PASA: Power Adaptation for Starvation Avoidance to Deliver Wireless Multimedia

Jiancong Chen, S.-H. Gary Chan, *Member, IEEE*, Qian Zhang, *Member, IEEE*, Wen-Wu Zhu, *Senior Member, IEEE*, and Jin Chen

***Abstract*—In recent years, there has been an increasing interest to deliver multimedia services over wireless ad hoc networks. Due to the existence of hidden terminal and absence of central control, the medium access control protocol as used in the ad hoc networks may lead to channel capture, where some flows monopolize the channel while others suffer from starvation. As a consequence, the system throughput and fairness are greatly degraded. After showing that static power control leads to channel capture, we propose and study a distributed dynamic power control scheme termed "power adaptation for starvation avoidance" (PASA), which dynamically adjusts the transmission power of a node so as to avoid starvation. PASA is shown to be effective in breaking channel captures, hence improving short-term fairness among contending flows. It is simple, fully autonomous and requires no communication overhead. Via extensive simulations, we show that our power control algorithm achieves much better fairness without compromising system throughput through better spatial reuse. Our experiments with video sequences transmitting over different network topologies show that PASA achieves much better video quality with lower start-up delay and buffer requirement.**

***Index Terms*—Channel capture, fairness, multimedia delivery, power control, throughput, wireless ad hoc networks.**

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 preestablished 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 transmission control protocol (TCP) or user datagram protocol (UDP) still presents many challenges related to throughput and fairness. This is mainly due to capture effect stemming from the medium access control (MAC) protocol in IEEE 802.11, the distributed coordination function (DCF), as used in ad hoc networks.

Manuscript received October 15, 2002; revised May 1, 2003. This work was supported in part by the Research Grant Council in Hong Kong under Grant HKUST 6199/02E and under Grant HKUST 6014/01E). This paper was presented in part at the IEEE International Conference on Multimedia and Expo (ICME), Baltimore, MD, July 2003.

J. Chen and S.-H. G. Chan are with the Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: kentchen@cs.ust.hk; gchan@cs.ust.hk).

Q. Zhang, W.-W. Zhu, and J. Chen are with the Microsoft Research Asia, Beijing 100080, China.

Digital Object Identifier 10.1109/JSAC.2003.815225

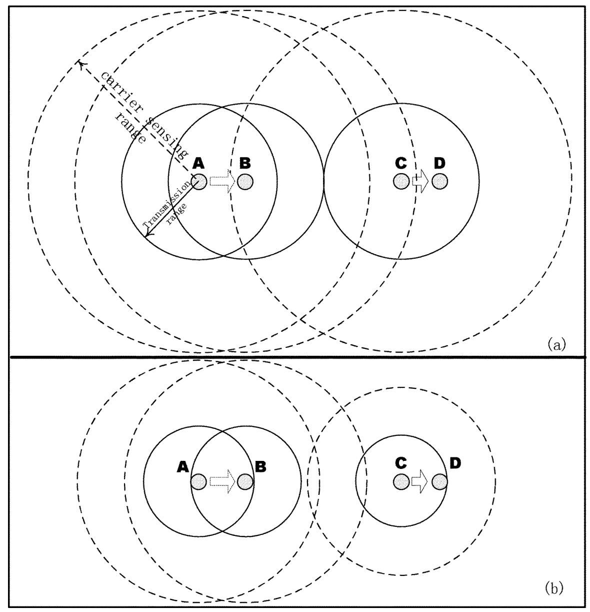


Fig. 1. IEEE 802.11 introduces channel capture. (a) A receiver capture topology. (b) Power control resolves the channel capture.

DCF is based on carrier sense multiple access with collision avoidance (CSMA/CA) [2]. To transmit data, a node employs a four-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 cannot eliminate collisions completely, which may lead to channel capture, where the common channel is monopolized by a single or a few nodes [3]. 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]–[6].

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. 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 (by about two times) than the transmission range. In this scenario, C may capture the channel. This is because 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 carrier and, hence, considers the channel idle. It can, therefore, keep transmitting packets, causing collisions with A's DATA at B. Upon a collision, both A and B backoff. When A resends the packet, it may collide again 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. It is worthwhile to point that 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.

Note that the above capture is caused by fixed power, in which some nodes (e.g., *C*) cannot detect RTS packets sent by other nodes while some other nodes (e.g., B) are kept "silent" because they always sense carrier. Observe that if the nodes monopolizing the channel (e.g., 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, hence, doubled because of such spatial reuse.

However, the use of minimum power, if not done carefully, can still lead to channel capture (we will illustrate this 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 algorithm termed power adaption for starvation avoidance (PASA) 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. Our algorithm has the following properties.

1) Simple and efficient—PASA is a control mechanism without control message overhead. It does not require any change in MAC or protocols in other layers.
2) Distributed and fully autonomous—Every node in the network adapts its power level independently.
3) Flexible—PASA is able to resolve starvation in many channel capture scenarios and implementable independent of the underlying MAC protocols. Note that although our discussion is based on ad hoc networks, PASA is also applicable in nonline-of-sight (NLOS) wireless systems with fixed antennas or base stations such as those wireless local-area network (LAN) networks set up in offices or on the rooftop of a buildings [7].
4) Fairer—PASA can achieve better short-term fairness in channel sharing among the nodes.

We evaluate the performance of our algorithm by simulation using TCP and UDP flows in different typical topologies and random networks. We show that PASA indeed efficiently avoids channel capture, and hence substantially improves channel throughput and fairness. Our experiments on video delivery using PASA algorithm show that better video quality can be achieved with lower start-up delay and buffer requirements.

There are, hence, two contributions of this work.

- We show that static power control leads to starvation in wireless ad hoc networks. Therefore, static power control is not recommended in reality.
- We propose and study an effective dynamic power control scheme to address the channel capture problem. Our control scheme achieves much better fairness without compromising throughput, and much better video quality, lower start-up delay and smaller buffer requirement.

Here, we briefly discuss previous work as follows. Capture effects have been widely studied both in Ethernet and wireless networks [8]. As opposed to the Ethernet environment, ad hoc networks have additional important considerations on power and transmission range [9]. For example, in a hidden terminal scenario, the connection with stronger power or closer to the receiver is more likely to capture the channel [5], [6]. Various protocols have also been proposed in the literature to address the fairness issues due to hidden terminal [10]–[13]. Most of them, however, focus on modifying the MAC protocol. This work differs by proposing a dynamic power control mechanism independent of (or without modifying) the MAC layer.

Regarding power control, much work has been done in cellular networks, where the base station acts as a central controller (see, for example, [14] 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 [3], [15], to create a desirable topology [16], [17], or to improve channel utilization [18], [19]. Little work has been done on dynamic power control to break channel capture. Our simulation indicates that such static power control may lead to channel capture, which can be broken with dynamic power control as in PASA.

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

II. Static Power Control and Capture Effects

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

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 types of channel capture. To illustrate, we show in Fig. 2(a) an ad hoc network with two flows from node *A* to *B* and from *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

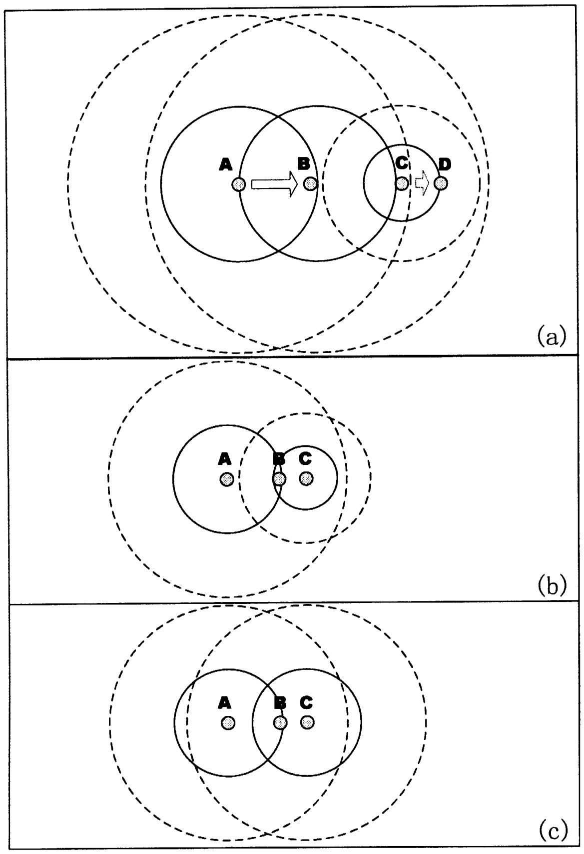


Fig. 2. Channel capture scenarios in different topologies. (a) A source capture topology. (b) Source capture in a hidden terminal topology. (c) Power capture in a hidden terminal topology.

C, *A* can keep on transmitting packets to *B*, whose replies to *A* causes collisions and, hence, exponential backoff, at *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*. Node *A*, therefore, is not aware of *C* and keeps transmitting data. The continuous backoff of *C* eventually leads to its starvation. We call this 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 level received from multiple sources at the receiver. In this case, the connection of the strongest signal power is likely to capture the channel [5]. By referring again to Fig. 2(c), since the source *C* is closer to the receiver *B*, *C* eventually captures the channel.

The examples above indicates 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, hence leading to better fairness and throughput, but also achieves lower power consumption. We propose such a scheme in the next section.

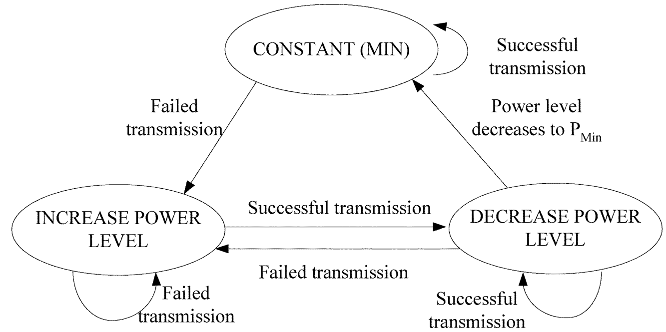


Fig. 3. State transition diagram of the power adaptation algorithm.

III. POWER ADAPTATION FOR STARVATION AVOIDANCE (PASA)

In this section, we propose a distributed dynamic power adaptation scheme to address the aforementioned capture problems. The main idea of the scheme is to adjust the transmission power in each node according to its current condition so that all the mobile nodes in the network can share the medium channel more efficiently. In the algorithm, the power is divided into L levels labeled from 1 to L with the power at level i denoted as P_i, where $P_i \in R^+$ and $P_1 < P_2 < \cdots < P_L$. We assume that the transmission power of any mobile node is upper bounded by a maximum power denoted by $P_{\rm Max}(= P_L)$ and lower bounded by a minimum power $P_{\rm Min}$. ($P_{\rm Max}$ is assumed to be the same for all nodes in this paper for simplicity though our scheme does not require it to be so. $P_{\rm Min}$ guarantees the communication connections and may be different in different nodes.) In the following, we describe the power adaptation algorithm followed by its implementation details. An example is given to illustrate the power adjustment procedure and how it breaks channel capture.

A. A Dynamic Power Adaptation Algorithm

Algorithm 1 shows how PASA works in each node by adjusting the power autonomously. In the algorithm, two counters, $TransCounter$ and $RetryCounter$, are defined to count the number of consecutive successful and failed transmissions, respectively. Each counter has an upper bound for each power level. Such bounds have to be selected carefully since it determines how many successful transmissions leads to power decrement or how many successive failures leads to power increment. The details are discussed in Section III-B.

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. If there are certain number of consecutive 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, which are further elaborated in the following.

- "Constant" state (CON): A node in the "constant" state keeps using the required minimum power level $P_{\rm Min}$ as the current power level $P_{\rm Cur}$ to transmit frames. This $P_{\rm Min}$ 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 using the same power level as P_{Cur}. 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. However, if the transmission fails, the $RetryCounter$ is increased by one and the $TransCounter$ is reset to zero. Once the $RetryCounter$ exceeds its upper bound corresponding to its current power level, 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 upper bound corresponding to its current power level, 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 decremented 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.

We illustrate in Fig. 4 a topology leading to source capture to illustrate how our power control breaks starvation. Two aggressive flows are sent from *A* to *B* and *C* to *D*. When all nodes use their minimum power level as shown in Fig. 4(a), node *C* captures the channel, (i.e., *A* is in starvation). Upon the timeout of an RTS frame, *A* increases its power level to retry. Once the interference range of *A* covers *C* as shown in Fig. 4(b), the transmission request of *A* is made known to *C*. Node *C* then backs off so that *A* has a chance to transmit data. Thus, the starvation is broken.

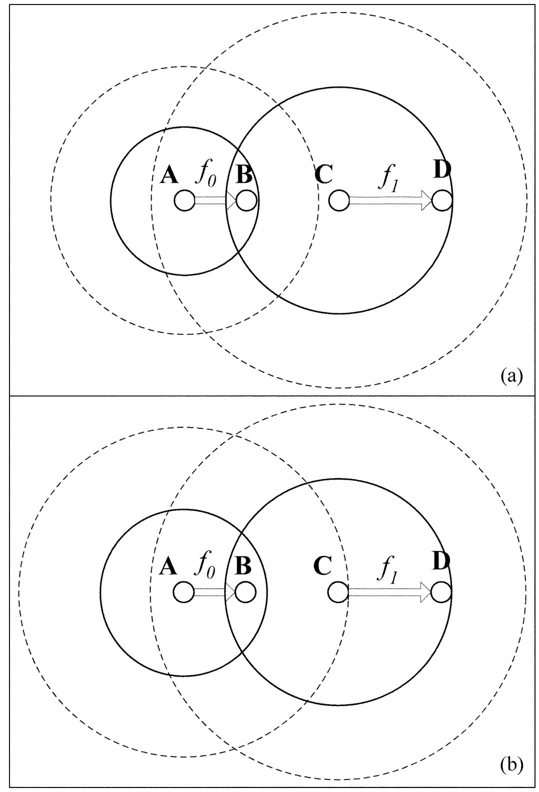


Fig. 4. PASA breaks starvation. (a) After A reduces its power. (b) After A increases its power.

Algorithm 1: PASA—the Dynamic Power Adaptation Algorithm

```
begin
  P_Cur ← P_Min;
  RetryCounter ← 0;
  TransCounter ← 0;
  CON:
    Transmit a frame;
    if succeed then goto CON;
  INC:
    Transmit a frame;
    if succeed then
      TransCounter++;
      RetryCounter ← 0;
      if TransCounter > α(P_Cur − P_Min + 1) then
        TransCounter ← 0;
        goto DEC;
      else goto INC;
    else
      TransCounter ← 0;
      RetryCounter++;
      if RetryCounter > β(P_Max − P_Cur + 1) then
        if P_Cur = P_Max then
          Reroute or reject;
        Increase power level P_Cur;
        RetryCounter ← 0;
        goto INC;
  DEC:
    Transmit a frame;
    if succeed then
      TransCounter++;
      RetryCounter ← 0;
      if TransCounter > α(P_Cur − P_Min + 1) then
        TransCounter ← 0;
        Decrease power level P_Cur;
        if P_Cur = P_Min then
          goto CON;
      else goto DEC;
    else
      TransCounter ← 0;
      RetryCounter++;
      if RetryCounter > β(P_Max − P_Cur + 1) then
        RetryCounter ← 0;
        goto INC;
end;
```

TABLE I
NEIGHBOR POWER

Node ID	Distance	Min-Power	RTS Power	CTS Power
2	80	3	5	3
4	170	7	7	7
⋮	⋮	⋮	⋮	⋮

B. Implementation Issues

In the power adaptation scheme, many implementation issues need to be addressed; specifically, how to maintain the minimum power level required for each node, how to adjust power and how to determine the upper bound of the aforementioned counters. We discuss these issues in detail in this section.

1) Neighbor Power Table (NPT): In an ad hoc network, a node can communicate directly with all of its neighbors within its signal range. Note that the signal coverage is different for different power levels. Instead of using a fixed power level for all nodes in the network, each node chooses different transmission powers to communicate with its neighbors. The power setting recently used to communicate with other nodes is maintained in a NPT, as shown in Table I. Each neighbor has a unique node ID (a unique identifier in the network), and each row of the table stores the information of the *distance* in meters between the transmitter and receiver. The entries *Min-Power* corresponds to the power required to maintain the connection, and the *RTS power* and *CTS power* corresponds to the power used to transmit the RTS and CTS frames, respectively. DATA frames are transmitted with the same power as the previous RTS frames. The minimum power required is maintained according to the distance, which can be obtained through some location service. Note that the power to transmit RTS and CTS frames may be different for a neighbor because their power levels are adjusted independently by the node. In other words, when bidirectional communications are concurrently going on between two nodes, the power of sending and receiving may be different.

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 minimum initial power level. Otherwise, it uses the RTS power level indicated to transmit the RTS and DATA frames. On the other hand, if a node needs to send a responsive CTS or ACK, it uses the CTS power level. If the transmission fails, the node increases the corresponding power level to retry according to the algorithm and updates the table. The cost of maintaining NPT is low because the table is small given that a node is unlikely to directly communicate with more than just a few neighbors simultaneously.

2) Power Increment and Decrement: Note that a node in the network continuously adjusts its power in order to transmit its frames successfully but not staying at a higher power level than necessary. In designing the power increment and decrement algorithm, we have the following objectives in mind:

a) The power level should increase fast enough so that transmission can be successful before the retry limit in the MAC protocol is exceeded.
b) The power level should not fluctuate too quick or too much.
c) The nodes should share the channel "fairly."

Given these objectives, we have used a binary increment and linear decrement method, in which

$$P_{\rm Cur} \leftarrow \left\lceil \frac{(P_{Cur} + P_{\rm Max})}{2} \right\rceil \tag{1}$$

to increase power level and

$$P_{\rm Cur} \leftarrow P_{\rm Cur} - 1 \tag{2}$$

to decrease power level. Using the increment method, it only takes at most $\log_2 P_{\rm Max}$ times to reach the maximum power level.

3) Minimum Power and Mobility: In practice, the minimum power is needed to reach the next hop of the flow, especially in a mobile environment. Obviously, if such power is not maintained, when a node moves out of the signal range, a broken flow would result, leading to an excessive failure in transmission. Therefore, each connection maintains the minimum power for the next hop and a node keeps its power above such minimum power when it is adjusting the level.

Note that the minimum power may vary with time, due to different network condition such as fading, mobility or introduction of new nodes. Hence, the field corresponding to the minimum power is updated continuously to take such network dynamics into account. Given that PASA does not critically depend on the accurate determination of the minimum power, we may simply use a crude model based on the distance between the two nodes, which can be obtained by approaches such as beacons and location services [20].

4) Retry Counter and Transmission Counter: Clearly, if the upper bound of the retry counter is set too low, the power adaptation would be too sensitive to the transmission failures caused by insufficient power and bit errors. On the other hand, if it is set too high, a node would waste much time in retrying when the power is not sufficient. For the same reason, the upper bound of the transmission counter is also needed to be chosen carefully for better performance.

In our algorithm, these upper bounds are dynamic and relative to the current power level used. The upper bound of retry times equals to

$$\alpha \left(P_{\rm Cur} - P_{\rm Min} + 1\right). \tag{3}$$

Clearly, α determines how sensitive the power adjustment reacts to transmission failures. If α is small, then the power adjustment reacts quickly. On the other hand, if α is large, it reacts slowly. Meanwhile, when the current power level is close to the minimum, it is easy for a node to increase power when there is failure. Otherwise, the node with high power level would retry more times before increasing power. Likewise, the upper bound of continuous transmission times ($TransCounter$) is given by

$$\beta \left(P_{\rm Max} - P_{\rm Cur} + 1\right) \tag{4}$$

where β is also a parameter to adjust the sensitivity. Both α and β are related to the number of power levels. Via extensive simulations, we find that $\alpha = 1$ and $\beta = 4$ achieve good performance with a total number of power levels of ten. By choosing such values, nodes can share the channel fairly and a node would not stay at a high power level for a long time.

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

IV. ILLUSTRATIVE SIMULATION AND EXPERIMENTAL RESULTS

In this section, we present the performance of PASA under various network topologies using NS2 (ns-2.1b8a) with the CMU wireless extension, as well as some experimental results for video transmitting over a network with and without PASA. We compare PASA to the static power scheme with the power levels chosen to the minimum power to reach the desired node. The wireless physical layer in NS2 is a model of a DSSS radio interface (Lucent WaveLan Direct-Sequence Spread-Spectrum) operating at 914 MHz with a throughput of 1952 kb/s. We implement our power control algorithm at the physical layer. As mentioned before, the power levels indicate the signal cover range. When the power level is increased by one, the coverage extends Δ meters, i.e., if the coverage of power level P_i is R_{P_i}, $i \in Z^+$, we hence have

$$R_{P_i} = R_{P_0} + i\Delta. \tag{5}$$

For the radio propagation, a two-ray path loss model is used. We do not consider fading and mobility in our simulations. We use a simple distance/power relationship to estimate the attenuation of the transmitted radio signal: The attenuation in dB is given by

$$P = P_0 + u \log\left(\frac{R}{R_0}\right) \tag{6}$$

where R is the distance between the transmitter and the receiver, and u is the path loss exponent. By selecting a certain value of Δ and R_{P_0}, a list of corresponding discrete power levels can then be generated. 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 ten power levels as shown in Table II with corresponding transmission ranges. All simulation results are the average of ten runs and each simulation runs for 20 s of simulation times.

To evaluate the performance of our power control scheme, we simulate different random networks 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 m, 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 m. For the receiver capture topology shown in Fig. 1(a) the distances between the four nodes are given by 70, 300, and 120 m. Regarding the random network, 25 random nodes uniformly distributed within a 1000×1000 m^2 flat area. One flow of 1 Mb/s originates at each node with the nearest node as its destination. Thus, a total 25 flows are generated.

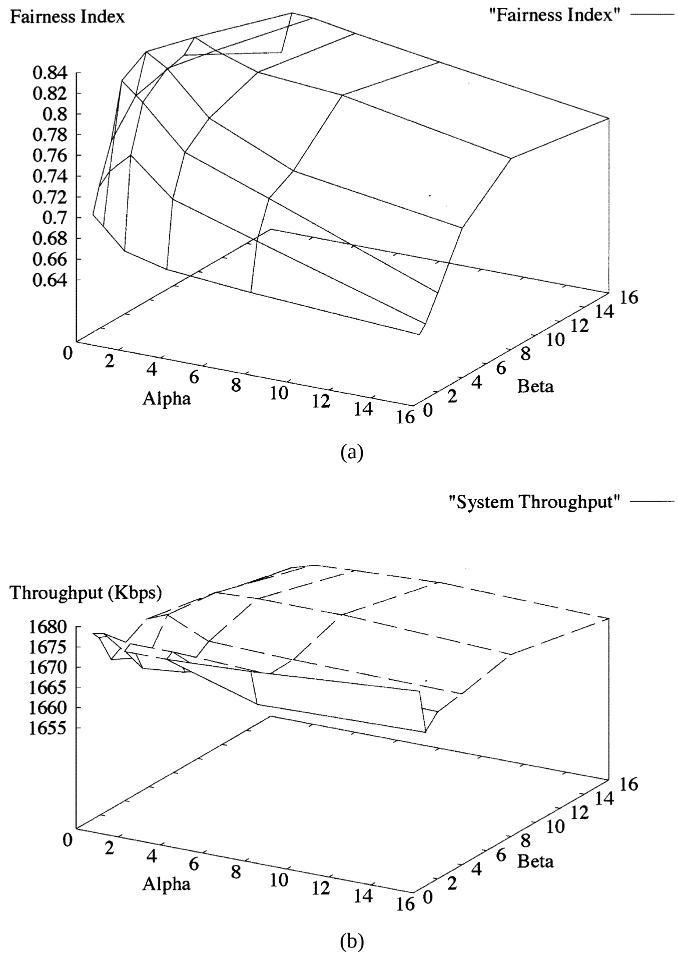


Fig. 5. Performance with respect to different values of α and β. (a) System throughput. (b) Average Jain's fairness index.

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

$$f(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{\left(n \sum_{i=1}^{n} {x_i}^2\right)} \tag{7}$$

where n is the number of flows in the system and x_i is the average throughput of flow i [21]. We define system throughput as the aggregate bitrate (or individual "throughput") of all the flows in the system, i.e.,

$$\text{System Throughput} = \sum_{i=1}^{n} x_i. \tag{8}$$

We compute the average sending bitrate of each flow every 0.5 second and plot the bitrate curve versus time for the source capture and hidden terminal topologies.

We first discuss the setting of our system parameters α and β. In Fig. 5, we show the performance of PASA with respect to the choice of α and β in terms of system throughput and fairness for the source capture environment. We can clearly see from Fig. 5(a) that the system throughput is quite insensitive

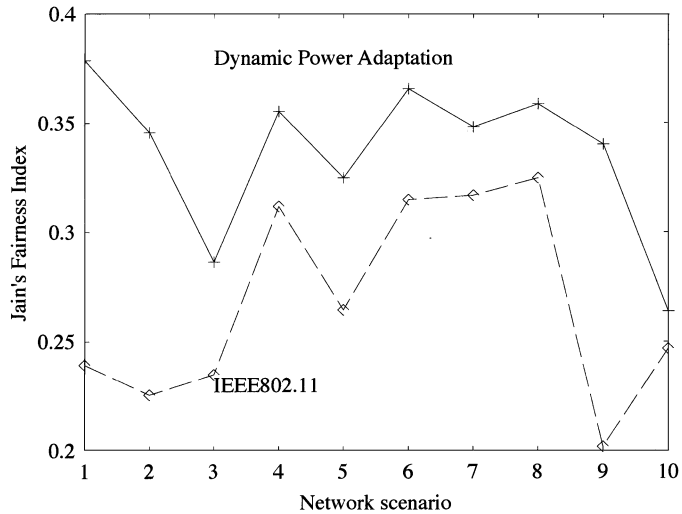
TABLE III
JAIN'S FAIRNESS INDEX AND THROUGHPUT COMPARISON: PASA ACHIEVES SUBSTANTIALLY BETTER FAIRNESS WITH SIMILAR THROUGHPUT

Topology	Jain's fairness index		
	without PASA	PASA	PASA w/o P_{Min}
Hidden terminal	0.526929	0.918870	0.890253
Source capture	0.501248	0.878585	0.865138
Receiver capture	0.532432	0.832820	0.801723
Random network	0.263292	0.346057	0.315738
Topology	System Throughput (Kbps)		
	without PASA	PASA	PASA w/o P_{Min}
Hidden terminal	1498.7	1642.3	1536.0
Source capture	1781.0	1711.1	1643.6
Receiver capture	1752.1	1735.2	1683.2
Random network	2356.2	2285.7	1862.8

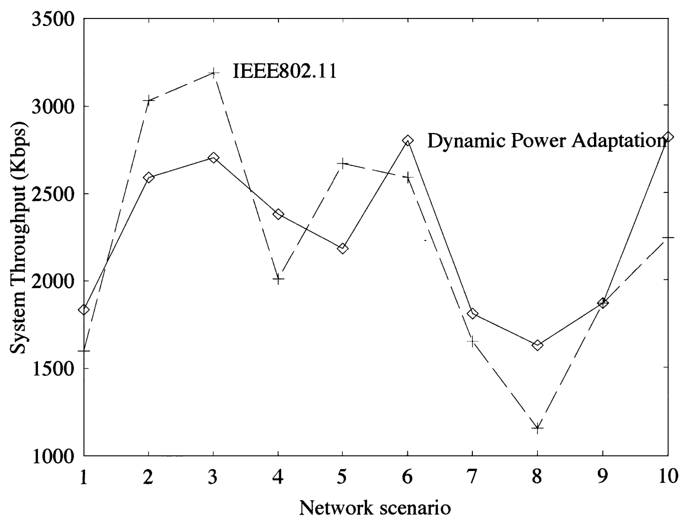
to the choice of α and β, and remains more or less the same for different combinations of α and β. The choice of α and β mainly affects the fairness among flows. In Fig. 5(b), we show Jain's index versus different combinations of α and β. Given β, there is an optimal α to achieve the highest fairness. This α, as β ranges, is around 1–2. On the other hand, given α there is an optimal β to achiever the best fairness (around 2–8). Clearly, there is an optimal choice of α and β in order to achieve the best fairness. This optimum is at $(\alpha, \beta) = (1, 4)$, which we will use throughout simulations.

In Table III, we compare the fairness and system throughput among different topologies with and without PASA, and, in addition, with an operation without the setting of the power floor P_{Min} (i.e., the node may visit P_1 if it wishes). We clearly see from the table that, our PASA scheme achieves the best fairness, much better than a system without. When P_{Min} is not set, the fairness suffers only slightly. With respect to throughput, PASA in general achieve better throughput, showing that it does not compromise fairness with throughput. We again see that the case without P_{Min} only suffers slightly in terms of throughput. The table shows that, in reality, P_{Min} does not need to be determined or known exactly in order to achieve a good performance. Indeed, PASA is self-adaptable in nature and the node would adjust its power continuously and dynamically so as to find this P_{Min}. Our simulation traces confirm that the power of a node would not visit a level far below P_{Min} for long before its level jumps back up.

We show in Fig. 10 the sending bitrate of the two sources for TCP and UDP flows in a hidden terminal topology. Clearly, when static minimum power scheme is used, node *C* enjoys a high bitrate at the sacrifice of *A* which is essentially starved. This is because the source *C*, hidden from the other, occupies the channel resulting in starvation of *A*. Node *A* has little chance to access the channel and, hence, its sending bitrate is almost zero. However, its attempts to transmit still conflict with the transmission of *C* resulting in decreasing the throughput as we can see in Fig. 10(b). The attempts of *A* collide with the transmission of *C* such that the sending bitrate of it is decreased sometimes. Clearly, after adopting our dynamic power control scheme, the two flows have similar bitrate most of the time. In other words, the two sources share the channel much more fairly. Such fairness is achieved by the power adjustment according to the status of each node.



(a)



(b)

Fig. 6. Performance comparison in different random networks. (a) Interflow fairness comparison. (b) Aggregated throughput comparison.

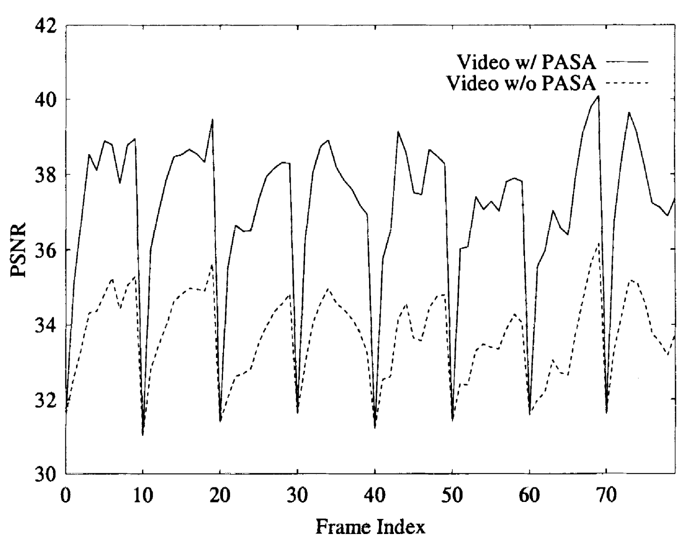
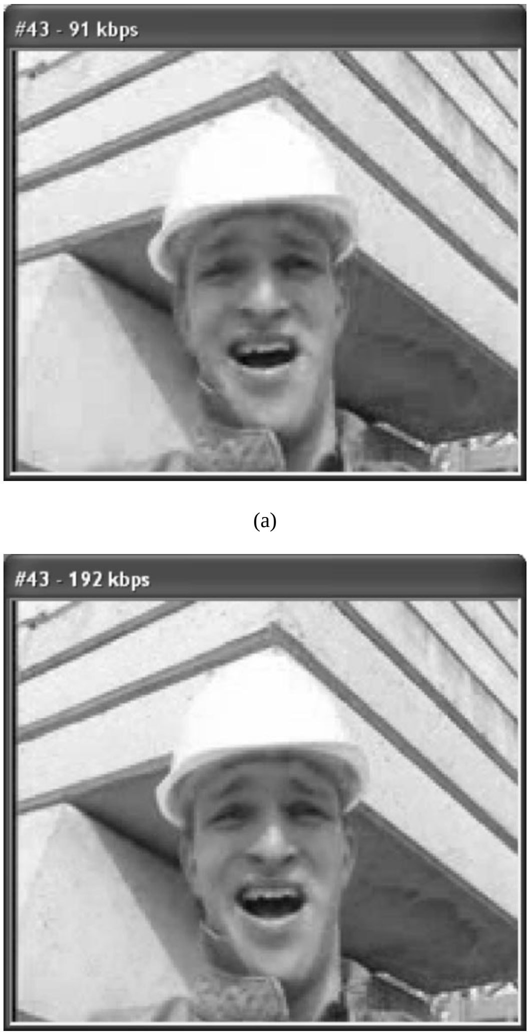
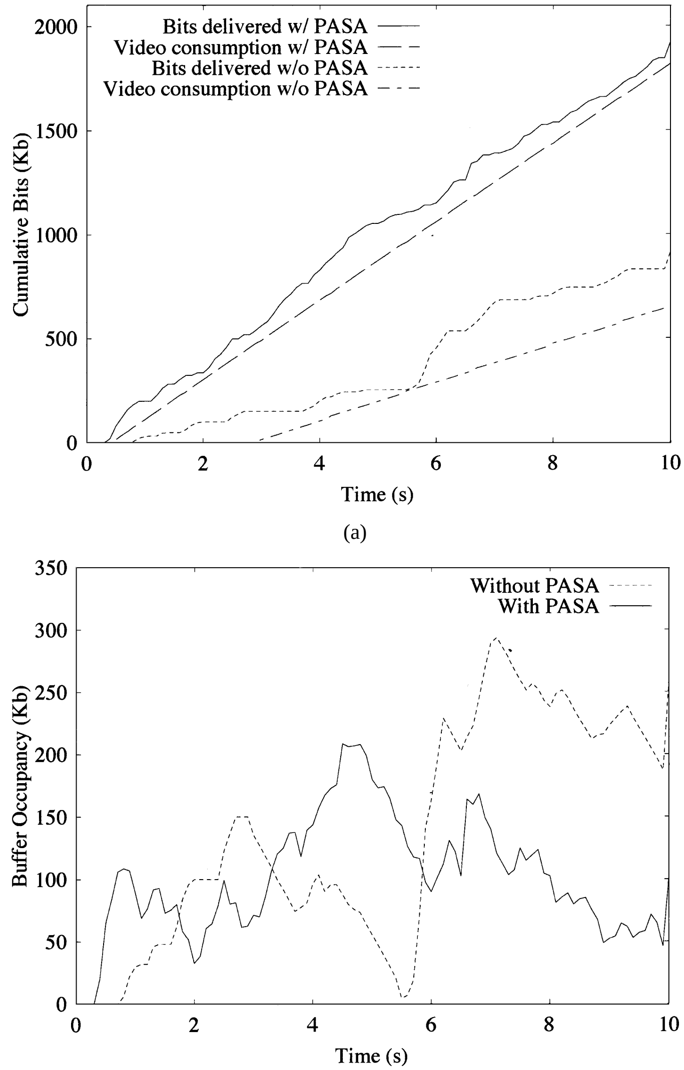


Fig. 7. PSNR comparison of the video transmitted with and without PASA.



(b)

Fig. 8. Subjective (visual) performance of a video with and without PASA for the foreman sequence. (a) Without PASA. (b) With PASA.



(b)

Fig. 9. Start-up delay and buffer occupancy comparison at the client with and without PASA. (a) Start-up delay comparison. (b) Buffer occupancy at the client.

To further understand the behavior of PASA, we show in Fig. 10(d) the power level of the two UDP sources with PASA. Recall that a node increases its power if it fails to send a number of frames, and decreases its power otherwise. Clearly, node *A* needs to increase its power level once the power drops to four, which is the minimum power required. After increasing its power level, it transmits a succession of frames successfully and decreases its power. Therefore, node *A* always adjusts its power according to whether the transmission is successful. Meanwhile, since node *C* can often transmit data successfully, it usually stays at its minimum power level. Thus, PASA does not significantly increase power consumption.

In Fig. 11, we plot the bitrate versus time and the power adaptation curve in the source capture topology (as shown in Fig. 2(a) for TCP and UDP flows. Clearly, there is a significant difference between the cases with and without PASA. Without deploying PASA, *C* monopolizes the channel, resulting in starvation to node *A*. With PASA, node *A* can compete for many more chances to access the channel by adjusting its power level. From the power adjustment curve shown in Fig. 11(d), we can see similar behavior, as it is in the hidden terminal topology.

For random networks, we simulated ten different topologies (scenarios). We plot in Fig. 6(a) the Jain's fairness index in different networks to show the interflow fairness. Clearly, without PASA, the channel is not shared fairly. With PASA, the nodes often adjust their power accordingly so that they can access the channel more fairly. In Fig. 6(b), we plot the aggravated throughput of the 25 flows in different random networks. Clearly, PASA achieves similar throughput as IEEE 802.11. In some networks, the system throughput is even higher. These results again show that PASA does not compromise fairness with throughput.

In addition to the simulation performed, we have also conducted some experiments on video delivery over wireless ad hoc networks with and without PASA. The objective is to show the strengths of PASA for video transmission in terms of video quality and start-up delay. We use a random network topology and traffic pattern in one of the simulation runs where with PASA the average throughput is 192 kb/s, while without PASA, the average throughput is 91 kb/s. We present the video quality for the SQCIF foreman sequence encode with H.263+, with

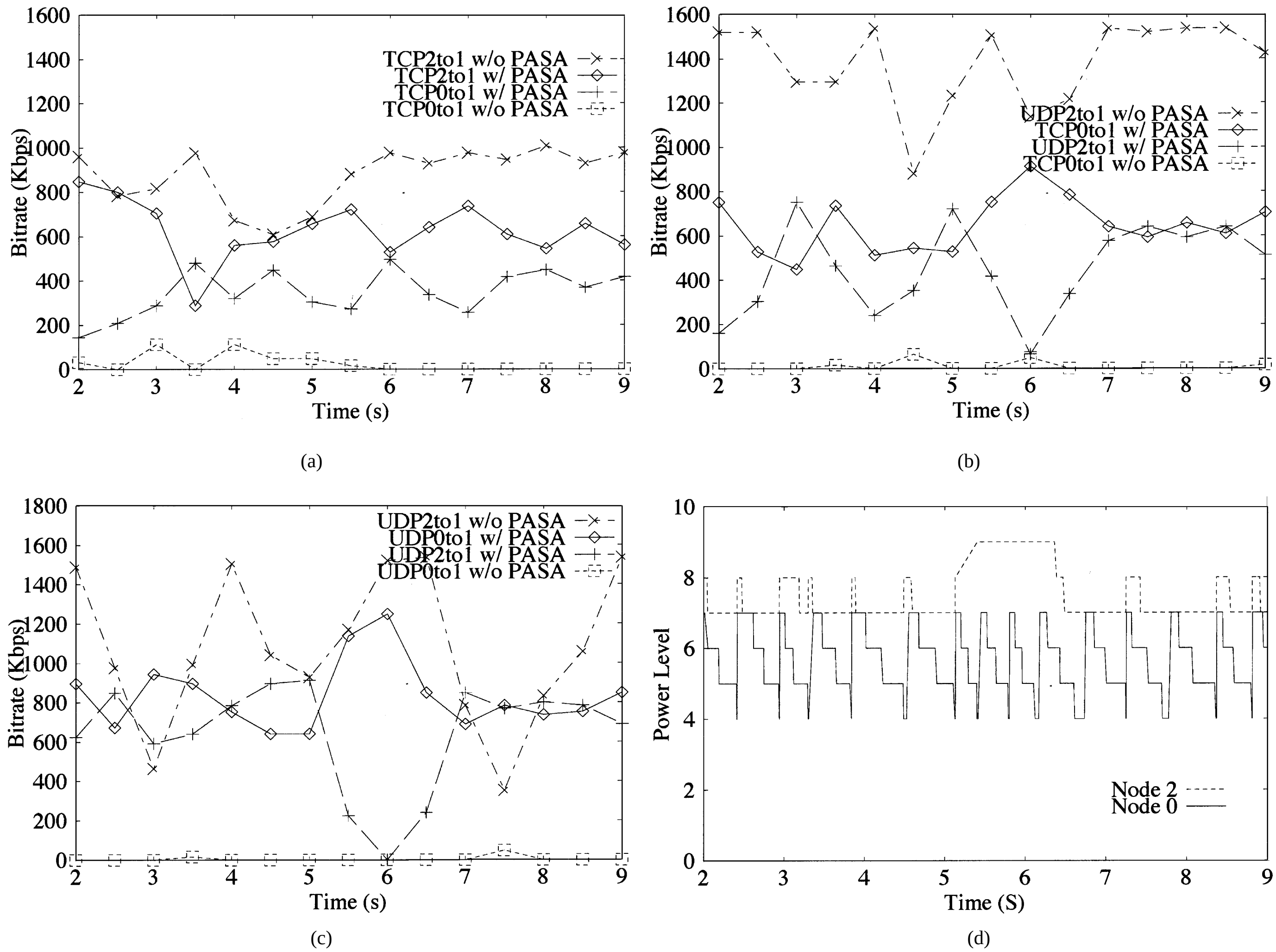


Fig. 10. Sending bitrate comparison and power adjustment in the hidden terminal topology. (a) Two TCP flows. (b) One TCP and one UDP flows. (c) Two UDP flows. (d) Power adjustment of the UDP sources in (c).

GOP = 10, frame skip = 2, and 8 frames/s. We use the objective measure PSNR given by

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{x}_i)^2}\right) \tag{9}$$

where N is the number pixels in a frame, x_i and $\hat{x}_i$ is the value of a original and encoded pixel, respectively. The value 255 is due to the peak value of a 8-bit quantized pixel. In transmitting the video sequence, rate control is used so that the resulting video bitrate matches with the average channel throughput.

We show in Fig. 7 the PSNR of the frames with and without PASA. Clearly, the PSNR with PASA is substantially better than that without (the average PSNR for PASA is 37.1 dB, while that without is 33.7 dB). The "dips" in the curves are the I-frames. Since the rate controller tries to allocate equal number of bits to all the frames, the resultant I-frame suffers from a degradation in quality due to its higher quantization parameter (QP) as compared with the P-frames (Mean QP for I-frames is 13 as compared with 3.4 and 6.3 for P-frames in the two video streams with and without PASA).

Besides the objective measure such as PSNR, in Fig. 8, we make subjective (visual) comparison between the video resulted from using PASA and without. Clearly, due to the lower QP used, the video with PASA is much better than that without.

We finally study the startup delay and buffer requirement for the transmission of the video sequence. Note that the throughput of the flow fluctuates with time (refer to Figs. 10 and 11). In Fig. 9, we plot the cumulative bits delivered to node 14, i.e., the production curve, in the simulation run versus time for the one with PASA (solid line) and without (dotted line). We clearly see that their throughputs fluctuate with time and the throughput of PASA higher than without. Also, shown in dashed lines are the cumulative frames (in bits) for the respective video, i.e., the consumption curves. The curves are nearly straight as the rate controller encodes the video with almost constant bitrate. In order to play back video continuously without buffer starvation, the consumption curve must lie below the production curve, with their difference being the buffer requirement at the client (Fig. 9(b). The intercept of the consumption curve on the x-axis is, hence, the start-up delay of the video (the prebuffer time).

We clearly see from the figure that the start-up delay with PASA is much lower (< 1 s) than that without (> 3 s). Fig. 9(b) shows that the buffer requirement with PASA (solid line) is

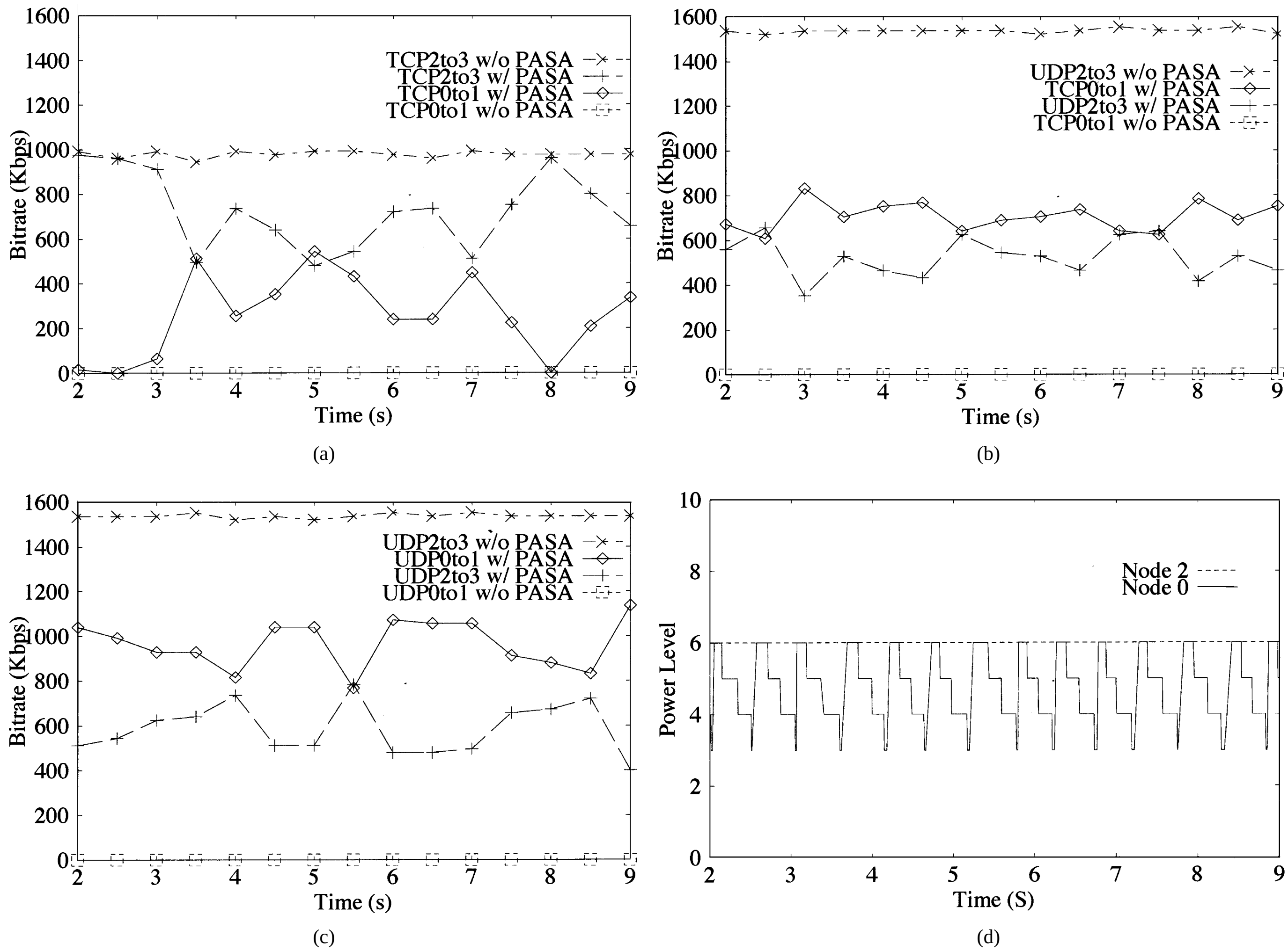


Fig. 11. Sending bitrate comparison and power adjustment in the source capture topology. (a) Two TCP flows. (b) One TCP and one UDP flows. (c) Two UDP flows. (d) Power adjustment of the UDP sources in (c).

lower than without (dotted line). This is due to the fact that the channel bandwidth with PASA is less bursty. Hence, the client only need to buffer smaller traffic bursts compare to the case without PASA.

V. Conclusion

Due to the existence of hidden terminal and a lack of central control, channel capture phenomenon can occur in wireless ad hoc networks. 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 for starvation avoidance called PASA, 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 PASA 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. Experiments with video transmission show that PASA achieves much better video quality with lower delay and buffer requirements.

References

[1] R. Ramanathan and J. Redi, “A brief overview of ad hoc networks: Challenges and directions,” *IEEE Commun. Mag.*, vol. 40, pp. 20–22, May 2002.

[2] W. Dieplstraten, G. Ennis, and P. Belanger, DFWMAC: Distributed Foundation Wireless Medium Access Control, vol. 802.11–93/190, 1993.

[3] E.-S. Jung and V. Nitin, “A power control MAC protocol for ad hoc networks,” presented at the ACM Int. Conf. Mobile Computing and Networking (MobiCom), 2002.

[4] E. 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 Trans. Commun.*, vol. 23, pp. 1417–1433, 1975.

[5] C. Ware, J. Judge, J. Chicharo, and E. Dutkiewicz, “Unfairness and capture behavior in 802.11 ad hoc networks,” in *Proc. Int. Conf. Communications*, June 2000, pp. 159–163.

[6] C. Ware, J. Chicharo, and T. Wysocki, “Simulation of capture behavior in IEEE 802.11 radio moderns,” in *Proc. Vehicular Technology Conf.*, Oct. 2001, pp. 1393–1397.

[7] H. Kreemer, “Non-line-of-sight wireless systems promise strong signals for high-speed internet access,” *IEEE Spectrum*, pp. 37–43, June 2002.

[8] K. K. Ramakrishnan and H. Yang, “The Ethernet capture effect: Analysis and solution,” in *Proc. Local Computer Networks*, Minneapolis, MN, Oct. 1994, pp. 228–240.

[9] M. D. Orange and K. W. Sowerby, “Capture effect in a Rayleigh faded cellular packet speech system,” *Electron. Lett.*, vol. 32, pp. 298–299, Feb. 1996.

[10] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Solutions to hidden terminal problems in wireless networks," in *Proc. SIGCOMM*, 1997, pp. 39–49.

[11] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LANS," in *Proc. SIGCOMM*, London, U.K., Sept. 1994, pp. 212–225.

[12] B. Bensaou, Y. Wang, and C. C. Ko, "Fair medium access in 802.11 based wireless ad-hoc networks," in *Proc. Mobile and AD Hoc Networking and Computing*, Boston, MA, Aug. 2000, pp. 99–106.

[13] T. Nandagopal, T.-F. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *Proc. Mobile Computing and Networking*, Boston, MA, Aug. 2000, pp. 87–98.

[14] S.-J. Oh, T. Olsen, and K. Wasserman, "Distributed power control and spreading gain, allocation in CDMA data networks," in *Proc. INFOCOM 2000*, Tel-Aviv, Israel, Mar. 2000, pp. 379–385.

[15] S. Agarwai, R. Katz, S. Krishnamurthy, and S. K. Dao, "Distributed power control in ad-hoc wireless networks," in *Proc. Personal, Indoor and Mobile Radio Communications*, Sept./Oct. 2001, pp. 59–66.

[16] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proc. INFOCOM 2000*, Tel-Aviv, Israel, Mar. 2000, pp. 404–413.

[17] R. Wattenhofer, L. L. P. Bahl, and Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proc. INFOCOM 2001*, Anchorage, AK, Apr. 2001, pp. 1388–1397.

[18] 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 J. Select. Areas Commun.*, vol. 18, pp. 1647–1657, Sept. 2000.

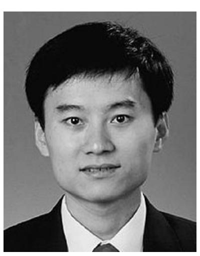
[19] J. Monks, V. Bharghavan, and W. Hwu, "A power controlled multiple access protocol for wireless packet networks," in *Proc. IEEE Conf. Computer Communications (INFOCOM)*, Apr. 2001, pp. 1–11.

[20] M. Mauve, J. Widiner, and H. Hactenstein, "A survey on position-based reuting in mobile ad hoc networks," *IEEE Network*, vol. 15, pp. 30–39, Nov./Dec. 2001.

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

Jiancong Chen received the B.S. degree in computer science with a minor in economics from Peking University, Beijing, China, in 1999. Currently, he is working toward the Ph.D. degree in computer science at Hong Kong University of Science and Technology, Clear Water Bay, Kowloon.

His research interest includes wireless communications networks and Internet technologies.



S.-H. Gary Chan (S'89–M'98) received the M.S.E. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1999, respectively, with a minor in business administration. He received the B.S.E. degree (highest honor) in electrical engineering from Princeton University, Princeton, NJ, in 1993, with certificates in engineering physics, applied and computational mathematics, and engineering management systems.

He is currently an Assistant Professor with the Department of Computer Science, The Hong Kong University of Science and Technology, and an Adjunct Researcher with the Microsoft Research Asia, Beijing, China. He was a Visiting Assistant Professor of networking at the Department of Computer Science, University of California, Davis, from September 1998 to June 1999. From 1992 to 1993, he was a Research Intern at the NEC Research Institute, Princeton, NJ. His research interest includes multimedia networking, peer-to-peer multicast networks, high-speed and wireless communications networks, and Internet technologies and protocols.

Dr. Chan was a William and Leila Fellow at Stanford University from 1993 to 1994. At Princeton, he was the recipient of the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence in 1993. He is a Member of Tau Beta Pi, Sigma Xi, and Phi Beta Kappa.

Qian Zhang (M'00) received the B.S., M.S., and Ph.D. degrees from Wuhan University, China, in 1994, 1996, and 1999, respectively, all in computer science. She joined Microsoft Research, Asia, Beijing, China, in July 1999, as an Associate Researcher in the Internet Media Group and now is a Researcher of the Wireless and Networking Group.

She has published over 40 refereed papers and is the inventor of several pending patents. Her current research interest includes multimedia delivery over wireless, Internet, next-generation wireless networks, P2P network/ad hoc network. Currently, she is participating in TCP/IP header compression in ROHC WG in IETF. She is the principal contributor of the IETF ROHC-TCP WG draft.

Wen-Wu Zhu (S'92–M'97–SM'01) received the B.E. and M.E. degrees from the National University of Science and Technology, China, in 1985 and 1988, respectively, the M.S. degree from Illinois Institute of Technology, Chicago, and the Ph.D. degree from Polytechnic University, Brooklyn, NY, in 1993 and 1996, respectively, all in electrical engineering.

From August 1988 to December 1990, he was with the Graduate School, University of Science and Technology of China (USTC), and the Institute of Electronics, Chinese Academy of Sciences, Beijing, China. He joined Microsoft Research, Beijing, in 1999 as a Researcher in the Internet Media Group. He is currently a Research Manager of the Wireless and Networking Group. Prior to his current post, he was with Bell Labs, Lucent Technologies, Whippany, NJ, Holmdel, NJ, and Murray Hill, NJ, as a Member of Technical Staff from 1996 to 1999. While he was with Bell Labs, he performed research and development in the areas of Internet video, video conferencing, and video streaming over IP networks. He has published over 100 refereed papers in international leading journals and key conferences in the areas of wireless/Internet video transport, wireless/Internet multimedia communications and networking, and multimedia signal processing, and has contributed to the IETF ROHC WG draft on robust TCP/IP header compression over wireless links. He is the inventor of more than 12 pending patents. His current research interest is in the area of wireless/Internet multimedia delivery, and delivery networks.

Dr. Zhu has served as Guest Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY for the Special Issues on Streaming Video and Wireless Video. He received the Best Paper Award in the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2001. He is a Member of Eta Kappa Nu, the Visual Signal Processing and Communication Technical Committee, and the Multimedia System and Application Technical Committee of the IEEE Circuits and Systems Society. He is also a Member of the Multimedia Communication Technical Committee of the IEEE Communications Society.

Jin Chen received the B.E. and M.E. degrees from Beijing University of Aeronautics and Astronautics (BUAA), Beijing, China, in 1998 and 2001, respectively, all in computer science.

She joined Microsoft Research, Asia, Beijing, China, in April 2001, as an Assistant Researcher in the Wireless and Networking Group. Her current research interests focus on architecutres and protocols for wireless networks and Internet, mobile and personal communications, and grid computing.