Proceedings of the 5th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Madrid, Spain, February 15-17, 2006 (pp40-44)

Multimedia Task Scheduling on OLSR enabled Ad hoc Networks

Adeel Akram, Shahbaz Pervez, Shoab. A. Khan
University of Engineering and Technology Taxila, Pakistan

http://www.uettaxila.edu.pk

Abstract: - *In this paper we present a consolidated scheduling and routing solution to the multimedia ad hoc network problem. We consider a generic ad hoc network with all kinds of heterogeneous devices. These devices are known to have different capabilities and capacities in terms of system resources and network connection speeds. We propose a task partitioning and scheduling mechanism which results in distribution of N tasks with various real-time constraints and work loads on M heterogeneous devices.It has been shown that in case of M=1 an optimal solution to task scheduling can be obtained with computational complexity O(N) however for M≥2 the scheduling algorithm can be demonstrated to be NP-hard by reducing it to the well known NP-Complete bin-packing problem The mechanism proposed in this paper considers device capabilities, Link capacities and task constraints to find a near optimal solution to the multimedia scheduling problem in Ad hoc networks.*

Key-Words: - Multimedia, Task Partitioning, Real-time constrained task scheduling

1. Introduction

Ad hoc wireless networks have been a topic of research for more than a decade now. Most of the research was done on using Ad hoc networks in areas such as disaster recovery, military assignments and scenarios where rapid deployment of networks is required in the absence of conventional infrastructure based network. The focus was to provide communication setup as a temporary measure in areas where deployment of conventional network was not feasible. With the passage of time, phenomenal improvements have been made in maximizing the computing power of devices and performance of communication networks. This technological advancement poses a greater challenge to the researchers to provide services on Ad hoc networks that are available on infrastructure based networks. With increase in the use of Mobile devices such as cellular phones, PDAs, Laptops etc. in our daily lives, the requirement of providing same levels of services on all kinds of devices must be addressed. Multimedia transmission is one such application that requires computational resources as well as network bandwidth to provide information rich contents to the receiving nodes in real-time.

Digital Multimedia transmission over Ad hoc network requires encoding of source media in such a format that is resilient to the intermittent jitters in transmission due to route changes or link failures. Moreover as the intermediate nodes in an Ad hoc network act as repeaters to forward multimedia packets towards the destination nodes, the probability of failure increases with the increase in the number of intermediate nodes.

Furthermore, the insertion and removal of nodes in an Ad hoc network at any given time is a known problem of probability.

Multimedia scheduling requires selection of devices based on their computational abilities as well as their possession of network bandwidths while reducing the number of intermediate nodes in a chain of communication.

This paper further extends the notion of a *wireless cooperative system*, which involves migration and execution of tasks on two or more heterogeneous computing devices. Recent advances in wireless communication architectures and *computation offloading* platforms have made it possible to build such a cooperative environment on the fly and to let resource-limited devices make use of nearby resources for processing and communication. This technique not only has a potential for performance improvement (especially if the processing capability of the neighbor server is much greater than that of the mobile client), it can also help conserve energy of mobile computing devices.

2. Problem Definition

As multimedia scheduling is a multi-objective and constrained problem with all its known difficulties, the objectives of our scheduler is to minimize the length of communication chain while ensuring delivery of contents to the desired target nodes in a bounded time frame as imposed by the multimedia traffic constraints.

The development of an appropriate model is one of the most difficult tasks in solving complex real world problems. The model for the multimedia task scheduling proposed in this paper works in the following steps:

1. Discovery of neighboring computation and network resources
2. Allocation of unique identifiers to the selected nodes
3. Partitioning of multimedia task into computational and communicational parts
4. Assignment and delivery of partitioned tasks to selected appropriate nodes
5. Keeping track of topological changes in the network using it as feedback to the Task Assignment algorithm
6. Meeting task constraints by varying the priorities and lengths of task allocation to selected nodes

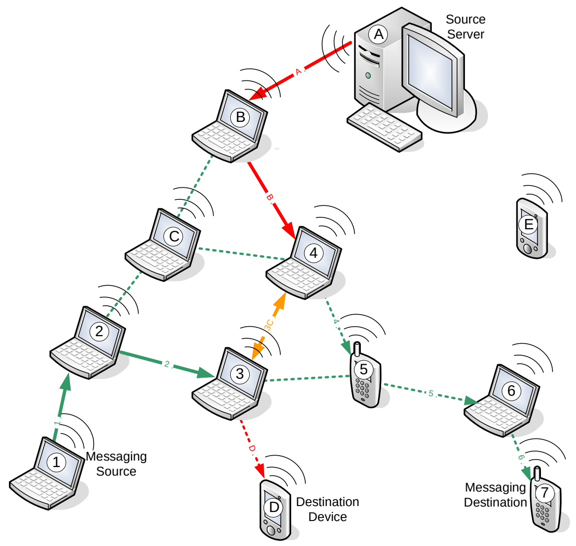

Figure1. Chain formation from Source devices to Destination devices (Path ABCD is utilized by Multimedia transmission, while path 1.2.3.4.5.6 is used by Messaging.

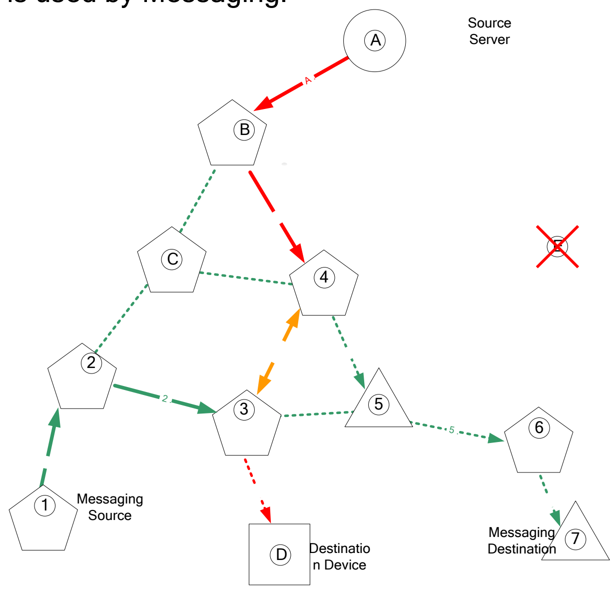

Figure2. Capability graph of neighboring devices

In Step 1, “Device Capability Query Packets” are sent from source device to its neighbor. This packet is forwarded by each node to its neighbor. Each node replies to this packet by sending its processing power, memory, storage capacity and network connectivity information (we call it “Device Capability Matrix”). This information is stored by each node and returned back towards the original source device. We use a modified version of OLSR with specialized “Hello Message” format the following packet format:

```
Message Format:

MessageType : HELLO_MESSAGE
CPU_Remaining: 0~100%
RAM_Remaining: 0~100%
HDD_Remaining: 0~100%
LNK_Remaining: in bps (Max. 1Gbps)
LNo: Link Interface Number (Max. 4)
```

To accommodate for remaining Processing Power, remaining memory, remaining storage and remaining network bandwidth, as well as to accommodate for future extensions, the packet format of a “Device Capability Matrix” is similar to the overall packet format of OLSR packet. The packet format is as follows:

```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved1            |     Htime     |  Willingness  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|CPU_Remaining|RAM_Remaining|HDD_Remaining|       Reserved      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      LNK_Remaining                        |LNo|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Link Code   |   Reserved2   |       Link Message Size       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Sender Interface Address                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Neighbor Interface Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                              . . .                            :
```

This is sent as the data-portion of the general packet format described in OLSR RFC3626 section 3.4, with the "Message Type" set to HELLO_MESSAGE, the TTL field set to 1 (one) and Vtime set accordingly to the value of NEIGHB_HOLD_TIME, specified in section 18.3

All nodes are identified using IP Addresses as routed by OLSR. This Device Capability information is propagated to all nodes using HELLO_MESSAGE. This mechanism is repeated periodically to confirm the availability of nodes prior to any communication.

In multimedia task scheduling, the tasks are either Computation Intensive or Bandwidth Intensive. This demands for a new type of scheduling algorithm that not only considers network connectivity between devices but also the device capabilities and resource availability. Our algorithm uses the “Device Capability Matrix” for routing and scheduling decisions. Computational Intensive tasks are forwarded towards more capable devices while Communication Intensive tasks are sent to devices having higher network bandwidth and sufficient resources.

Topological changes are discovered by OLSR protocol by sending special “Topology Control” Message of Message Type TC_MESSAGE.

Task Allocation to nodes requires selection of suitable candidate nodes depending on the type of task. For computation intensive tasks, computation offloading scheme is used. Previous work on computation offloading focused on monitoring of different resources, predicting the cost of local execution and that of a remote one and deciding between local and remote execution.

The computation offloading approach makes it feasible to have a set of heterogeneous computing devices working in a cooperative manner to tackle the resource limitation problem of mobile computing devices. However, this general cooperative approach has not been used in Ad hoc multimedia framework. The purpose of this paper is to fill this gap by introducing a new wireless cooperative remote execution and communication scheme that makes the most out of it.

3. The Scheduler Architecture

The task scheduling problem in its general form and its several restricted forms are NP-Complete. Only few scheduling problems have polynomial optimal algorithms. Moreover, these optimal algorithms work only under unrealistic assumptions. Scheduling problems are represented by task graphs.

An optimal scheduling algorithm was presented by Hu in [1] which works only when the task graph is a tree. The length of a path in a task graph is the sum of the execution times of all tasks along that path, but communication times are not included in the length of a path.

For multimedia scheduling, the communication delays must also be considered.

Various algorithms have been discussed in [2] to generate the minimal delay schedule. DAG scheduling in Heterogeneous Multiprocessor Systems is most suitable.

We propose a multi-stage scheduler architecture that works on all nodes of the Ad hoc network. This

Proceedings of the 5th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications, Madrid, Spain, February 15-17, 2006 (pp40-44)

scheduler works in conjunction with a modified version of the well known proactive protocol OLSR. OLSR uses Multi Point Relays to perform controlled flooding. Initially control packets are sent to discover the neighboring nodes. The Hello packet not only discovers the neighboring nodes but also query the device capability repeatedly after specific time intervals. The device capability matrix is updated using this information from time to time. This provides link and resource availability for the scheduler. Node capability matrix and other scheduling related information is available on all MPR nodes. The selection of MPR nodes not only depends on their ability to route data but also on their computation capabilities. The node that selects an MPR is called the selector node.

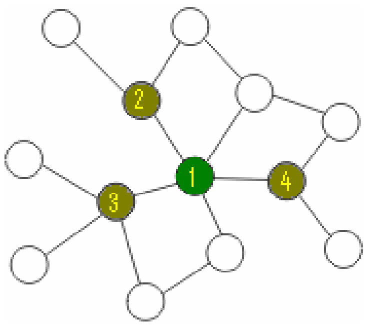

Figure3. Selection of MPR nodes (2, 3 and 4) by selector node 1

The device capability matrix contains remaining computation power and link delays from an MPR to its neighbors. This allows us to include or exclude nodes from the schedule if their computation capacity is below a threshold level. Similarly for a communication intensive task, any node having a large communication delay value can be safely excluded from the schedule. This helps in removal of nodes that are busy in computation or that have choked links in the subsequent schedules.

When multimedia data is required to be sent from any source node to a destination node, the device capability matrix is used to select the suitable nodes to carry out this task. The source node's scheduler becomes master scheduler where as the intermediate nodes use their scheduler as slave. Although the scheduling algorithm is same on all nodes but the overall schedule is maintained by the source node. The results of intermediate node schedules are sent back to the source node to create a summary of transaction. This allows us to perform any optimization in current overall schedule in light of the feedback gathered from the intermediate nodes.

Before the multimedia stream begins its transmission, the optimal route is decided in advance based on the type of nodes required for that multimedia transaction. In case of unavailability of a node, other alternate routes are used.

4. Allocation of Task

This section proposes a heuristic-based, greedy algorithm for the problem defined in Section 3. This algorithm tries to minimize the total delay in Data by assigning each task to the processing node that completes it with the minimal delay. By completion we mean that if a task is computation intensive, it has fully executed on that node while if the task is communication intensive, it has been transmitted by that node. When an assignment decision is to be made for a task *t*, this algorithm considers the time required of *t*'s completion on different processing nodes and the time required of *t*'s interaction with tasks that have been already assigned to other nodes. However, *t*'s interaction with tasks that have not yet been assigned is not considered because the cost of such interaction depends heavily on the assignment of those tasks and is impossible to predict. The algorithm can be described as follows:

```
unallocated_tasks=n;
While (unallocated_tasks > 0)
  Do {
        select_any_task(t_i);
        for(j=0,k=1;k<=m;k++)
          if (Exec(i,k) < Exec (i,j))
            j=k;
        Assign(t_i,p_j);
        unallocated_tasks--;
        if ( recv_Data_from_ti(t_u, p_k) )
         Exec(u,k) += Data(i,u) * Comm(j,k);
        if ( send_Data_to_ti(t_u, p_k) )
         Exec(u,k) += Data(i,u) * Comm(k,j);
      }
```

5. Execution Order of Tasks

Once the task assignment is decided, the exact communication delay and the exact execution time

of each task can be determined and, therefore, a more efficient execution order can be made based on the exact execution times and the exact communication delays.
Even though finding the optimal order for a given assignment is NP-hard, several heuristics have been proven to be effective in minimizing the total time of completion of tasks [4].

6. Future Work

We have used OLSR as the basis of our multimedia scheduling algorithm. As OLSR is a proactive protocol, the use of other reactive protocols such as AODV, TORA etc may provide faster response to link failures. Although OLSR provides good overall route if no problem occurs during a transaction.

We also noticed that OLSR behaves strangely when the nodes are aligned closely within coverage range of a node. This may be due to interference between node signals. This needs further investigation.

Interesting experimentation can be performed by using different types of protocols on different nodes (instead of OLSR on all nodes) of Ad hoc network with a translation mechanism to translate packets among them.

Energy conservation mechanisms may also be introduced to affect scheduling decisions e.g. Remote execution is preferred when a higher performance device is connected to wall socket instead of performing it locally on a battery operated node.

QoS routing algorithms such as those proposed by Lohier et al in [3] can be utilized to improve the schedule.

Different Scheduling algorithms can be used on source and intermediate nodes instead of copy of same algorithm on all nodes.

7. References

[1] T C. Hu, "Parallel Sequencing and Assembly Line Problems", Operations Research, No 9, pp. 841- 848, 1961.

[2] YU-KWONG KWOK and ISHFAQ AHMAD. "Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors"

[3] Lohier et al, "QoS Routing in ad hoc networks" , 2002

[4]. D. T. Peng and K.G. Shin, "Optimal Scheduling of Cooperative Tasks in a Distributed System Using an Enumerative Method", March 1993 (Vol. 19, No. 3) pp. 253-267

[5] C. Toh, "Ad Hoc Mobile Wireless Networks: Protocols and Systems". PTR Prentice Hall, 2002.

[6] J. Flinn, D. Narayanan and M. Satyanarayanan, "Self-Tuned Remote Execution for Pervasive Computing," In Procceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), May 2001.

[7] E. Coffman and R. Graham, "Optimal Scheduling of Two-Processor Systems," Acta Informatica, No 1, pp. 200-213, 1972.

[8] T. Yang and A. Gerasoulis, "List Scheduling with and without Communication Delays," Parallel Computing, 19, No 12, pp. 1321-1344, 1993.

[9] X. Gu, A. Messer, I. Greenberg, D. Milojicic and K. Nahrstedt, "Adaptive Offloading for Pervasive Computing," In Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom), March 2003.

[10] J. Ullman, "NP-Complete Scheduling Problems," Journal of Computer and System Sciences, No 10, pp. 384-393, 1975.

[11] Clausen & Jacquet, OLSR; rfc3626, October 2003

[12] Raffo et al, "An Advanced Signature System for OLSR"