Least Bandwidth Allocation for Periodic Streams Wireless Network System

Elumalai SV
M.E Student, Dept of CSE,
Sri Krishna Engineering College, Chennai,
Tamilnadu, India.
E-mail: elumalaielumalai178@gmail.com,

Ramalingam N M K
Asst.Professor,Dept of *CSE*,
Sri Krishna Engineering College, Chennai,
Tamilnadu, India.

Abstract - Allocation based as opposed to contention-based channel access in WLANs provides predictable and deterministic transmission and is therefore able to provide timeliness guarantees for wireless and embedded real-time applications. Also, reservation-based channel access is energy-efficient since a wireless adaptor is powered on only during its exclusive channel access times. While scheduling for Quality of Service at the central authority has received extensive attention, the problem of determining the actual resource requirements of an individual node in a wireless real-time system has been largely ignored. Minimum channel bandwidth reservation that meets the real-time constraints of all periodic streams of a given node. Keeping the bandwidth reservation of a node to a minimum leads to reduced energy and resource requirements and leaves more bandwidth for future reservations by other nodes. To the minimum bandwidth reservation problem, we transform it to a generic uniprocessor task schedulability problem, which is then addressed using a generic algorithm. This algorithm works for a subclass of priority-driven packet scheduling policies, including three common ones: fixed-priority, EDF, and FIFO. Moreover, we then specialize the generic algorithm to these three policies according to their specific characteristics. Their computation complexities and bandwidth reservation efficiencies are evaluated and guidelines for choosing scheduling policies and stream parameters are presented.

Keywords: *Bandwidth reservation, schedulability test, earliest deadline first, fixed-priority, first-in-first-out, wireless.*

I. INTRODUCTION

Minimum bandwidth reservation usingthe continuous increase in streaming applications such as video/audio communications, industrial automation, networked and embedded control systems, and wireless sensor and actuator networks. This has called for research efforts to enhance the support of timeliness and Quality of Service (QoS) in wirelessly networked embedded environments. The goal of this work is to develop a strategy for the computation of the required channel access reservations for a given packet scheduling policy, such that 1) the real-time constraints of each node's traffic are satisfied and 2) resource reservations are minimized. The minimum bandwidth reservation problem at a given node, we treat the complement of the periodic bandwidth reservation as a special periodic stream (the periodic sleep stream), i.e., the bandwidth reservation per channel access period is equal to the complement of the execution time (sleep time) of the sleep stream with period equal to the channel access period. We add the sleep stream to the original stream set to form an extended stream set. Accordingly, the scheduling policy for the extended stream set is extended from the original scheduling policy for the original stream set such that 1) the sleep stream always has the highest priority and is no interruptible and 2) the priority relationship among the original stream set is unchanged. As a consequence, we transform the minimum bandwidth reservation problem to the maximum sleep time problem. In other words, there exists a schedule for the original stream set with a given scheduling policy if and only if there exists a schedule for the extended stream set using the extended scheduling policy. Therefore, minimizing the bandwidth reservation is equivalent to maximizing the sleep time of the sleep stream. The remainder of this paper is structured as follows: We discuss related work in Section 2. In Section 3, we present the generic traffic model, the network access model, and the problem this paper is investigating. The model transformation is presented in Section 4. Then, the generic algorithm of the minimum bandwidth reservation problem is in Section 5, and its refinements for fixed-priority policies Section 5.2, EDF Section 5.3, and FIFO Section 5.4 are given subsequently. In Section 6, we present simulation results for various stream configurations under various packet scheduling polices. Finally, we conclude this paper in Section 7.

II. RELATED WORK

Minimum bandwidth reservation using Scheduling and schedulability analysis have been extensively studied in previous work, particularly for processing resources. In networking environments, reservation-basedmechanisms are becoming highly prominent in supporting latency-critical and energy-aware traffic. In this section, we discuss existing protocol standards and techniques related to resource and channel access reservations. shared packet transmission schedule. Packets are transmit during their allocated slots, thus avoiding contention.

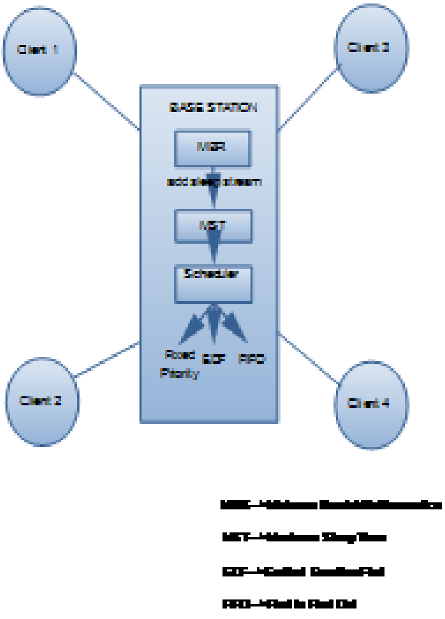
Traffic from the same node is interspersed into discrete slots. 1) The channel access period is smaller than every stream period, 2) datagram deadlines must be equal to or less than their respective periods, and 3) datagram must be transmitted consecutively and without interruption.

Also, the work implicitly assumes that the underlying scheduling policy is FIFO. Although it has a linear complexity, it overreserves bandwidth in general. This paper removes these restrictions, takes into account the impact of different scheduling policies on the computation of the required bandwidth, and presents algorithms for several scheduling policies to efficiently compute the minimum bandwidth reservation, although at the cost of higher complexity compared. However, even with increased complexity, the presented algorithms are practical considering that the computational capacity of wireless end devices continuously increases and that a wireless end device usually has only a limited number of concurrent real-time streams. In this work, we transform the minimum bandwidth reservation problem to the maximum sleep time problem, which is then solved by computing the schedulable execution time of the sleep stream for a subclass of scheduling policies, including fixed-priority, EDF, and FIFO.

A similar approach has been taken in to solve the minimum EDF-feasible deadline problem of a given task, given its period and execution time. There exist a lot of efforts on exact schedulability tests for various scheduling policies, e.g., fixed-priority, and FIFO . Our generic algorithm for the bandwidth reservation problem is based on the time-demand analysis techniques provided by these earlier research results, but applied to a new problem. Our work is also closely related to previous work on resource partition/composition models, which usually focus on processor resources in real-time systems. Such models include the static resource partition model, the bounded delay resource partition model, the periodic resource model, and the explicit deadline periodic resource model. These prior efforts differ from each other mainly in the chosen scheduling model. The resource partition/reservation model used in this paper corresponds to the single time slot periodic partition (STSPP) model (a special case of the static resource partition model)introduced.

The schedulability test scheme for the STSPP model in can also be used to solve the problem, by iteratively executing the schedulability test algorithm proposed in(similar to a binary search). Therefore, this approach would be very costly. It is particularly inefficient for the fixed prioritypolicy since the change of the resource supply and the change of job response times are nonlinear and irregular. Our algorithm augments the traditional time demand analysis to compute the finished/unfinishedportion before a job's deadline, which avoids computing fixed-point equations iteratively. As a result, our solution to the reverse problem (the minimum resource requirement problem) has the same complexity as the original problem (the exact schedulability test problem). Besides resource partition/composition models, hierarchical schedulers can also provide temporal isolation among applications on a uniprocessor. This property prevents a misbehaving task from interfering with other tasks in another application, i.e., only the tasks within the same application as the misbehaving one could be affected. In hierarchical scheduling, each application is composed of a set of correlating entities (e.g., tasks or streams), where applications are scheduled by a global scheduler and each application schedules its tasks using its local scheduler. The approach proposed in this paper can be applied to hierarchical schedulers, i.e., it can be used to determine the minimum resource requirements of an application, given this application's local scheduling policy (fixed-priority, EDF, or FIFO).

Figure1. System Architecture



III. BANDWIDTH RESERVATION MODEL

This section presents our network access model, traffic model, and the problem statement.

3.1 Network Access Model

The concept of reservation-based channel access model since it forms the basis for the problem we intend to solve. Such a mechanismresource reservations to ensure contention-free accesses.This is achieved through a central authority at a BSthat regulates the channel accesses of individual nodes.Here, the BS takes control of the channel and starts pollingeach of the nodes in a predetermined order.Upon reception of a polling frame, a node gainsaccess to the channel. The HCCA mode defined in the IEEE802.11e standard is an example of a protocol whichadopts the reservation-based channel access approach toenhance the QoS support for real-time applications inwireless environments. A reservation-based channel access mechanism,each node is provided a Service Period (SP), during whichthe node has exclusive access to the wireless medium.

Polling frames issued by the BS specify the start time and maximum duration of the SP allotted to a node. At the end of an SP for a node, the BS begins polling the next node in its schedule. The period of recurrence of the SPs is referred to as the Service Interval (SI), which is usually specified by the BS in advance and equal to a multiple of the beacon interval of the BS shared by all client nodes. There are three advantages of using STSPP in these network areas: 1) it saves energy since nodes only need to wake up to communicate within their respective reserved time intervals; 2) it leads to better latency predictability and possibly higher throughput since wireless contention is avoided a priori; 3) it greatly decreases the runtime complexity of resource partition scheduling due to its simple partition structure.

3.2 Traffic Model

We consider a set of wireless nodes with applications on each node generating one or more periodic real-time streams. Nodes connect wirelessly to a common BS to access an external network. We denote the set of periodic streams generated by a node . Each stream Si periodically generates a certain number (worst case or average case) of bytes (called a datagram) for a given period pi for transmission. The datagram generated at the beginning of the jth period of Si for transmission is denoted as Ji;j. Wireless channel conditions are time-varying and error-prone. The worst-case estimation of the transmission time of a datagram of Si is needed, which has been the focus of many prior efforts. Each datagram of Si has a relative transmission completion deadline Di. The release time and deadline of are denoted as ri;j and Di, respectively. Our framework requires no specific relationship between stream periods and datagram deadlines, i.e., Di can be less than, equal to, or greater than pi. Due to the similarity between the concept of tasks in the literature and the concept of streams in this paper, we will use stream and task interchangeably in this paper. Similarly, the terms datagram and job are also used interchangeably. Datagram's are often fragmented at the network and/or link layer, depending on the datagram size, network parameters and the scheduling policy.

Therefore, a datagram can be also A wireless device's bandwidth profile (SP; SI). Shaded intervals (SPs) are the exclusive access periods for a node; the scheduler may need to transmit these newly arrived urgent packets before packets. These interleaved transmissions happen frequently since applications treat the network as a dedicated resource and issue packets regardless of the network reservation. In this paper, we investigate the impact of traffic scheduling policies on bandwidth reservation from the perspective of a single node. The policies under consideration are a subclass of priority-driven polices, and we assume there is a shared queue for all released packets. The priority driven scheduling policies under consideration include fixed-priority policies, e.g., rate/deadline monotonic scheduling (RM and DM), and dynamic-priority policies, e.g., EDF, and FIFO. All nodes in a WLAN can use their own scheduling policies.

IV. MINIMUM BANDWIDTH RESERVATION

A priority-driven scheduling policy can be considered as a time-varying function for any two jobs. This while adding flexibility, would significantly add to the complexity of the network scheduler. this work assumes that the network scheduler is job-level static, i.e., all packets from the same datagram have the same priority, which is assigned at the release time of the datagram. We would like to point out that many priority driven scheduling policies (especially widely used ones such as RM, DM, EDF, FIFO, LILO, and round-robin) satisfy these constraints. Policies such as least slack time (LST) do not fall into this category. In the following sections, we restrict our focus to the class of priority-driven policies that satisfy these constraints. Moreover, we require that for a scheduling policy the synchronous (in-phase) busy interval of any task set is the worst-case scenario. A synchronous busy interval of the stream set starts with all streams generating their datagram at the same time and ends with the transmission of the last one of these datagram.

4.1 Generic Framework

We develop a generic algorithmic framework (Algorithm 1) to solve the MET problem, based on an augmented time-demand analysis. In the following description, we distinguish between the finished portion and the unfinished portion of execution before a given deadline. This concept allows us to conservatively reduce the sleep time of the sleep stream to approach its minimum value, assuming that the reduced sleep time is solely utilized by the job missing its deadline. To compute the finished/unfinished portions of a job, we define a generic time-demand function at every job release event point. The available time at a job release event point for lower priority jobs is equal to the dedicated time supply minus the generic time demand.

4.2 Scheduler

This algorithm works for subclass of periodic driven packet scheduling Policies, including three common: 1.FIFO, 2. EDF, 3. Fixed Priority.

FIFO : In the First In First Out scheduling algorithm , only jobs released no later than job Ji,j have higher priority than Ji,j.

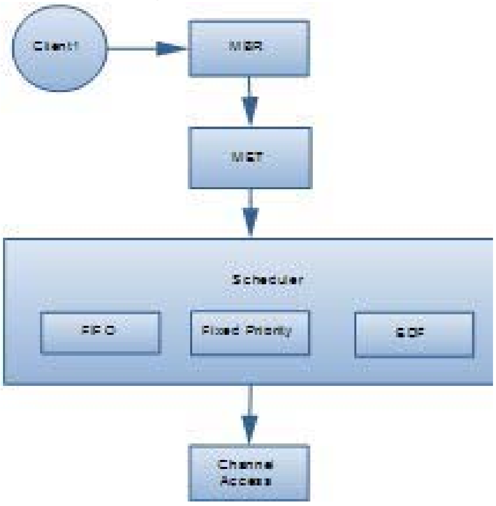
EDF: In the Earliest Deadline First scheduling algorithm, jobs with priorities higher than the priority of a job Ji,j are those whose deadlines are no later than di,j , and therefore they will be executed before di,j .

Fixed Priority: The priority rule of a fixed-priority scheduling policy A is static at stream level.

Algorithm 1. Minimum bandwidth reservation of stream set $\mathcal{S}'$ using policy A'

```
function MinBW():
/* $e_0$ is initialized to the maximum value since the goal is to
   minimize bandwidth overreservation for $\mathcal{S}'$ */
$e_0 = D_0 = (1 - \sum_{1 \le i \le n} \frac{e_i}{p_i}) * SI$
/* scan every job $\mathcal{J}_{i,j} \in \Pi$ (except the sleep job) in increasing
   order of release times or deadlines, where $0 < i \le n$ and $\Pi$ is
   the synchronous busy interval starting at time 0 */
for each $\mathcal{J}_{i,j}$ within $\Pi$ do
    loop
        compute the finished portion $e^+_{i,j}$ and
        unfinished portion $e^-_{i,j}$ using (5) and (6), and
        the completion time $f_{i,j}$ using (7)
        if $e^-_{i,j} > 0$ then
            decrement the execution time $e_0$ of $\mathcal{S}_0$
            using (2)
        else
            Break
        end if
        /* abort when it is impossible to meet the job's
           deadline even if the node were allocated the entire
           bandwidth (i.e., $SP = SI$) */
        if $e_0 < 0$ then
            Abort
        end if
    end loop
    /* determine if the synchronous busy interval $\Pi$ is
       terminated using (8) or whether it is impossible to meet
       the job's deadline even if the node were allocated the
       entire bandwidth (i.e., $SP = SI$) */
    if $v(f_{i,j}) <= f_{i,j}$ then
        Break
    end if
    move on to check next job (other than the sleep job)
end for
return $\max\{SI - e_0 - \theta, 0\}$
```

Figure 2. Scheduler



4.3 EDF

In EDF, jobs with priorities higher than the priority of a job J i;j are those whose deadlines are no later than di;j, and therefore they will be executed before di;j. Thus, thetime-demand function (3)finished/unfinished portion function (5), and completion time function (7) can be which are the same . The jobs released within di;j with higher priorities than J i;j are those whose deadlines are also. Since these jobs will complete within ðri;j; di;j using the EDF policy, theirdemand times are equal to their actual execution times. In other words, if the release time of the next job to be analyzed is no less than terminates.

4.4 FIFO

In FIFO, only jobs released no later than job J i;j have higher priority than J i;j. The time-demand function (3), finished/unfinished portion function (5), and completion time function (7) can be customized as: The time-demand function is a constant function within Reflecting on the shape of, we see that the function is a horizontal line. Furthermore, if wi;j is greater than di;j, the unfinished portion ei;j can be compensated by the reduction of the sleep time of S0. In particular, the total reduced amount of sleep times of all jobs of S0 released before di;j is equal.

V. PERFORMANCE EVALUATION

The benefits of reservation-based channel accesses are twofold: 1) they provide contention-free access within allocated/reserved channel access intervals to meet timing constraints predictably and 2) they allow a wireless radio to be powered down when the channel is not needed. Careless resource allocations may lead to poor support for real-time traffic or over provisioning of scarce network resources. This paper solves the minimum bandwidth reservation problem to allow all streams to meet their timing constraints. To obtain a solution to the minimum bandwidth reservation problem, we transform it to a generic uniprocessor task schedulability problem, which is then addressed using a generic algorithm based on time-demand analysis. The generic minimum bandwidth reservation algorithm works for a subclass of priority-driven packet scheduling policies, including three common ones: fixed-priority, EDF, and FIFO.

VI. CONCLUSION

The generic minimum bandwidth reservation algorithm works for a subclass of priority-driven packet scheduling policies, including three common ones: fixed-priority , EDF, and FIFO. Refinements of the generic solution to these three types of policies are presented and discussed as well. The simulation results show that the generic algorithm is correct and practical in terms of computation complexity. The proposed bandwidth reservation scheme leads to minimal amounts of bandwidth waste if appropriate scheduling policies and stream parameters are selected for a given stream set. However, it also leads to potentially large energy savings, while being simple to implement and deploy. In our future work, we will address how the base station chooses the optimal SI value to minimize the bandwidth/energy consumption of the entire wireless local be composed efficiently to form a super frame.

REFERENCES

1. Mok .A.K, X.A. Feng, and D. Chen, “Resource Partition for Realtime Systems,” Proc. Seventh IEEE Real-Time Technology and Applications Symp., pp. 75-84, 2001.
2. Zhang .F and A. Burns, “Analysis of Hierarchical EDF Pre-Emptive Scheduling,” Proc. 28th IEEE Int’l Real-Time Systems Symp., pp. 423-434, 2007.
3. Lehoczky.J, “Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines,” Proc. 11th Real-Time Systems Symp., pp. 201-209, Dec. 1990.
4. Shin .I and I. Lee, “Periodic Resource Model for Compositional Real-Time Guarantees,” Proc. 24th IEEE Int’l Real-Time Systems Symp., pp. 2-13, 2003.
5. Shin.I and I. Lee, “Compositional Real-Time Scheduling Framework with Periodic Model,” ACM Trans. Embedded Computing Systems, vol. 7, no. 3 pp. 1-39, 2008.
6. George.L and P. Minet, “A FIFO Worst Case Analysis for a Hard Real-Time Distributed Problem with Consistency Constraints,” Proc. 17th Int’l Conf. Distributed Computing Systems, pp. 441-448, 1997.
7. Feng. X.A. and A.K. Mok, “A Model of Hierarchical Real-Time Virtual Resources,” Proc. 23rd IEEE Real-Time Systems Symp.,pp. 26-35, 2002.
8. Fishe .N and F. Dewan, “Approximate Bandwidth Allocation for Compositional Real-Time Systems,” Proc. 21st Euromicro Conf. Real-Time Systems, pp. 87-96, 2009.
9. R.I. Davis, A. Zabos, and A.Burns, “Efficient Exact Schedulability Tests for Fixed Priority Real-Time Systems,” IEEE Trans. Computers, vol. 57, no. 9, pp. 1261-1276, Sept. 2008.