A DISTRIBUTED POWER ADAPTATION ALGORITHM FOR MULTIMEDIA DELIVERY OVER AD HOC NETWORKS

Jiancong Chen† S.-H. Gary Chan
Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong
Email: {kentchen, gchan}@cs.ust.hk
Fax: +852 2358-1477

Qian Zhang Wen-Wu Zhu Jin Chen
Microsoft Research Asia
5F, Beijing Sigma Center, No. 49, Zhichun Road
Haidian District, Beijing 100080,
P.R. China

Abstract—**There has been an increasing interest in recent years to deliver multimedia services over wireless ad hoc networks. Due to the existence of hidden terminal and absence of central control, the MAC protocol as used in the ad hoc networks may lead to channel capture, where some flows monopolize the channel while others suffer from starvation. This greatly degrades system throughput and fairness. After showing that static power control leads to channel capture, we propose and study a distributed dynamic power control scheme to break the capture. Our power control algorithm offers much higher fairness without compromising system throughput by better spatial reuse.**

Keywords—**Ad hoc networks, power control, fairness, throughput, channel capture, multimedia delivery**

I. INTRODUCTION

With the increasing multimedia capability in wireless devices, there has been growing interest in delivering multimedia through wireless networks. Wireless ad hoc networks bear important applications in conferencing, file exchange, and disaster relief [1]. In an ad hoc network, devices (termed "nodes" in this paper) self-organize into a communication network without any pre-established infrastructure. The nodes autonomously share a common broadcast channel and collaborate to transport information. Due to the high bandwidth requirement and delay-sensitive nature of multimedia services, transmitting multimedia using either TCP or UDP over this network still presents many challenges in terms of throughput and fairness. This is mainly because of capture effect stemming from the medium access control (MAC) protocol, the Distributed Coordination Function (DCF) in IEEE 802.11, as used in ad hoc networks.

DCF is based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) [2]. To transmit data, a node employs a 4-way handshake messaging mechanism given by request-to-send (RTS), Clear-to-Send (CTS), DATA, and ACK. During the handshake, the node reserves the channel by advertising to others in the network the duration of its transmission as indicated by its network allocation vector (NAV).

Despite the aforementioned collision avoidance mechanism, IEEE 802.11 does not eliminate collisions completely [3]. These collisions may lead to channel capture, where the common channel is monopolized by a single or a few nodes. This is the case especially when nodes use the same power (e.g., the maximum power) to transmit packets. Such capture phenomenon seriously degrades network throughput and fairness [4], [5].

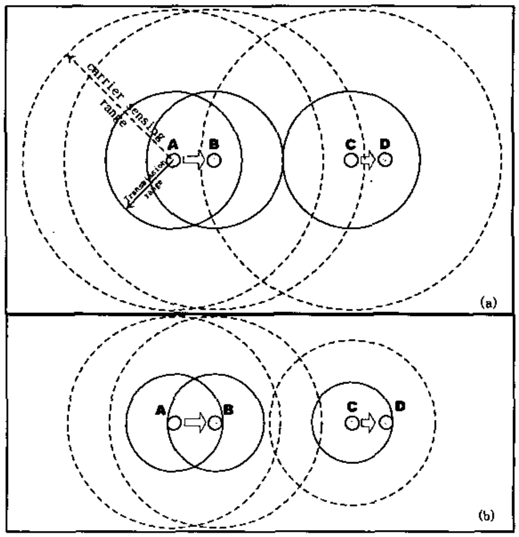


Fig. 1. IEEE 802.11 introduces channel capture. (a) A receiver capture topology; (b) Power control resolves the channel capture.

To illustrate this, we show a simple topology in Fig. 1 (a), where nodes A and C are sending packets to nodes B and D, respectively. In this scenario, C may capture the channel. We have indicated in the figure in solid lines *transmission range* of the senders, within which the receiver can correctly decode packets. Also indicated in dotted lines are the *carrier sensing range*, within which a node can sense the carrier indicating a busy channel. The carrier sensing range is typically larger (about two times larger) than the transmission range. In the figure, the ideal case is that both A and C can transmit concurrently. If this is not possible, at least they should share the channel fairly. Unfortunately, this is not the case here. Note that C is in B's carrier sensing range, but not in A's. Therefore, during A's DATA transmission to B, C does not sense any signal and hence considers the channel idle. It can therefore transmit packets, causing a collision with the A's DATA at B. Upon this collision, both A and B back off. When A resends the packet, it may collide at B with the on-going transmission between C and D. Node A hence backs off even further and eventually node C "captures" the whole channel, resulting in starvation of A and B (and hence a degradation in throughput and fairness). Since the receiver B is captured, we call this "receiver-capture" topology.

Note that the above capture is caused by fixed power, in which some nodes cannot receive RTS packets of other nodes while some other nodes are kept "silent" because they always sense carrier. If the nodes monopolizing the channel (node C in the above example) turns down its transmission power, the starved nodes may be able to respond to RTS, thereof breaking the capture. This is shown in Fig. 1 (b), where the two flows can exist concurrently using the minimum power given by the distance between the communicating nodes. The system throughput is

---

This work was supported, in part, by the Areas of Excellence (AoE) Scheme on Information Technology funded by the University Grant Council in Hong Kong (AoE/E-01/99), and by the High Impact Area of the Research Grant of Research Grant Council in Hong Kong (HIA01/02.EG06).

†Correspondence author: Jiancong Chen

hence doubled because of such spatial reuse.

Note that the use of minimum power, if not done carefully, can still lead to channel capture (as illustrated in Section II). In this paper, we study how to control the transmission power properly so as to offer better fairness and throughput by avoiding channel capture. First we show that static power control leads to channel capture and hence is not recommended. Then we propose a distributed power adaptation (DPA) algorithm to dynamically adjust the transmission power in each node to break capture and achieve higher spatial reuse. In our algorithm, a node starts its transmission with the maximum power RTS. After it succeeds in sending a packet, it "politely" uses a lower power level (while maintaining connectivity) to continue its communication to give other nodes opportunity to use the channel. If a node finds itself in the starvation state (as indicated by, for example, repeated failures in transmission), it increases its power level so that its transmission attempt can be made known to the others in the network.

We evaluate the performance of our algorithm by simulation with TCP and UDP flows. We show that DPA indeed efficiently avoids channel capture, and hence substantially improves channel throughput and fairness. Note that although our discussion is based on ad hoc networks, DPA is also applicable in non-line-of-sight (NLOS) wireless systems with fixed antennas or base-stations such as those wireless LAN networks set up in offices or on the roof-top of a buildings [6].

We briefly discuss previous work as follows. Regarding power control, much work has been done in cellular networks, where the base station acts as a central controller (see, for example, [7] and references therein). We consider power control in ad hoc networks where decision has to be made autonomously. In ad hoc networks, power control is mainly studied in the context of fixed or static control with the objective to conserve energy [8], [3], to create a desirable topology [9], [10], and to improve channel utilization [11], [12]. Little work has been on dynamic power control to break channel capture. Our simulation indicates that static power control leads to channel capture, which can be broken with a dynamic scheme.

The rest of the paper is organized as follows. In Sect. II we discuss the channel capture phenomenon with static power control. In Sect. III, we present our distributed power control algorithm to achieve better fairness and throughput. We show in Sect. IV some illustrative simulation results of our algorithm and conclude in Sect. V.

II. STATIC POWER CONTROL AND CHANNEL CAPTURE

In this section, we show that static power control often leads to channel capture phenomenon and hence starvation of flows. Therefore, a dynamic power control is highly desirable. We illustrate the capture phenomena using some topological examples.

Recall that the receiver capture topology shown in Fig. 1 (a) can be broken by reducing the transmission power, and thereof the carrier sensing range. However, using static (the minimum) power to transmit packets leads to other channel captures. To illustrate, we show in Fig. 2 (a) an ad hoc network with two flows being sent from node A to B and C to D, respectively, using minimum power. Note that C is within A's carrier sensing range, but not vice versa. Therefore, without knowing the existence of C, A can keep on transmitting packets to B, whose replies to A causes collisions, and hence exponential backoff, in C. C hence suffers from starvation. Since the source C (of flow C to D) is captured, we call this a *source capture* topology.

Channel capture is also widely observed in a hidden terminal topology as shown in Fig. 2 (b), where two sources A and C contend for a common receiver B. Note that the distance between A and B is larger than that between C and B. If all nodes use static minimum power, C is within the carrier sensing range of A, but not vice versa. A hence is not aware of C and keeps transmitting data. C hence backs off, leading to its starvation. This is hence a source capture topology with A capturing the channel.

Note that if all nodes in the network use the same power (such as the maximum power level), so that A and C are aware of each other, the source capture can be resolved. However, the use of maximum power leads to yet another channel capture called "power capture" due to the different signal power received from the two sources at the receiver. In this case, the connection of stronger signal power captures the channel, even though the signal difference is small [5]. Refer to Fig. 2 (c). Because the source C is nearer to the receiver B, C eventually captures the channel.

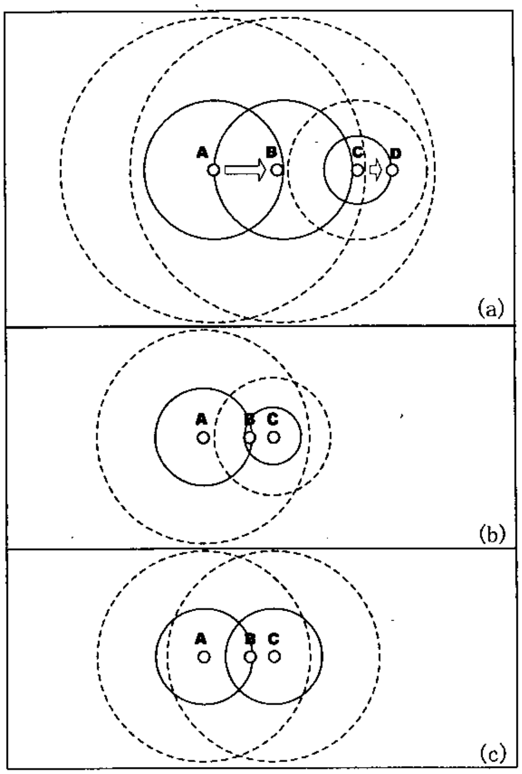


Fig. 2. Channel capture scenarios in different topologies. (a) A source capture topology; (b) Source capture in hidden terminal topology; (c) Power capture in hidden terminal topology.

The above shows that channel capture occurs because of static power level. In order to break the capture, a dynamic power control scheme can be used. Such dynamic scheme not only breaks captures and hence leads to better fairness and throughput, but also achieves lower power consumption. We propose such a scheme in the next section.

III. A DISTRIBUTED POWER ADAPTATION ALGORITHM

In this section, we propose a distributed dynamic power adaptation scheme to break the aforementioned channel capture problem. In this scheme, instead of using a fixed power level in all nodes, each node uses different transmission power to communicate with its neighbors. The power levels most recently used are maintained in a Neighbor Power Table (NPT), whose entries are shown in Table I. For each of the neighbors the node communicates with, an entry is maintained with the *Node ID*, the corresponding *Min-Power* required to maintain the connection and the power levels used for sending RTS (the *RTS power*) and CTS (the *CTS power*). The minimum power can be obtained by the handshake messaging as discussed in [12]. Note that the power to transmit RTS and CTS frames may be different for the same neighbor because their power levels are adjusted independently and dynamically to achieve bidirectional communication.

TABLE I
NEIGHBOR POWER TABLE (NPT)

Node ID	Min-Power	RTS Power	CTS Power
2	3	5	3
4	7	7	7
⋮	⋮	⋮	⋮

Algorithm 1:* *Dynamic Power Adaptation Algorithm

```
begin
    $P_{Cur} \leftarrow P_{Max}$;
    $RetryCounter \leftarrow 0$;
    $TransCounter \leftarrow 0$;
    DEC:
        Transmit a frame;
        if succeed then
            $TransCounter++$;
            $RetryCounter \leftarrow 0$;
            if $TransCounter > \alpha(P_{Cur} - P_{Min} + 1)$ then
                $TransCounter \leftarrow 0$;
                Decrease power level $P_{Cur}$;
                if $P_{Cur} = P_{Min}$ then
                    goto CON;
            else goto DEC;
        else
            $TransCounter \leftarrow 0$;
            $RetryCounter++$;
            if $RetryCounter > \beta(P_{Max} - P_{Cur} + 1)$ then
                $RetryCounter \leftarrow 0$;
                goto INC;
    CON:
        Transmit a frame;
        if succeed then goto CON;
    INC:
        Transmit a frame;
        if succeed then
            $TransCounter++$;
            $RetryCounter \leftarrow 0$;
            if $TransCounter > \alpha(P_{Cur} - P_{Min} + 1)$ then
                $TransCounter \leftarrow 0$;
                goto DEC;
            else goto INC;
        else
            $TransCounter \leftarrow 0$;
            $RetryCounter++$;
            if $RetryCounter > \beta(P_{Max} - P_{Cur} + 1)$ then
                if $P_{Cur} = P_{Max}$ then
                    Reroute or reject;
                Increase power level $P_{Cur}$;
                $RetryCounter \leftarrow 0$;
                goto INC;
end;
```

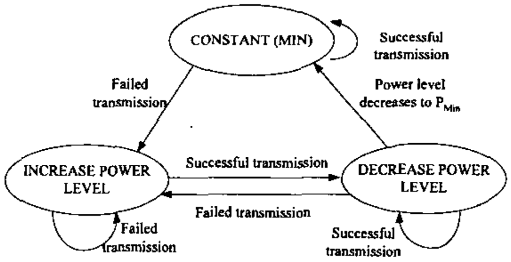


Fig. 3. The state transition diagram of the power adaptation algorithm.

When a node has a packet to send to a neighbor, it first looks up the neighbor's ID in the table. If it is not there, the node allocates a new entry with the maximum initial power level. Otherwise, it uses the RTS power level indicated to transmit the RTS and DATA frames. Upon receiving an RTS, a node looks up the table for the sender. If it is found, the indicated CTS power is used to reply the CTS frame. Otherwise, a new entry is established and the maximum power is used. The minimum desired power is sent along with the CTS to the source node. After a successful handshaking, the NPT entries for the communication in both sides are allocated. The upcoming transmissions will use dynamic power and update the table according to the *power adaptation algorithm*. The cost of maintaining NPT is low because the per-neighbor entries in the table is small given that a node is unlikely to directly communicate with more than just a few neighbors simultaneously.

In the power adaptation algorithm, we assume that all nodes have the same set of power levels labelled from P_1 to P_L, where $P_1 < P_2 < \cdots < P_L$. Let $P_{Max} = P_L$ and P_{Min} be the minimum desired power level maintaining the communication connections. Clearly, P_{Min} may be different in different nodes.

There are three states in the algorithm: constant, increase and decrease. The state transition diagram is shown in Fig. 3. Briefly speaking, when there is no transmission failure, a node stays in the 'constant' state and uses its minimum power level to transmit data. Once there are continuous failures, a node enters the 'increase' state and increases its power level to retry. After a number of successful transmissions, it goes to the 'decrease' state and decreases its power. We show in *Algorithm 1* the detailed operations in each of the states.

- 'Constant' state (CON): A node in the 'constant' state keeps using the required minimum power level P_{Min} as the current power P_{Cur} to transmit frames. This P_{Min} is determined by the connectivity requirement of the flow. After sending an RTS frame, if the node receives CTS successfully, it sends the DATA frame to the receiver. Otherwise, if the RTS frame times out, it transits to the 'increase' state to retry.
- 'Increase' state (INC): In the 'increase' state, a node first uses P_{Cur} to transmit an RTS frame. If the frame is successfully transmitted, a DATA frame follows. The $TransCounter$ is increased by one and the $RetryCounter$ is reset to zero. Otherwise, if the transmission fails, the $RetryCounter$ is increased by one and the $TransCounter$ is reset to zero. Once the $RetryCounter$ exceeds its current upper bound, the node increases its power level and stays in the 'increase' state. If the current power is already at the maximum power level, the communication request will be rerouted or rejected. On the other hand, if the $TransCounter$ exceeds its current upper bound, the node transits to the 'decrease' state.
- 'Decrease' state (DEC): The operations in the 'decrease' state is similar. After a node transmits a frame, it updates the counters. If the $TransCounter$ exceeds its upper bound, the node decreases its power level and stays in the 'decrease' state. If the current power is already at the minimum power level, the node goes back to the 'constant' state. On the other hand, if the $RetryCounter$ exceeds its upper bound, the node transits to the 'increase' state.

A more detailed discussion on the algorithm is presented in [13]. Interested readers may refer there for more parameter settings and implementation details.

IV. ILLUSTRATIVE SIMULATION AND EXPERIMENTAL RESULTS

In this section, we present the performance evaluation using NS2 (ns-2.1b8a) with the CMU wireless extension. We use 2Mbps for chan-

TABLE II

POWER LEVELS AND THE CORRESPONDING TRANSMISSION RANGES.

Power (mW)	1	2	3.45	4.8	7.25
Ranges (m)	40	60	80	90	100
Power (mW)	10.6	15	36.6	75.8	281.8
Ranges (m)	110	120	150	180	250

TABLE III

JAIN'S FAIRNESS INDEX AND THROUGHPUT COMPARISON: DPA ACHIEVES SUBSTANTIALLY BETTER FAIRNESS WITH SIMILAR THROUGHPUT.

	Jain's fairness index		System Throughput (Kbps)	
Topology	IEEE 802.11	DPA	IEEE 802.11	DPA
Hidden terminal	0.526929	0.918870	1498.7	1642.3
Source capture	0.501248	0.878585	1781.0	1711.1
Receiver capture	0.532432	0.832820	1752.1	3035.2

nel bitrate. For the radio propagation model, a two-ray path loss model is used. We do not consider fading and mobility in our simulations. We assume that the carrier sensing range is about two times larger than the transmission range. When maximum power level is used, the transmission range is 250 m and the carrier sensing range is 550 m. In our simulation, we have used 10 power levels as shown in table II with corresponding transmission ranges. All simulation results are the average of 10 runs and each simulation runs for 20 seconds of simulation times. We use the IEEE 802.11 standard as the baseline to compare with.

To evaluate the performance of our power control scheme, we simulate a random network and three typical topologies including hidden terminals and the source and receiver capture topologies. There are three nodes in the hidden terminal topology as shown in Fig. 2 (b), where the distances between them are 60 and 180 meters, respectively. The source capture topology has been shown in Fig. 2 (a), where the distances between the four nodes are given by 60, 150 and 180 meters. For the receiver capture topology shown in Fig. 1 (a) the distances between the four nodes are given by 70, 300 and 120 meters. Regarding the random network, 25 random nodes uniformly distributed within a 1000x1000 m^2 flat area. One flow of 1 Mbps originates at each node with the nearest node as its destination. Thus, a total 25 flows are generated.

We consider fairness and throughput as our performance metrics. For fairness, we have used Jain's fairness index in the form of

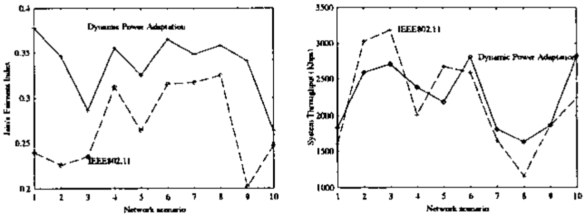
$$f(x_1, x_2, \cdots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} {x_i}^2},$$

where n is the number of flows in the system and x_i is the average throughput of flow i [14]. We define system throughput as the aggregate bitrate of all the flows in the system.

In Table III, we compare the fairness and system throughput among different topologies with and without DPA. We clearly see from the table that, our DPA scheme achieves the best fairness, much better than a system without. With respect to throughput, DPA in general achieve better throughput, showing that it does not compromise fairness with throughput.

For random network, we simulated 10 different topologies (scenarios). We plot in Fig. 4 (a) the Jain's fairness index in different networks to show the inter-flow fairness. Clearly, without DPA, the channel is not shared fairly. With DPA, the nodes often adjust their power accordingly so that they can access the channel more fairly. In Fig. 4 (b) we plot the aggravated throughput of the 25 flows in different random networks. Clearly, DPA achieves similar throughput as IEEE 802.11. In some networks, the system throughput is even higher. These results again show that DPA does not compromise fairness with throughput.

In addition to the simulation performed, we have also conducted some experiments on video delivery over the ad hoc networks with and without DPA. The objective is to show the strengths of DPA for video transmission in terms of video quality and start-up delay. Due to the limited space, we don't discuss the results here. Interested readers may also refer to [13] for more details.



(a) Inter-flow fairness comparison.
(b) Aggregated throughput comparison.

Fig. 4. Performance comparison in different random networks.

V. CONCLUSIONS

Due to the existence of the hidden terminal and a lack of central control, capture phenomenon can occur in ad hoc wireless networks. Channel capture leads to starvation in some nodes, thereof degrading the network fairness and throughput. In this paper, we first illustrate that static power often leads to channel capture in some common topologies in ad hoc networks. To break the capture, therefore, a dynamic power algorithm is needed.

We propose and study an effective distributed power adaptation algorithm (DPA), which adjusts the transmission power in each node according to the network condition. Our algorithm is simple, efficient, autonomous and without control message overhead. Simulations using TCP and UDP flows show that DPA can efficiently break starvation and hence achieve substantially better fairness without compromising throughput. Our scheme also conserves much power as compared with the static power control.

REFERENCES

[1] R. Ramanathan and J. Redi, "A brief overview of ad hoc networks: challenges and directions," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 20 – 22, May 2002.

[2] W. Dieplstraten, G. Ennis, and P. Belanger, *DFWMAC: Distributed Foundation Wireless Medium Access Control*, vol. P802.11-93/190. IEEE, 1993.

[3] E.-S. Jung and V. Nitin, "A power control MAC protocol for ad hoc networks," in *Proceedings of ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2002.

[4] F. Tobagi and L. Kleinrock, "Packet switch in radio channels: Part II-the hidden terminal problem in carrier sense multiple-access and the busy tone solution," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.

[5] C. Ware, J. Judge, J. Chicharo, and E. Dutkiewicz, "Unfairness and capture behaviour in 802.11 adhoc networks," in *Proceedings of International Conference on Communications*, pp. 159 –163, IEEE, 2000.

[6] H. Kroemer, "Non-line-of-sight wireless systems promise strong signals for high-speed Internet access," *IEEE SPECTRUM*, pp. 37–43, June 2002.

[7] S.-J. Oh, T. Olsen, and K. Wasserman, "Distributed power control and spreading gain allocation in CDMA data networks," in *Proceedings of INFOCOM 2000*, (Tel Aviv, Israel), pp. 379 – 385, IEEE, March 2000.

[8] S. Agarwal, R. Katz, S. Krishnamurthy, and S. K. Dao, "Distributed power control in ad-hoc wireless networks," in *Proceedings of Personal, Indoor and Mobile Radio Communications*, pp. 59 – 66, IEEE, Sep/Oct 2001.

[9] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proceedings of INFOCOM 2000*, (Tel Aviv, Israel), pp. 404 – 413, IEEE, March 2000.

[10] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proceedings of INFOCOM 2001*, (Anchorage, AK, USA), pp. 1388 – 1397, IEEE, April 2001.

[11] S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu, "Intelligent medium access for mobile ad hoc networks with busy tones and power control," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 9, pp. 1647 – 1657, Sept 2000.

[12] J. Monks, V. Bharghavan, and W. Hwu, "A power controlled multiple access protocol for wireless packet networks," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, pp. 1–11, IEEE, April 2001.

[13] J.-C. Chen, S.-H. Chan, Q. Zhang, W.-W. Zhu, and J. Chen, "PASA: Power adaptation for starvation avoidance to deliver wireless multimedia," *IEEE Journal on Selected Areas in Communications special issue on Wireless Multimedia*, 2003, to appear.

[14] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," DEC Research Report TR-301, Sept. 1984.