An Admission Control Algorithm for Scheduling Mixed Traffic in Ubiquitous Environment

Yeonseung Ryu, Seokwon Hong, Hyuksoo Jang, Sehyeong Cho, and Chungki Lee

Department of Computer Software, Myongji University
Nam-dong, Yongin, Gyeonggi-do, 449-728, Korea
ysryu@mju.ac.kr

Abstract. In ubiquitous environment the traffic is expected more diverse than ever - ranging from traditional file transfer to continuous media applications such as audio and video conferencing. In this paper, we present a traffic scheduling scheme and an admission control algorithm for servicing a mix of *non-real-time* and *real-time* traffic in ubiquitous environment. Proposed scheduling algorithm assigns a deadline to the non-real-time packet by calculating the *slack time* on-line and services the non-real-time packet along with the real-time packet using *Earliest Deadline First* algorithm. The time complexity for calculation of the slack time and deadline assignment to the non-real-time packet is $O(1)$ and thus the scheduling of non-real-time traffic can be performed *at a low cost*. Since proposed scheme services the non-real-time traffic while the link bandwidth is not used by the real-time traffic, it can guarantee the schedulability of real-time flows. Moreover, by limiting the number of admitted real-time flows in admission control algorithm, proposed method can provide the non-real-time flows with fast response time.

Key Words : Ubiquitous network, Mixed-traffic scheduling, Admission control, Slack stealing

1 Introduction

The recent trend of computing is progressing towards the vision of ubiquitous computing, in which devices are seamlessly integrated into the life of everyday users, and services are rapidly available to users anywhere they go at any time. In ubiquitous environment the data traffic is expected more diverse than ever - ranging from traditional file transfer to continuous media applications such as audio and video conferencing. The continuous media applications require the ubiquitous network to provide quality of service (QoS) guarantee such as packet delay, minimum bandwidth, delay jitter and loss.

The function of traffic scheduling algorithm in the traditional network switches is to select the packet to be transmitted in the next cycle from the available packets belonging to the flows sharing the output link. The traffic scheduling algorithm is of great importance because various quality-of-services are possible depending on how to schedule the packet transmission. For the past decade, a number of traffic scheduling schemes have been extensively studied to provide the real-time traffic with deterministic upper bounds on the end-to-end delay when the burstness of the flow traffic is bounded (for example, shaped by a leaky bucket).

The problem of scheduling hybrid traffic, consisting of real-time and non-real-time traffic, has been considered in some literatures [14, 8, 12]. In [14], *Rate-Controlled Static Priority* (RCSP) was presented. A RCSP scheduler consists of multiple prioritized *First Come First Served* (FCFS) queues and services packets using a non-preemptive static-priority discipline: which non-preemptively chooses packets in FCFS order from the highest-priority non-empty queue. Non-real-time packets have the lowest static priority and are serviced only when there are no real-time packets. While RCSP can provide service guarantees to real-time flows at a *low cost*, it cannot provide high-performance to non-real-time traffic. In [8], *Rotating Combined Queueing* (RCQ) was proposed. RCQ uses frame-based scheduling, where each connection is allocated some number of packet slots in a fixed frame time. RCQ can provide good performance to non-real-time traffic by allowing bursty traffic to utilize unused bandwidth. However, it cannot efficiently support communications with low delay bounds requirements since the worst-case packet delay is coupled to the frame size. In [12], aperiodic server mechanism was studied.

Real-time packet schedulers based on *Earliest Deadline First* (EDF) policy have been studied to support end-to-end bounded delay for real-time traffic [10, 7, 4, 5, 1]. In [10], a general necessary and sufficient schedulability (deadline guarantee) condition of EDF was obtained for flows with traffic regulation assumption. The EDF algorithm is regarded the best choice since it has been proven to be optimal and achieve full resource utilization [11].

In this paper, we present a traffic scheduling scheme and an admission control algorithm to service a mix of *non-real-time* and *real-time* traffic under EDF scheduling algorithm. Goals of this work are to:

- guarantee the real-time constraints (end-to-end bounded delay) of the real-time flows.
- service the non-real-time flows with fast response time.

Proposed traffic scheduler services real-time flows and non-real-time flows using EDF algorithm. After real-time flows are determined to be schedulable under EDF, we can service non-real-time flows using unused link capacity. The available time to use link at a given time is called *slack time*. Proposed traffic scheduler assigns the deadline to non-real-time packet by calculating the slack time on-line and services the non-real-time packet along with the real-time packet using EDF algorithm. Since it services the non-real-time traffic when the link bandwidth is not used, it can guarantee the schedulability of the real-time flows. In proposed scheme, the time complexity for calculation of slack time and deadline assignment to the non-real-time packet is $O(1)$ and thus traffic scheduling can be performed at a low cost. Through the experiments, we investigate the length of deadlines of the non-real-time packets and examine the responsiveness of non-real-time traffic.

The rest of this paper is organized as follows. In Section 2, we review previous works that are relevant for this paper. In Section 3, we present a new scheduling method and admission control algorithm to service a mix of real-time and non-real-time traffic under EDF. Section 4 presents the experimental results to show the performance of proposed scheme. The conclusions of this paper are given in Section 5.

2 Related Work

Recently a number of analytical models for Earliest Deadline First (EDF) scheduling were proposed to provide delay bounds for real-time communications [10, 7, 4, 5, 1, 13, 15]. The real-time flows have real-time constraints, such as end-to-end bounded delay. We assume that flow setup protocol such as Resource Reservation Protocol (RSVP) is used to provide guaranteed service to the real-time flows. After establishing a flow i, a packet j of flow i arriving at each node has deadline, d_i. The real-time packets are serviced by EDF scheduling policy. The EDF scheduler selects the real-time packet with the earliest deadline for transmission. In general, an EDF scheduler assigns each arriving packet a deadline, computed as the sum of the arrival time at the scheduler and the delay bound. Let d_i be a local delay bound of flow i and $a_{i,j}$ be an arrival time of a packet j of flow i at the scheduler, respectively. Then the deadline of packet j of flow i is assigned as

$$D_{i,j} = a_{i,j} + d_i \tag{1}$$

The EDF scheduler needs to know the condition that must hold at the scheduler such that delay bound violations of real-time flows do not occur. These conditions are referred to as *schedulability conditions*. In [6, 10], the authors presented necessary and sufficient schedulability conditions for a single node under which a set of delay bounds can be met.

Since a network traffic could be distorted and local schedulability conditions will be violated, two mechanisms have been proposed to enforce that traffic entering a packet scheduler conforms to the given traffic constraint function. One mechanism is a *traffic policer* which rejects traffic if it does not comply to traffic constraint function. The other is a *rate controller*, which temporarily buffers packets to ensure that traffic entering the scheduler queue conforms to traffic constraint function. The idea of per-node traffic shaping called RC-EDF (Rate-Controlled EDF) was proposed [7, 2]. In [7], the authors showed that with identical shapers at each node along a flows' path, EDF can provide end-to-end delay bounds.

Traffic constraint functions can be derived from deterministic traffic models that characterize the worst-case traffic by a small set of parameters. Let $A_i[s, t]$ be the amount of traffic (measured in bits/second) from flow i arriving over interval $[s, t]$. We assume that $A_i[t] = A_i[0, t]$. Let $A_i^*(t)$ be a right continuous subadditive function. $A_i^*(t)$ is said to be an envelope of flow $A_i[t]$, if for all times $s > 0$ and for all $t \geq 0$ we have

$$A_i[s,t] \leq A_i^*(t-s) \tag{2}$$

where $A_i^*(t) = 0$ for all $t < 0$. For example, the traffic constraint function for the (σ, ρ)-model [3] is given by $A_i^*(t) = \sigma_i + \rho_i t$, where σ_i is a burst parameter and ρ_i is a rate parameter of flow i.

Let $R = \{1, 2, ..., N\}$ be a set of real-time flows, where flow $i \in R$ is characterized by the traffic constraint function. Then the schedulability condition for EDF scheduler is given as follows [10, 5]:

Theorem 1 *The set R is EDF-schedulable if and only if for all $t \geq 0$*

$$\sum_{i \in R} A_i^*(t - d_i) \leq ct \tag{3}$$

where c is the capacity (maximum rate) of the link (bits/second).

Informally the theorem states that the real-time flows are EDF-schedulable iif the sum of the time for transmitting the real-time traffic that arrived with deadline before or at time t does not exceed the available time in the interval [0, t].

3 Scheduling mixed traffic under EDF

3.1 Scheduling of Real-time Flows

The real-time flows have real-time constraints, such as end-to-end bounded delay. The flow setup protocol such as RSVP is used to provide guaranteed service to the real-time flows. During the flow setup procedure, EDF scheduler checks the schedulability whenever a new real-time flow joins. When the real-time flows (including a new flow) are characterized by the traffic constraint function, $A_i^*(t)$, the schedulability condition for EDF scheduler at each node along the flow's path is given in Theorem 1 (see Section 2). After establishing a flow i, a packet j of flow i arriving at each node has deadline as $D_{i,j} = a_{i,j} + d_i$, where $a_{i,j}$ is arrival time of packet j of flow i and d_i is a local delay bound of flow i, respectively.

We assume that each node has the rate controller, such as RC-EDF (Rate-Controlled EDF) [7] which temporarily buffers the real-time packets to ensure that traffic entering the EDF scheduler queue conforms to traffic constraint function. Let $A_i[s,\ t]$ be the amount of traffic (measured in bits/second) from flow i arriving over interval $[s,\ t]$. We assume that $A_i[t] = A_i[0, t]$. $A_i^*(t)$ is said to be a traffic constrain function of flow $A_i[t]$, if for all times $s > 0$ and for all $t \geq 0$, we have $A_i[s,t] \leq A_i^*(t-s)$, where $A_i^*(t) = 0$ for all $t < 0$. In this work, we assume that the real-time traffic is characterized by leaky bucket:

$$A_i^*(t) = \sigma_i + \rho_i t, \tag{4}$$

where σ_i is a burst parameter and ρ_i is a rate parameter of flow i.

3.2 Analysis of slack time

The non-real-time flows do not have real-time constraints and do not require end-to-end flow setup protocol like RSVP. But they require fast response time. We assume that traffic arrival of the non-real-time flows is not shaped or bounded by any traffic constraint function.

After real-time flows are determined to be schedulable under EDF, we can service non-real-time packets using unused link capacity. The available time to use link at a given time is called *slack time*. If we can calculate the slack time on-line and assign the deadline to non-real-time packets using the slack time, then non-real-time traffic can be serviced without hurting the real-time requirements of real-time traffic. We define *utilization factor* of non-real-time flows, U_R, as

$$1 - \frac{1}{c}\sum_{i\in R}\rho_i. \tag{5}$$

For example, $U_R = 0.1$ means that the available link bandwidth for non-real-time flows is at most 10% of the total link bandwidth since the link utilization due to current admitted real-time flows is 90%. We assume $0 < U_R < 1$. The following theorem gives the amount of slack time at a given time.

Theorem 2 *For any $t \geq 0$, the time interval $[t, t+t_1]$ where*

$$t_1 = \frac{x + \xi_R}{U_R} \tag{6}$$

contains at least x amount of slack time, where

$$\xi_R = \frac{1}{c}\sum_{i\in R}(\sigma_i - \rho_i d_i) \tag{7}$$

Proof: We prove the theorem by contradiction. Let us assume that the amount of slack time in $[t, t+t_1]$ is smaller than x. Then there must be a deadline miss before $t+t_1$ if we add a link use time x into the interval $[t, t+t_1]$. Furthermore, from a certain time $t^{'} < t+t_1$, only real-time packets ready at $t^{'}$ or later and having deadline less than or equal to $t+t_1$ are transmitted. Let C be the total transmission time demanded by these real-time packets. Since there is a violation at $t+t_1$, it must be $C > t+t_1-t^{'}$. Moreover,

$$C \;\leq\; \frac{1}{c}\sum_{i\in R}A_i^*(t+t_1-t^{'}-d_i)+x = \frac{1}{c}\sum_{i\in R}(\sigma_i+\rho_i(t+t_1-t^{'}-d_i))+x \tag{8}$$

$$\leq \frac{1}{c}\sum_{i\in R}\rho_i(t+t_1-t^{'}) + \frac{1}{c}\sum_{i\in R}(\sigma_i-\rho_i d_i)+x \tag{9}$$

Thus,

$$(1-\frac{1}{c}\sum_{i\in R}\rho_i)(t+t_1-t^{'}) < \frac{1}{c}\sum_{i\in R}(\sigma_i-\rho_i d_i)+x \tag{10}$$

$$t + t_1 - t^{'} < \frac{x}{U_R} + \frac{\frac{1}{c}\sum_{i\in R}(\sigma_i - \rho_i d_i)}{U_R} \tag{11}$$

Since $t_1 = \frac{x+\xi_R}{U_R}$, it is followed by $t < t^{'}$, which leads to a contradiction. Therefore, there is at least x amount of slack time in the interval $[t, t + t_1]$ where $t_1 = \frac{x+\xi_R}{U_R}$. □

3.3 Deadline Assignment for Non-real-time Packets

Consider that a non-real-time packet i requesting T_i transmission time arrives at time s_i. In order to service packet i, at least T_i amount of slack time is needed. From the theorem 2, there is slack time of T_i in the interval $[s_i, s_i + t_1]$ where $t_1 = \frac{T_i+\xi_R}{U_R}$. Therefore, we can assign

$$max\{s_i, f_{prev}\} + \frac{T_i + \xi_R}{U_R} \tag{12}$$

as its deadline, where f_{prev} is the time at which the previous non-real-time packet is finished. If a non-real-time packet has arrived before the completion of a previous non-real-time packet, we can assign its deadline using f_{prev} just after the completion of the previous packet. Then the EDF scheduler can schedule the non-real-time packet along with real-time packets according to EDF policy while guaranting that no deadline of real-time traffic will be missed.

The time complexity for deadline assignment is $O(1)$, because we can keep track of ξ_R and U_R which are calculated only using the real-time traffic which are already guaranteed.

3.4 Admission Control Algorithm

In this section, we propose a new admission control algorithm to guarantee the delay requirements of the real-time flows while providing the non-real-time flows with fast response time. The admission control algorithm is executed whenever a new real-time flow joins the packet scheduler. It checks serviceableness of the non-real-time flows as well as the schedulability of the real-time flows.

Since the non-real-time packets are serviced by EDF scheduler according to their assigned deadlines, the response time of the non-real-time packets are dependent on their deadlines. Deadlines of the non-real-time packets are determined by traffic parameters of the admitted real-time flows and the link utilization due to the real-time flows (See Equ. 12). If the number of admitted real-time flows increases and the link utilization due to the real-time flows becomes higher, the non-real-time packets are assigned longer deadlines and thus are serviced with high response time. Hence, in order to provide non-real-time packets with bounded response time we should limit the number of admitted real-time flows.

Let T_{RT} be the average transmission time for non-real-time packets and B_{RT} be the bound of response time for non-real-time packets, respectively. In Figure 1,

```
ADMISSION_CONTROL_ALGORITHM(
input : set R of real-time flows with (σ_i, ρ_i, d_i), and system parameters (B_RT and
T_RT)
output : schedulable or not)

begin
    if Equ. 3 is not satisfied then
        return(not schedulable);
    endif;
    calculate U_R and ξ_R;          /* Equ. 5 and 7 */
    if (T_RT+ξ_R)/U_R > B_RT then
        return(not schedulable);
    endif;
    return(schedulable);
end.
```

Fig. 1. Admission Control Algorithm

we give an admission control algorithm which is executed whenever a new real-time flow joins the packet scheduler. We can easily see that this algorithm can be executed in time $O(1)$, because we can keep track of $\sum_{i \in R} A_i^*(t - d_i)$, ξ_R and U_R which are calculated only using the real-time traffic which are already guaranteed.

4 Experiments

In this section, we have performed some experiments in order to investigate the responsiveness of non-real-time flows given a set of admitted real-time flows. When a non-real-time packet i requesting T_i transmission time arrives, its response time is $\frac{T_i + \xi_R}{U_R}$ if there is no previous non-real-time packet or previous non-real-time packet was already transmitted. The response time of non-real-time traffic is mainly determined by traffic parameters of real-time flows and link utilization due to real-time flows.

To understand how real-time traffic parameters affect the responsiveness of non-real-time traffic, we have calculated worst-case response time of non-real-time flows by changing the real-time traffic parameters and the load of real-time traffic. The first scenario of our experiments generates real-time traffic which is charaterized by $\sigma = 10000$ bits and $\rho = 10000$ bps. This traffic parameters represent audio-like traffic. In the second scenario, for video-like traffic, we take the value of σ and ρ of 1 Mbits and 1.5 Mbps, respectively. In the third scenario, we increase the value of burst parameter, σ, to 1.6 Mbits but do not change rate parameter (i.e. $\rho = 1.5$ Mbps). In the fourth scenario, we generate mixed traffic patterns including audio-like traffic and video-like traffic. We take $\rho = 10^p Kbps$, where p is uniformly distributed in [1, 3]. And we take $\sigma = r * \rho Kb$, where r is uniformly distributed in [0.8, 1.6]. We take a delay requirement $d = 10^s * 30ms$,

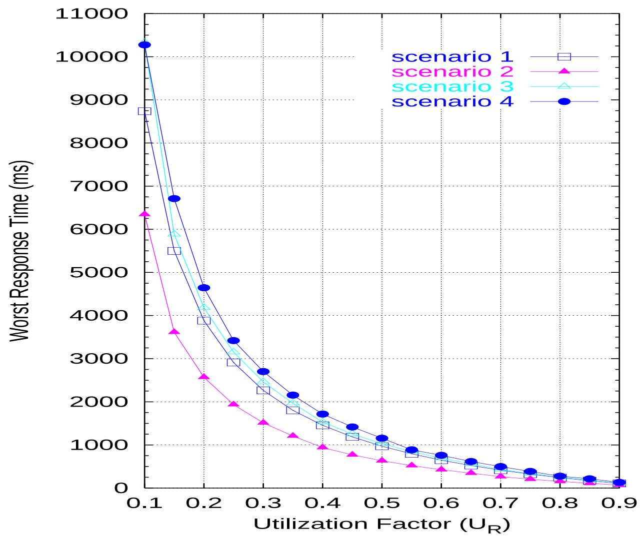

Fig. 2. Worst response time vs. Utilization factor

where s is uniformly distributed in [0, 0.52], thus d ranging in [30ms, 100ms]. The generated traffic patterns include the typical video and audio traffic [9].

In all the experiments, we compute the worst response time (i.e. deadline) of non-real-time packet using Equation (12) with varying the utilization factor (U_R). U_R means the available fraction for non-real-time traffic of the total link capacity. We assume the link has a capacity of 155 Mbps.

Figure 2 illustrates the results of our experiments. All the scenarios exhibit a similar performance. Notice that for utilization factor is greater than about 0.8, the worst response time is less than about 200 ms. However, performance becomes poor as the utilization factor tends to 0.0 (i.e. the link utilization of real-time traffic tends to 100%). We can find that the worst response time becomes large ($>$ 1 sec) when the link utilization of real-time traffic begins to be greater than 60%.

Figure 3 illustrates the effect of varying reponse time bound B_{RT} of non-real-time flows on the number of the real-time flows which can be admitted. In the experiment, we change B_{RT} from 100 ms to 1000 ms, and we use the traffic parameter of the second scenario. The y-axis means the ratio the number of admitted real-time flows to the maximum number of admitted real-time flows (i.e., B_{RT} is infinite). We vary the local deadline of the real-time flows from 30 ms to 300 ms. Figure shows that as the response time bound of non-real-time packets (B_{RT}) increases and the local deadline of the real-time flows increases, more real-time flows can be admitted.

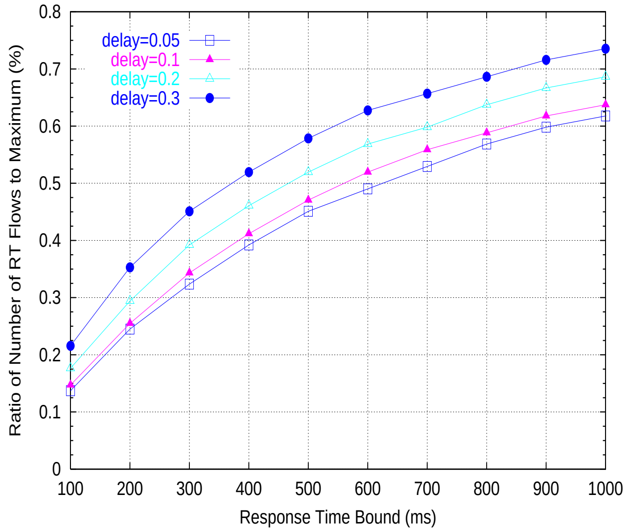

Fig. 3. Admittable real-time flows vs. Response time bound

5 Conclusion

In this paper, we presented a packet scheduling scheme which services a mix of *non-real-time* and *real-time* traffic under deadline-driven scheduling algorithm (EDF). We have developed an analytical method for obtaining the amount of *the slack time* at a given time assuming that the real-time traffic is characterized by leack bucket (σ, ρ)-model. Proposed method assigns a deadline to the non-real-time packet by calculating the slack time on-line and services the non-real-time packet along with the real-time packet using EDF algorithm. In proposed scheme, the time complexity for calculation of the slack time and deadline assignment to the non-real-time packet is $O(1)$ and thus the scheduling of non-real-time traffic can be performed *at a low cost*. Moreover, it can guarantee the schedulability of the real-time flows because it services non-real-time traffic only when the link bandwidth is not used by real-time traffic.

In proposed method, deadlines of the non-real-time packets are determined by characteristic of real-time traffic and link utilization factor. We presented an admission control algorithm which limits the number of admitted real-time flows to provide the non-real-time flows with fast response time. The experiment reveals that the responsiveness of the non-real-time flows is highly dependent on the link utilization due to real-time traffic. We also found that the response time of the non-real-flows can be improved by controlling the number of admitted real-time flows.

6 Acknowledgement

This work was performed as a part of National Research Laboratory supported by Ministry of Science and Technology in Korea (M10302000087-03J0000-044000)

References

1. M. Andrews. Probabilistic end-to-end delay bounds for earliest deadline first scheduling. In *IEEE INFOCOM*, 2000.
2. F. Chiussi and V. Sivaraman. Achieving high utilization in guaranteed services networks using early deadline first scheduling. In *IEEE/IFIP International Workshop on Quality of Service (IWQoS '98)*, 1998.
3. R.L. Cruz. A calculus for network delay, part i:network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):132–141, 1991.
4. Domenico Ferrari and Dinesh C. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, 1990.
5. Victor Firoiu, James F. Kurose, and Donald F. Towsley. Efficient admission control for EDF schedulers. In *INFOCOM (1)*, pages 310–317, 1997.
6. Leonidas Georgiadis, Roch Guerin, and Abhay K. Parekh. Optimal multiplexing on a single link: Delay and buffer requirements. *IEEE/ACM Transactions on Information Theory*, 43:1518–1535, 1997.
7. Leonidas Georgiadis, Roch Guérin, Vinod Peris, and Kumar N. Sivarajan. Efficient network QoS provisioning based on per node traffic shaping. *IEEE/ACM Transactions on Networking*, 4(4):482–501, 1996.
8. Jae H. Kim and Andrew A. Chien. Rotating combined queueing (RCQ): Bandwidth and latency guarantees in low-cost, high-performance networks. In *ISCA*, pages 226–236, 1996.
9. Edward W. Knightly, Dallas E. Wrege, Jorg Liebeherr, and Hui Zhang. Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic. In *Measurement and Modeling of Computer Systems*, pages 98–107, 1995.
10. Jörg Liebeherr, Dallas E. Wrege, and Domenico Ferrari. Exact admission control for networks with a bounded delay service. *IEEE/ACM Transactions on Networking*, 4(6):885–901, 1996.
11. C.L. Liu and J.W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the ACM*, 20(1):46–61, 1973.
12. Yeonseung Ryu. Considering non-real-time traffic in real-time packet scheduler. *Lecture Notes in Computer Science*, 2515:216–228, November 2002.
13. V. Sivaraman and F. Chiussi. Statistical analysis of delay bound violations at an earliest deadline first (edf) scheduler. In *Performance'99*, 1999.
14. Hui Zhang and Domenico Ferrari. Rate-controlled static-priority queueing. In *INFOCOM (1)*, pages 227–236, 1993.
15. K. Zhu, Y. Zhuang, and Y. Viniotis. Achieving end-to-end delay bounds by edf scheduling without traffic shaping. In *IEEE INFOCOM*, April 2001.