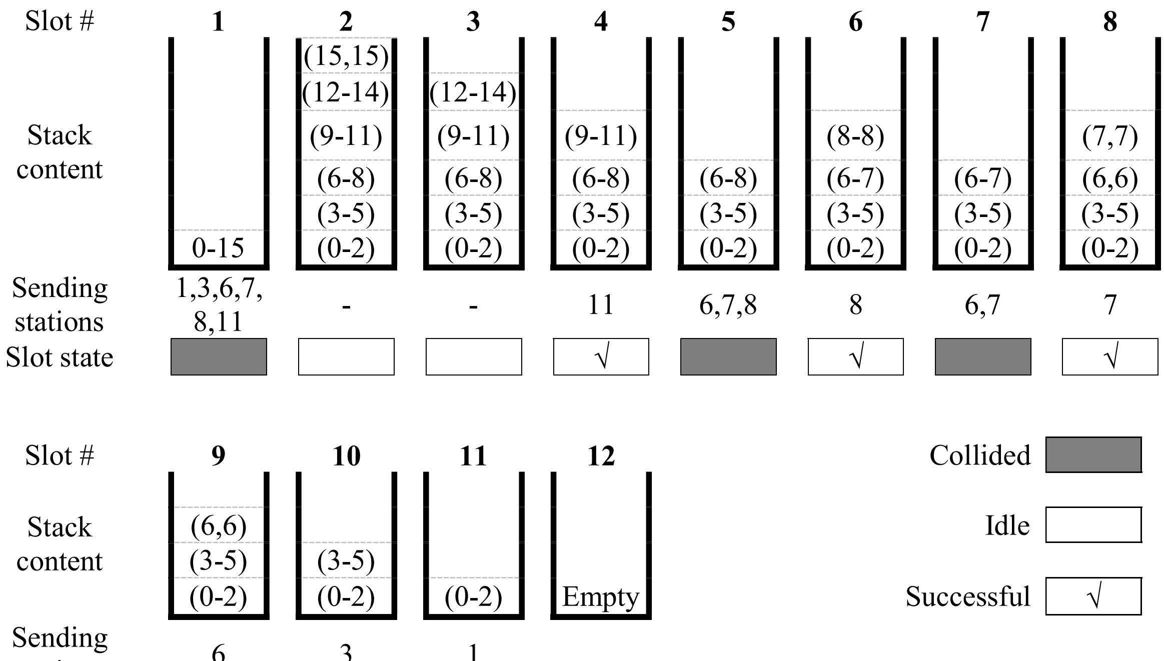


Figure 7 Detailed steps MICRMA takes for example 1

Figure 7 explains in details how the MICRMA uses the stack for this example; it shows the stack content, the sending stations and the resulting state (succeeded or collided or idle) for each slot. Figure 8 shows the used part (a subset) of the whole tree in figure 6.b.

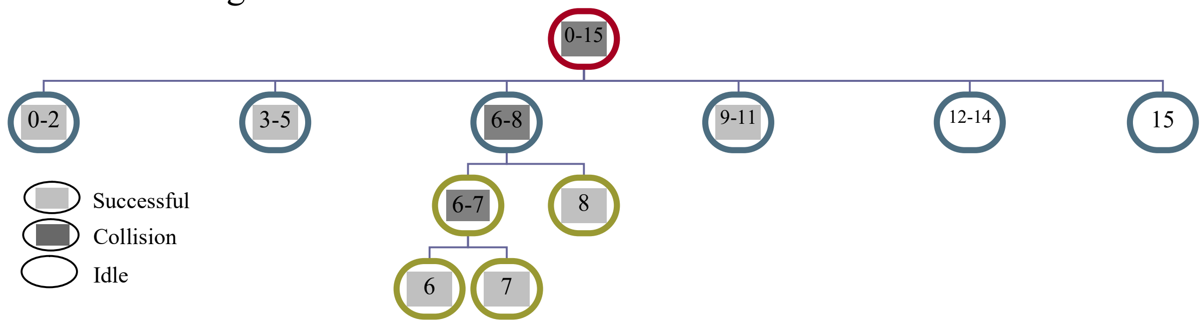


Figure 8 Used part of the 16-leaves non-binary tree in figure 6.b

3.1.2 Example 2

Consider the same 32-stations network in example 2.1.2, with virtual identifier from 0 to 31, where same stations 0, 6, 9, 14, 15, 18, 27 and 28 have frames to transmit. Figure 9 explains in details how the MICRMA uses the stack for this example; it shows the stack content, the sending stations and the resulting state (succeeded or collided or idle) for each slot. Figure 10 shows the used part of the 32-leaves non-binary tree (correspondent to the number of nodes in the network).

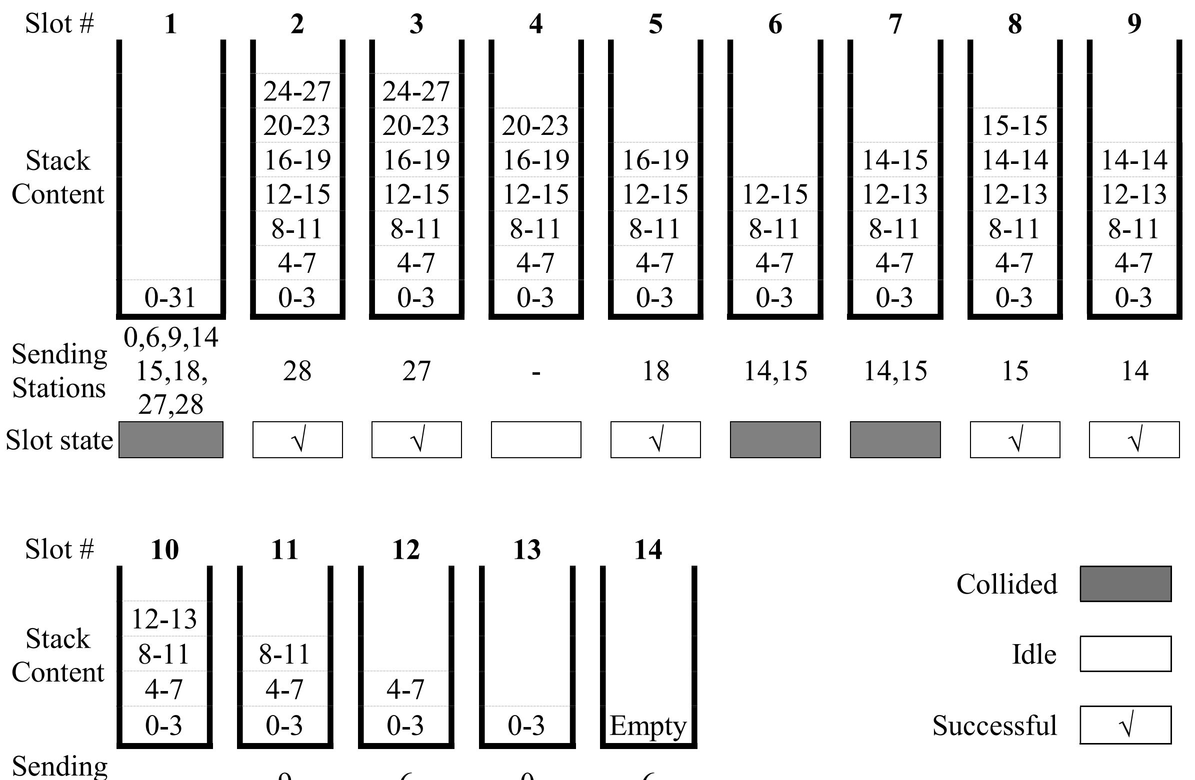


Figure 9 Detailed steps MICRMA takes for example 2

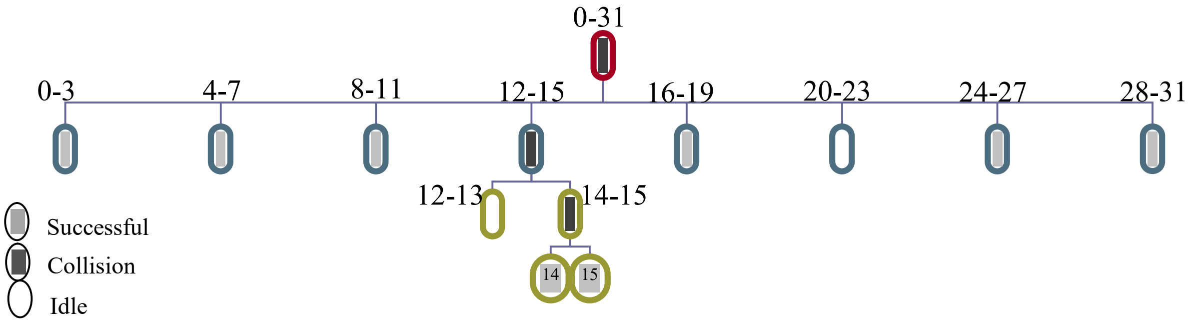


Figure 10 Used part of the 32-leaves non-binary tree

3.2.3 Comments on the examples

From the two above examples, it is clear that MICRMA protocol has reached the goals illustrated in the beginning of the section:

- The ratio between the slots needed to send successfully a number of frames and this number is much less than the ICRMA protocol, i.e. it reduces the number of wasted slots, so it saves the medium's bandwidth.
- It decreases the number of collided slots in the wasted slots, so it saves the stations' energy.
- It decreases the frames' delay, so it saves time.

IV. Simulation results

This section is divided into three subsections. The first one illustrates the performance parameters used to evaluate both protocols. The second one illustrates the simulation results that prove the efficiency of the proposed MICRMA protocol over the original one; ICRMA. The third one analyses the results obtained in the second subsection.

4.1 Performance parameters

The following six parameters are calculated for each protocol alone; to prove the efficiency of the proposed MICRMA over the original ICRMA.

4.1.1 Percentage of used slots to needed ones

This parameter calculates the percentage of the slots (succeeded or collided or idle) used to send successfully a number of frames to this number.

$$Percentage\ of\ used\ slots\ to\ number\ of\ frames = \frac{Total\ number\ of\ used\ slots}{Total\ number\ of\ frames} x100$$

4.1.2 Percentage of idle slots

This parameter calculates the percentage of the idle slots (i.e. the slots in which no station sends) to the total number of slots (succeeded or collided or idle) the stations use to send their frames successfully.

$$Percentage\ of\ idle\ slots = \frac{Number\ of\ idle\ slots}{Total\ number\ of\ used\ slots} x100$$

4.1.3 Percentage of collided slots

This parameter calculates the percentage of the collided slots (i.e. the slots in which more than one station sends) to the total number of slots (succeeded or collided or idle) the stations use to send their frames successfully.

$$Percentage\ of\ collided\ slots = \frac{Number\ of\ collided\ slots}{Total\ number\ of\ used\ slots} x100$$

4.1.4 Percentage of succeeded slots

This parameter calculates the percentage of the succeeded slots (i.e. the slots in which only one station sends) to the total number of slots (succeeded or collided or idle) the stations use to send their frames successfully.

$$Percentage\ of\ succeeded\ slots = \frac{Number\ of\ succeeded\ slots}{Total\ number\ of\ used\ slots} x100$$

4.1.5 Average frame delay

This parameter calculates the average time between each station wants to send a frame and the time it is really sent (using a successful slot).

$$Average\ frame's\ delay = \frac{\sum_{\forall\ frames}\left(Arrival\ time - Sent\ time\right)}{Total\ number\ of\ frames}$$

4.1.6 Slots' distribution

This parameter summarizes the slots usage; it shows the distribution of used slots; succeeded or wasted (idle or collided) slots.

$$slots'\ distribution = Number\ of\ slots\ in\ a\ state_{\substack{State \in Used(\ suceeded,\ wasted(collided,\ dle)) \\ Used\ slots = Succeeded\ slots + Wasted\ slots \\ Wasted\ slots = collided\ slots + idle\ slots}}$$

4.2 Simulation' results

This section shows the simulation results obtained to compare the two MAC protocols, the original ICRMA and the proposed MICRMA.

A simulation program was built using C++ language; comparisons are done using two networks; 50-nodes and 100-nodes networks. Resulted parameters are the average of 2500 rounds where; for each round; [1]Sending nodes are chosen randomly. [2]For each sending node, four options are used for the sent number of frames; exactly one, exactly three, exactly five and a random number between one and five. Percentage number of sending nodes varies from 10% to 100%. However, in order to obtain correct results, we used the same information (number of node, sending nodes and number of sent frames) for the two protocols. Considering networks with a round-trip time of 54 μs; the slot duration is 81μs[5]. The eight following figures; from 11 to 18; are used to plot the average value of the performance parameters illustrated in the previous section, each figure is composed of seven figures; labeled from a to g. Table 1 illustrates these figures' descriptions.

<table>
<tr><th>Figure #</th><th># of sent frames by a sending node</th><th># of nodes in the net</th><th>Figure description</th></tr>
<tr><td>11</td><td>1</td><td rowspan="4">50</td><td rowspan="8">(a) Percentage of used slots to number of frames for each protocol.
(b) Percentage of idle to used slots for each protocol.
(c) Percentage of collided to used slots for each protocol.
(d) Percentage of succeeded to used slots for each protocol.
(e) Average frame delay for each protocol.
(f) Slots distribution for ICRMA protocol; used (succeeded- wasted- (collided- idle)).
(g) Slots distribution for MICRMA protocol; used (succeeded- wasted- (collided- idle)).</td></tr>
<tr><td>13</td><td>3</td></tr>
<tr><td>15</td><td>5</td></tr>
<tr><td>17</td><td>Random from 1 to 5</td></tr>
<tr><td>12</td><td>1</td><td rowspan="4">100</td></tr>
<tr><td>14</td><td>3</td></tr>
<tr><td>16</td><td>5</td></tr>
<tr><td>18</td><td>Random from 1 to 5</td></tr>
</table>

Table 1 Figures 11 to 18 descriptions

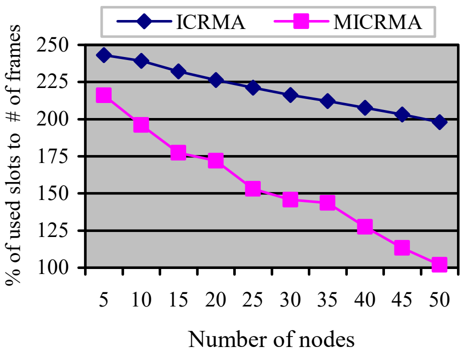
4.3 Analysis of the results

From results plotted in figures 11 to 18, the following are concluded:

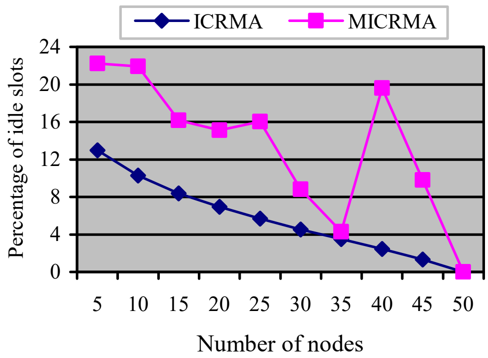
- Each two successive figures have almost identical shapes, i.e. both protocols' performance doesn't depend on the number of nodes in the network, but depends on the number of sending nodes in the network.
- The ratio between the slots used to send successfully a number of frames and this number in MICRMA protocol is much less than that used in ICRMA protocol ('a' figures), i.e. the number of wasted slots in MICRMA protocol ('g' figures) is less than that of the ICRMA protocol ('f' figures). So MICRMA saves the medium's bandwidth.

  For both protocols, the ratio between the slots used to send successfully a number of frames and this number decreases as the number of sending stations increases ('a' figures), i.e. percentage of wasted slots for both protocol decreases as the number of sending stations increases. But the decreasing rate is much higher in MCIRMA protocol than in ICRMA protocol, i.e. with the increase of sending nodes; the percentage of wasted to used slots in the MICRMA protocol decreases with a much greater rate than in the ICRMA protocol.
- The percentage of succeeded to used slots is much higher for the MICRMA protocol than for the ICRMA protocol ('b' figures).

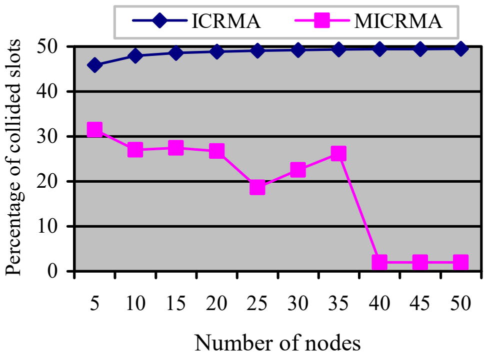
  For both protocols, the percentage of succeeded to used slots ('b' figures) increases as the number of sending stations increases. But the increasing rate is much higher in MCIRMA protocol than in ICRMA protocol, i.e. with the increase of sending nodes; the percentage of succeeded to used slots in the MICRMA increases with a much greater rate than in the ICRMA protocol.



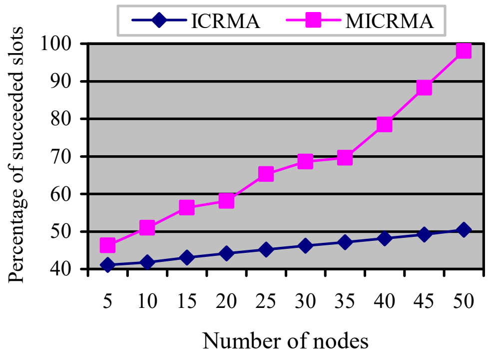
(a) Percentage of used slots to number of frames for both protocols



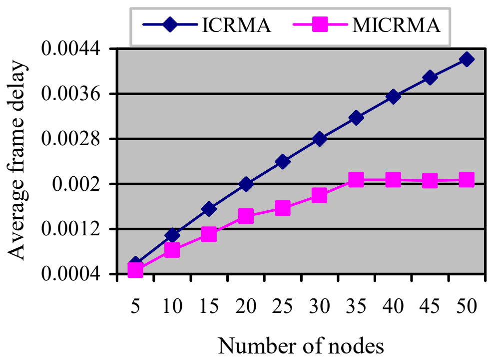
(b) Percentage of idle to used slots for both protocols



(c) Percentage of collided to used slots for both protocols



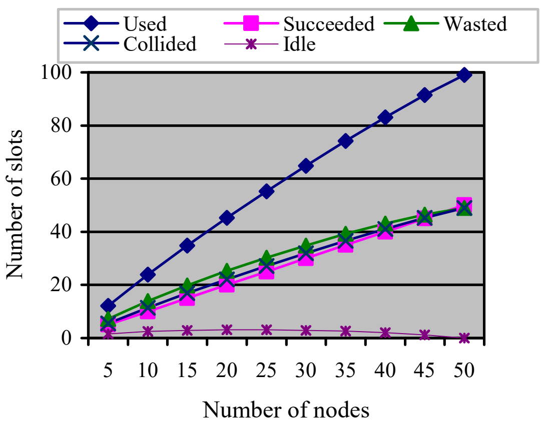
(d) Percentage of succeeded to used slots for both protocols



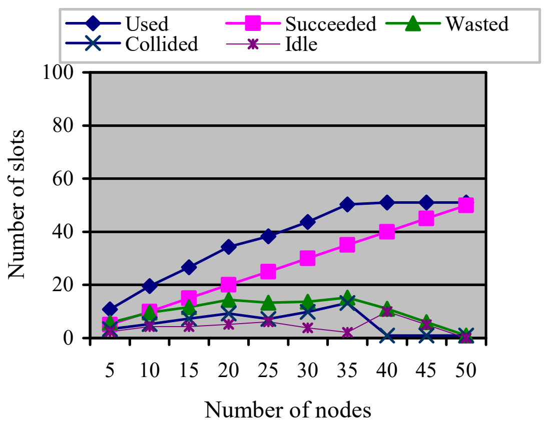
(e) Average frame delay for both protocols

Figure 11

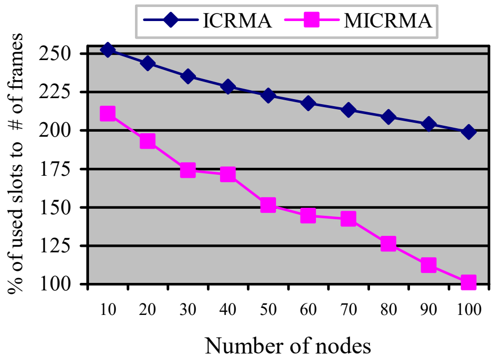
Network consists of 50 nodes. Transmitting nodes are selected randomly. Each transmitting node sends only one frame at a time.



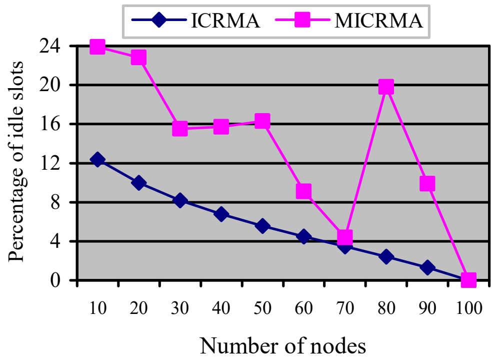
(f) ICRMA Slots' distribution



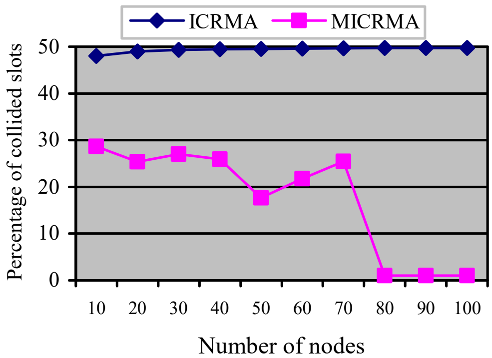
(g) MICRMA Slots' distribution



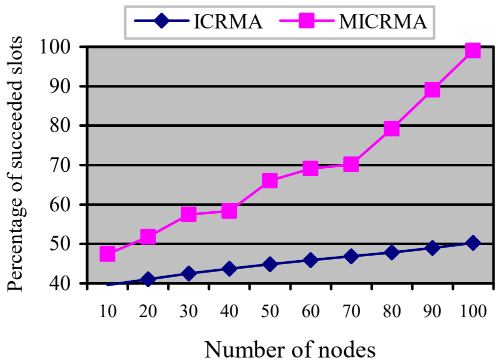
(a) Percentage of used slots to number of frames for both protocols



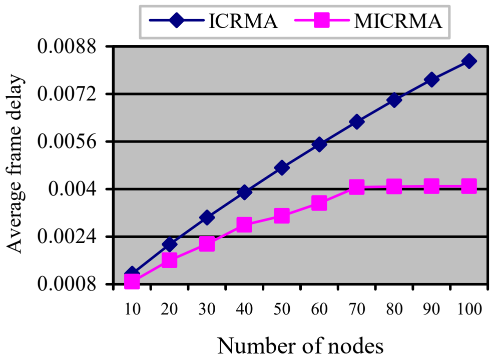
(b) Percentage of idle to used slots for both protocols



(c) Percentage of collided to used slots for both protocols



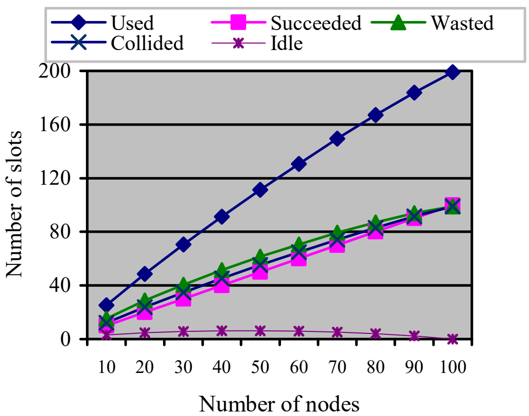
(d) Percentage of succeeded to used slots for both protocols



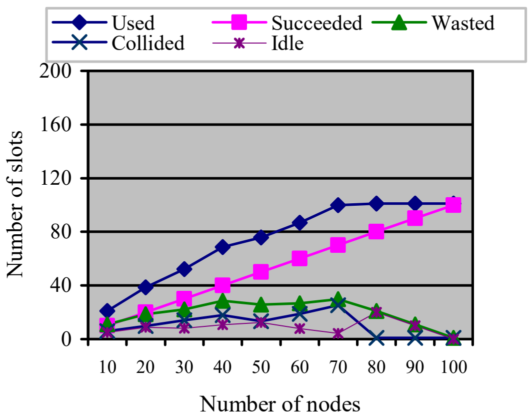
(e) Average frame delay for both protocols

Figure 12

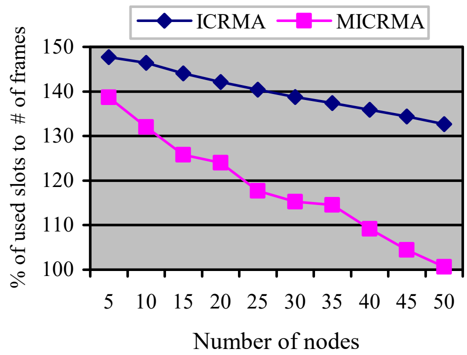
Network consists of 100 nodes. Transmitting nodes are selected randomly. Each transmitting node sends only one frame at a time.



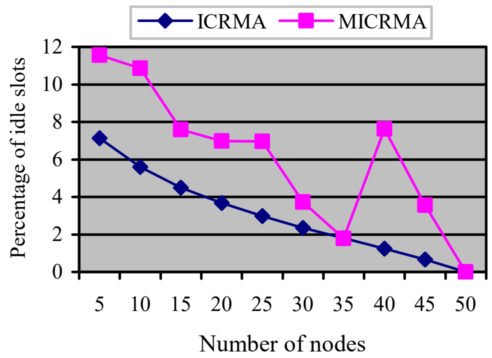
(f) ICRMA Slots' distribution



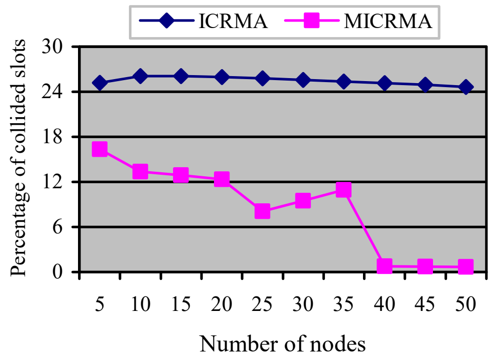
(g) MICRMA Slots' distribution



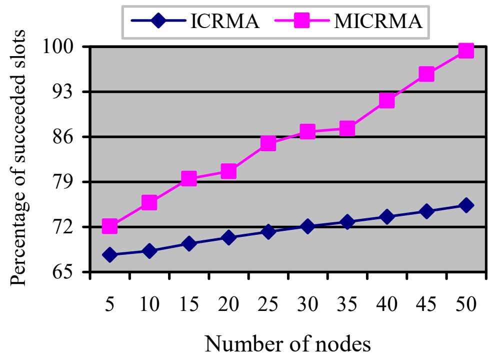
(a) Percentage of used slots to number of frames for both protocols



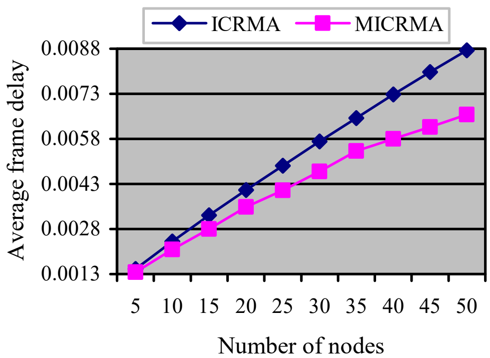
(b) Percentage of idle to used slots for both protocols



(c) Percentage of collided to used slots for both protocols



(d) Percentage of succeeded to used slots for both protocols

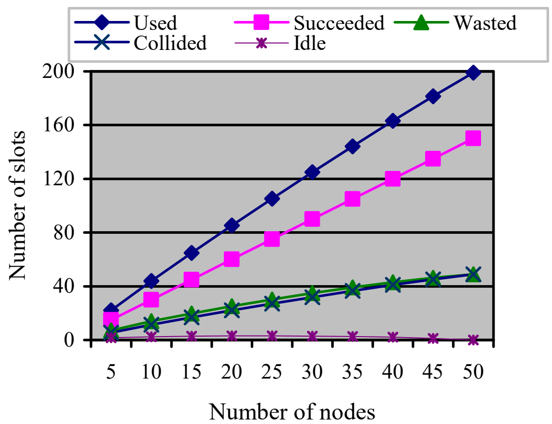


(e) Average frame delay for both protocols

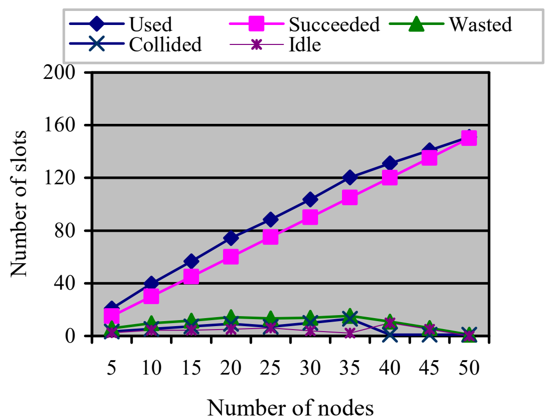
8-15

Figure 13

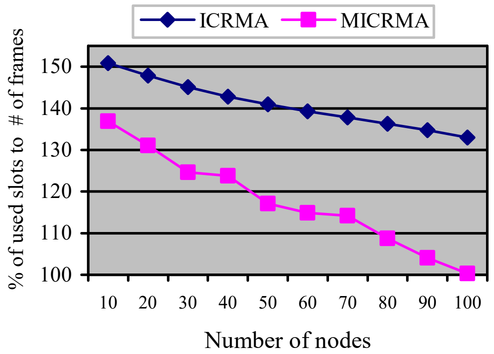
Network consists of 50 nodes. Transmitting nodes are selected randomly. Each transmitting node sends exactly three frames at a time.



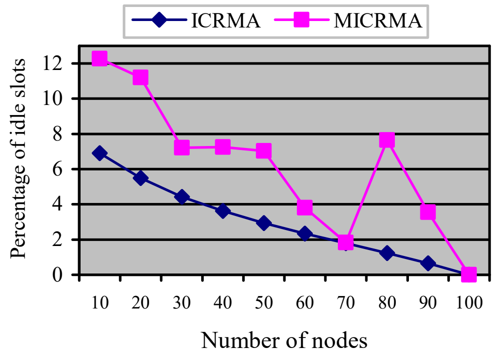
(f) ICRMA Slots' distribution



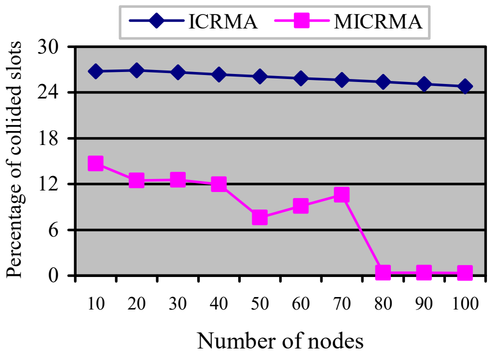
(g) MICRMA Slots' distribution



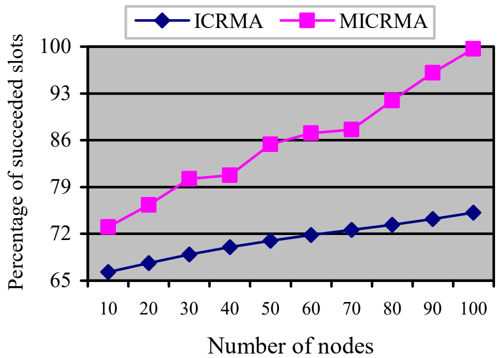
(a) Percentage of used slots to number of frames for both protocols



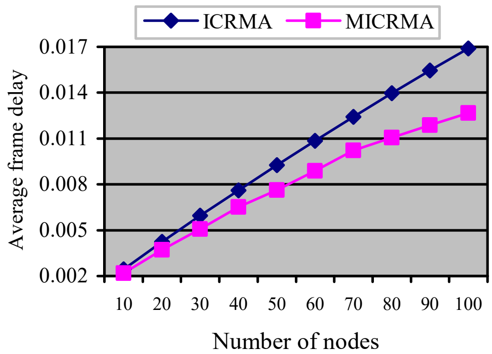
(b) Percentage of idle to used slots for both protocols



(c) Percentage of collided to used slots for both protocols



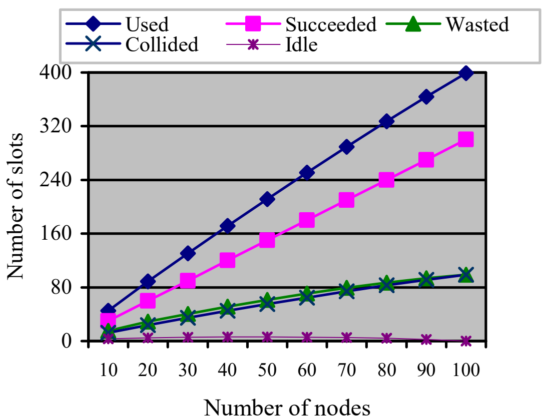
(d) Percentage of succeeded to used slots for both protocols



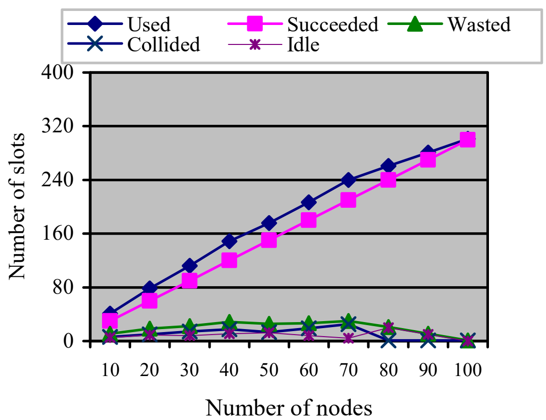
(e) Average frame delay for both protocols

Figure 14

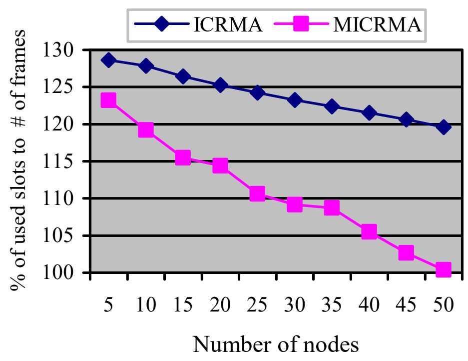
Network consists of 100 nodes. Transmitting nodes are selected randomly. Each transmitting node sends exactly three frames at a time.



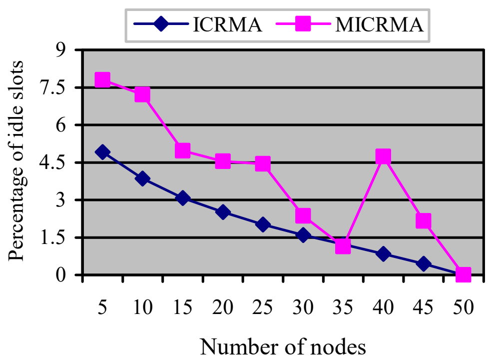
(f) ICRMA Slots' distribution



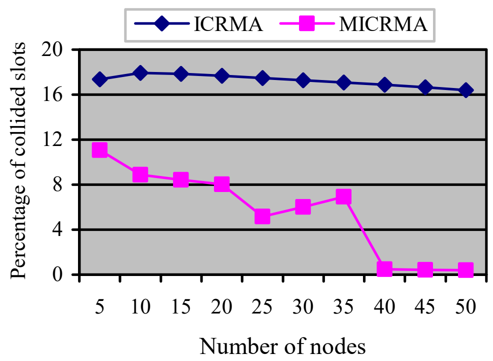
(g) MICRMA Slots' distribution



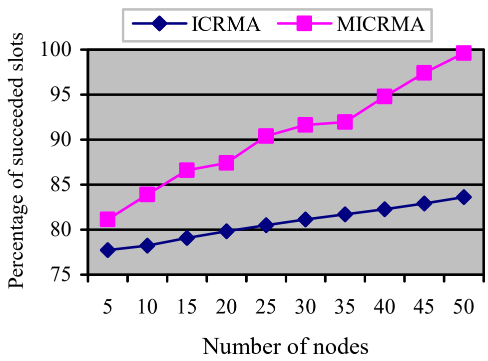
(a) Percentage of used slots to number of frames for both protocols



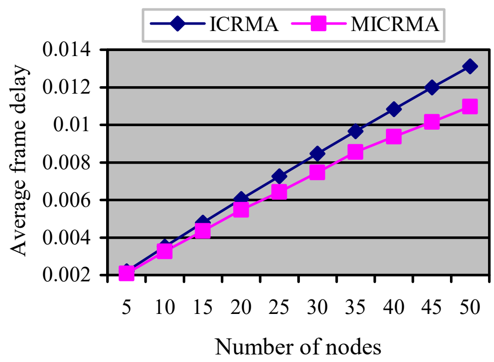
(b) Percentage of idle to used slots for both protocols



(c) Percentage of collided to used slots for both protocols



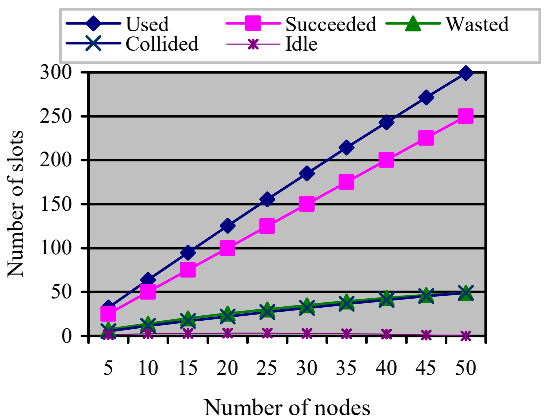
(d) Percentage of succeeded to used slots for both protocols



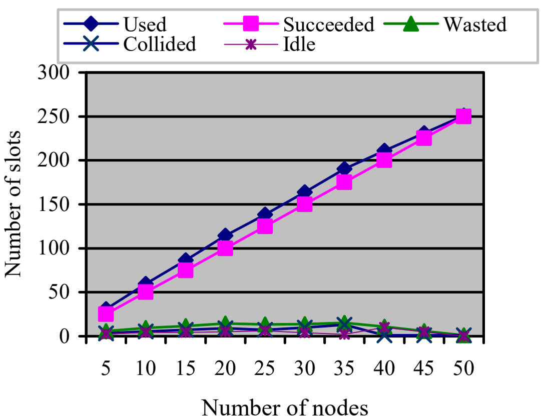
(e) Average frame delay for both protocols

Figure 15

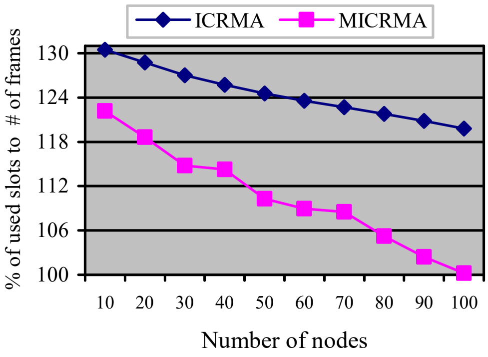
Network consists of 50 nodes. Transmitting nodes are selected randomly. Each transmitting node sends exactly five frames at a time.



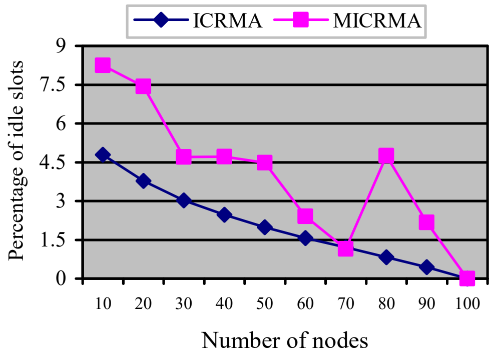
(f) ICRMA Slots' distribution



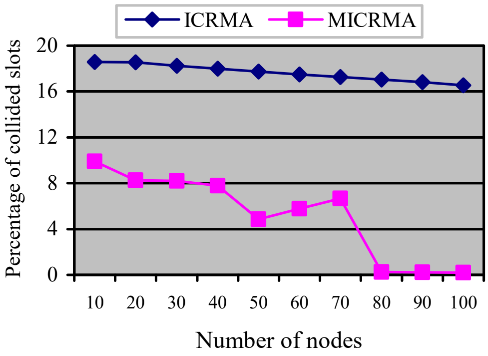
(g) MICRMA Slots' distribution



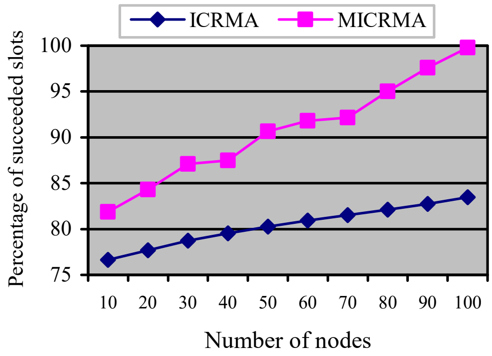
(a) Percentage of used slots to number of frames for both protocols



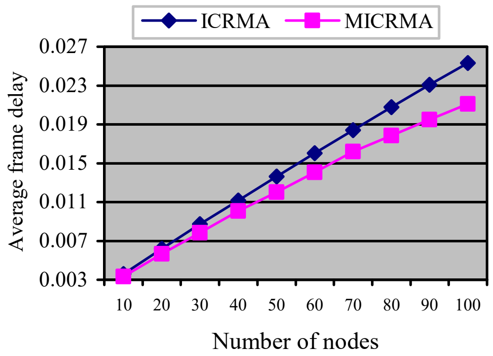
(b) Percentage of idle to used slots for both protocols



(c) Percentage of collided to used slots for both protocols



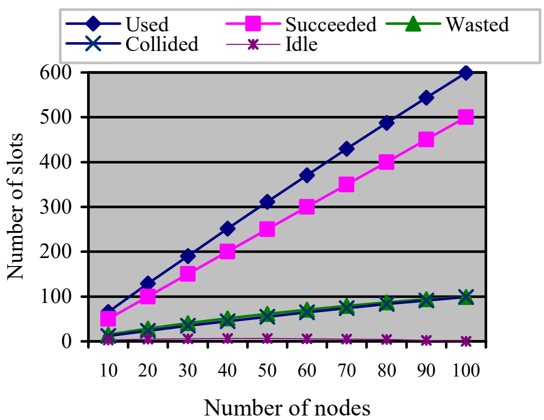
(d) Percentage of succeeded to used slots for both protocols



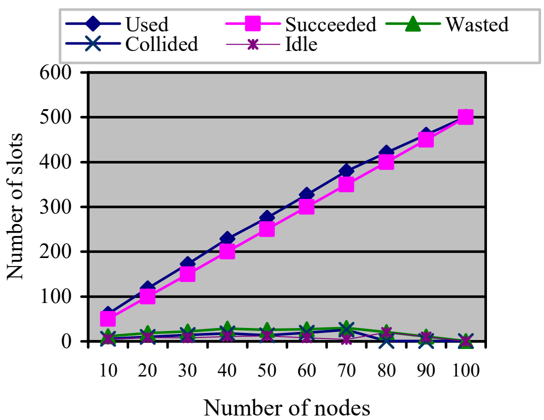
(e) Average frame delay for both protocols

Figure 16

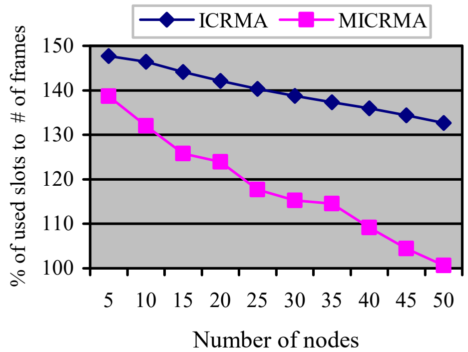
Network consists of 100 nodes. Transmitting nodes are selected randomly. Each transmitting node sends exactly five frames at a time.



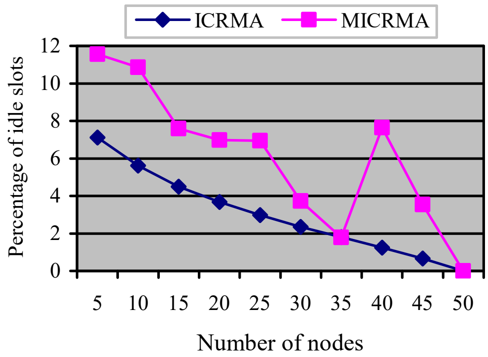
(f) ICRMA Slots' distribution



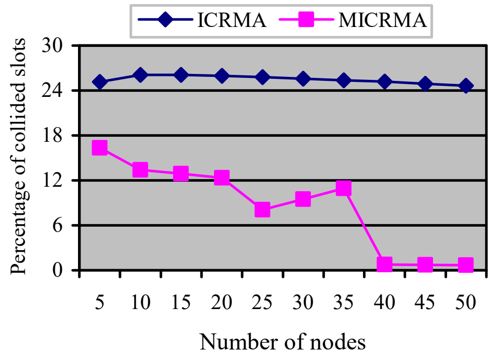
(g) MICRMA Slots' distribution



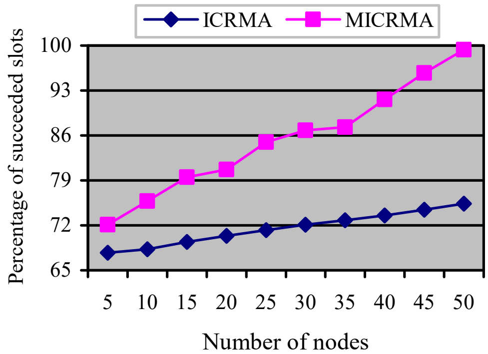
(a) Percentage of used slots to number of frames for both protocols



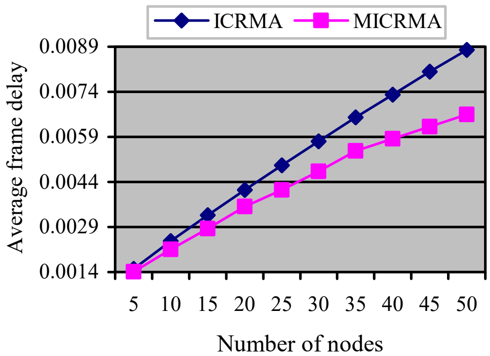
(b) Percentage of idle to used slots for both protocols



(c) Percentage of collided to used slots for both protocols



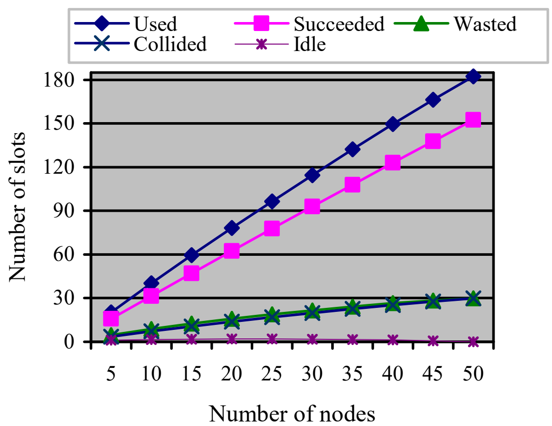
(d) Percentage of succeeded to used slots for both protocols



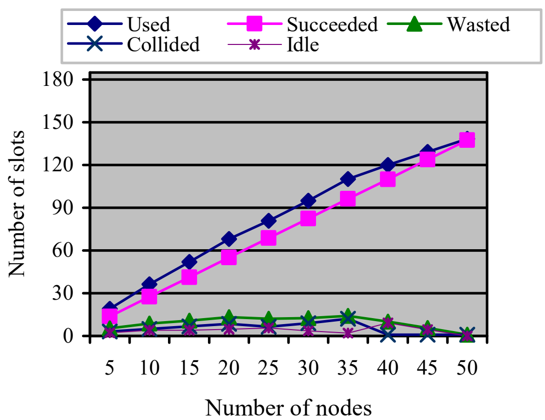
(e) Average frame delay for both protocols

Figure 17

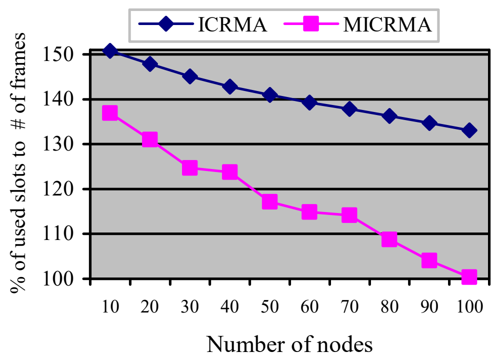
Network consists of 50 nodes. Transmitting nodes are selected randomly. Each transmitting node is allowed to send from one to five frames at a time.



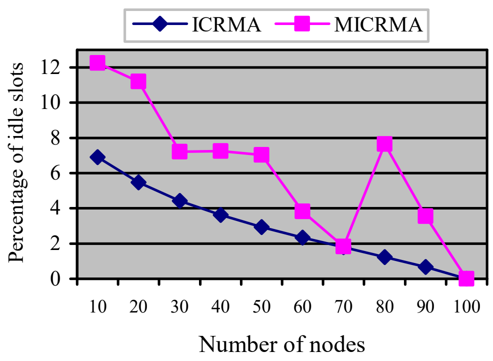
(f) ICRMA Slots' distribution



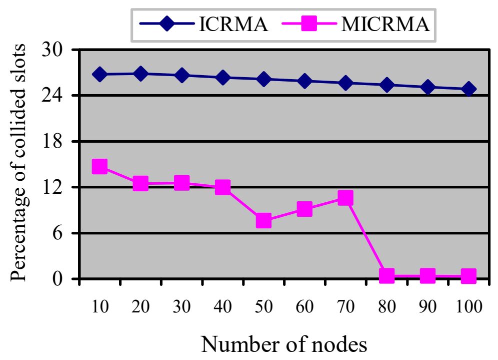
(g) MICRMA Slots' distribution



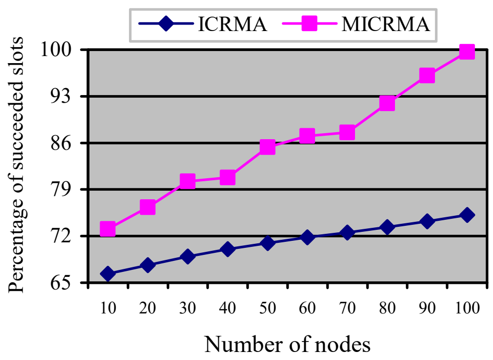
(a) Percentage of used slots to number of frames for both protocols



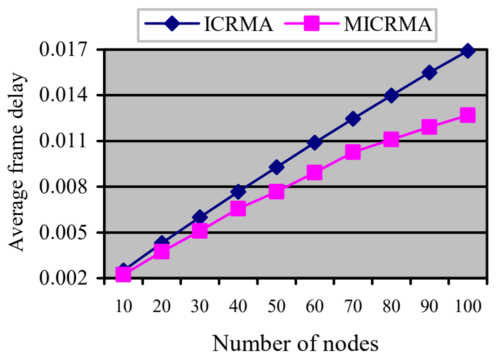
(b) Percentage of idle to used slots for both protocols



(c) Percentage of collided slots to used ones for both protocols



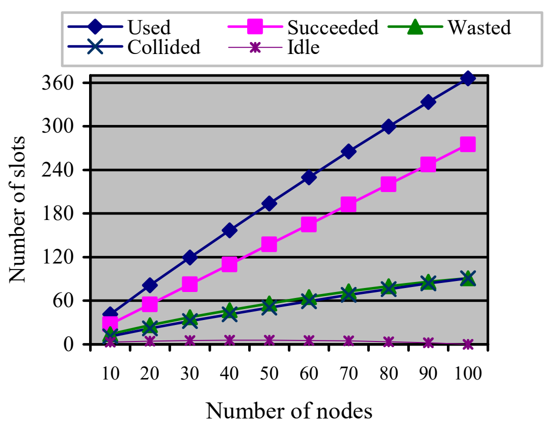
(d) Percentage of succeeded to used slots for both protocols



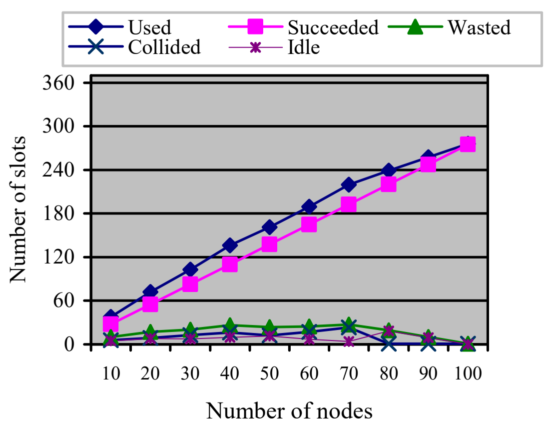
(e) Average frame delay for both protocols

Figure 18

Network consists of 100 nodes. Transmitting nodes are selected randomly. Each transmitting node is allowed to send from one to five frames at a time.



(f) ICRMA Slots' distribution



(g) MICRMA Slots' distribution

- The average frame's delay in MICRMA protocol is much less than the ICRMA protocol ('e' figures). The difference in this delay increases as the number of sending stations increases, i.e. the average frame' delay increases with a slower rate in the MICRMA protocol than in the ICRMA protocol ('e' figures).
- For ICRMA protocol, most of the wasted slots are collided slots not idle slots ('f' figures); this leads to high wastage of stations' energy beside the high wastage of slots (medium's bandwidth). While for MICRMA protocol, the numbers of collided slots and idle slots are close ('g' figures), so much less wastage of stations' energy.
- Both protocol's performance improves with the increases of the number of sent frames by each sending stations for each cycle (all figures).

From all the above we conclude that the proposed MICRMA protocol performs better than the original ICRMA in all cases:

- The ratio between the slots used to send successfully a number of frames and this number in MICRMA protocol is much lower than in ICRMA protocol, i.e. MICRMA reduces the number of wasted slots, so it saves the medium's bandwidth.
- For the used slots; [1]The percentage of succeeded to used slots in MICRMA protocol is much higher than that in ICRMA protocol. [2]The percentage of collided to used slots in MICRMA protocol is much lower than that in ICRMA protocol, so it saves the stations' energy (which is so important in wireless).
- Average frame's delay in MICRMA protocol is much lower than ICRMA protocol, so it saves time.
- As the number of sending stations increases; i.e. as the network's load increases, the network's performance that uses MICRMA improves with a higher rate than that uses the ICRMA protocol.

V. Summary and Conclusion

In this paper the Modified Incremental Collision Resolution Multiple Access (MICRMA); which is a development of the ICRMA; is a MAC protocol proposed to improve the network's performance. Both ICRMA and MICRMA protocols dynamically divide the medium into cycles of variable length; each cycle consists of a contention period and a queue-transmission period. During the contention period, stations with frames to send compete for the right to be added to the data-transmission queue using a deterministic tree-splitting. The main difference between ICRMA and MICRMA protocols is that; upon collision occurrence at the beginning of each round; the proposed MICRMA protocol splits the first interval (whole number of stations) into number of subintervals equal to the number of nodes wanting to send, instead of splitting it into two parts in the ICRMA protocol. Meaning that; ICRMA protocol builds the distributed transmission queue using a deterministic binary tree-splitting; while MICRMA builds it using non-binary-tree splitting.

A simulation program was built to evaluate the MICRMA protocol. Six parameters were calculated and plotted for both protocols; percentage of used slots to needed ones, percentage of idle, collided and succeeded slots, average frame delay and slots' distribution. Comparisons are done using two networks; one consists of 50 nodes and the other consists of 100 nodes. Resulted parameters are the average of 2500 rounds with randomly selected inputs.

Results proved that the proposed MICRMA protocol performs better than the original ICRMA in all cases; where: [1]The ratio between the slots used to send successfully a number of frames and this number in MICRMA protocol is much lower than in ICRMA protocol, i.e. MICRMA decreases the number of wasted slots, so saving the medium's bandwidth. [2]The percentage of succeeded to used slots in MICRMA protocol is much higher than that in ICRMA protocol. [3]The percentage of collided to used slots in MICRMA protocol is much lower than that in ICRMA protocol, so saving the stations' energy (which is so important in wireless networks). [4]The average frame's delay in MICRMA protocol is much lower than ICRMA protocol, so saving time. [5]Also, as the number of sending stations increases; i.e. as the network's load increases; the MICRMA protocol performs better with a higher rate than the ICRMA protocol.

As a future work, we will try to prove the efficiency of the MICRMA over the ICRMA by using a mathematical model for each protocol.

REFERENCES

[1] Ryan G. Lane, "Network Protocols", http://ww2.cs.fsu.edu/~ryanlane/papers/NWproto.html

[2] Philip Koopman, "10 Embedded Communication Protocols", October 2004 http://www.ece.cmu.edu/~ece549/lectures/10_protocols.pdf

[3] Telematics l, "Medium Access Control", 2005, http://www.tu-ilmenau.de/fakia/fileadmin/template/startIA/telematik/lehre/Telematik_I/Handouts/05_MAC.pdf

[4] Younggoo Kwon, Yuguang Fang and Haniph Latchman, "A Novel MAC Protocol with Fast Collision Resolution for Wireless LANs," Proceeding of IEEE INFOCOM 2003.

[5] R. Garcés and JJ. Garcia-Luna-Aceves, "A Near-Optimum Channel Access Protocol Based on Incremental Collision Resolution and Distributed Transmission Queues," Proceeding of IEEE INFOCOM, 1998, pp. 158–165.

[6] M.Meenakshi and S. Malarvizhi, 'A Near Optimum MAC Protocol based on the Incremental Collision Resolution Multiple Access Protocol for CDMA based communication system', IEEE International Symposium on Wireless Personal Multimedia Communications, Hawaii , October 2002.

[7] R. Srinivasan and Arun K. Somani, "On achieving fairness and efficiency in high-speed shared medium access," IEEE/ACM Transactions on Networking, vol.11, no.1, pp. 111-124, February 2003.

[8] László Györfi and Sándor Győri, "Analysis of tree algorithm for collision resolution", Proceeding of the 2005 International Conference on Analysis of Algorithms, pp. 357-364.