

QoS Assurances Through Class Selection and Proportional Differentiation in Wireless Networks

Kuang-Ching Wang, *Member, IEEE*, and Parameswaran Ramanathan, *Senior Member, IEEE*

Abstract—Quality-of-service (QoS) in wireless ad hoc networks is adversely affected by node mobility, changing network topologies, and uncontrolled medium contention. The paper addresses the challenges in concurrently providing a wide range of end-to-end throughput and delay assurances in such networks. The proposed solution is based on the neighborhood proportional delay differentiation (NPDD) service model. With NPDD, applications achieve their desired end-to-end QoS using dynamic class selection (DCS) algorithms. With simulations in various distinct mobile network scenarios, we demonstrate the significantly better QoS assurances achieved with the proposed mechanism as compared with best effort and strict priority approaches. With game theoretic concepts, we model DCS applications in an NPDD network as selfish players in a noncooperative game. For such games, we prove for single-hop and multihop NPDD networks the existence of an equilibrium, the feasibility of an equilibrium, and the guaranteed convergence to a feasible equilibrium when one exists.

Index Terms—Game theory, IEEE 802.11, medium access, proportional differentiation, quality-of-service (QoS), scheduling.

I. INTRODUCTION

WIRELESS ad hoc networks are expected to be an integral part of the next-generation communication infrastructure. Such a network provides packet-switched communication among a group of nodes committed to forwarding packets for each other. As a result, a packet may traverse multiple hops toward its destination. With the increasing use of mobile wireless devices such as personal digital assistants, pocket PCs, and smart phones, wireless ad hoc networks are envisioned to be useful in two modes: *peer-to-peer ad hoc networks* that facilitate communication among arbitrary mobile devices without infrastructure support, and *relay-to-infrastructure networks* that extend infrastructure access beyond direct coverage of wireless gateways. In this paper, we address the key challenges in concurrently providing a wide range of end-to-end quality-of-service (QoS) assurances in such networks.

Manuscript received October 14, 2003; revised September 24, 2004. This work is supported in part by the Defense Advanced Research Projects Agency, Air Force Research Laboratory, Air Force Material Command, USAF, under Agreement F30602-00-2-0555, and in part by the U.S. Army under Research Grant DAAD19-01-1-0504 through a Subrecipient Agreement S01-24 from the Pennsylvania State University. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the views of funding agencies.

K.-C. Wang is with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: kwang@clemson.edu).

P. Ramanathan is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706 USA (e-mail: parmash@bece.wisc.edu).

Digital Object Identifier 10.1109/JSAC.2004.842543

The challenges stem from two intrinsic characteristics of a wireless ad hoc network.

- *Node mobility*: Nodes are inherently allowed to enter or leave an ad hoc network at any time. They may also wander around the network during active communication. Node mobility changes the number of nodes, the amount of traffic, and the network topology. These changes result in link breaks and route changes, which inevitably cause packet losses and extended delays [10].
- *Shared medium and decentralized access*: Nodes share the same medium in an ad hoc network. Without centralized infrastructure, nodes access the medium with a decentralized scheduling scheme such as the distributed coordination function (DCF) of IEEE 802.11 [11]. Without global coordination, medium sharing among nodes is potentially unfair [35].

The dynamics due to node mobility and decentralized access of nodes often cause unacceptable variations in QoS of ongoing communications [2], [8], [18].

Several end-to-end QoS assurance techniques exist in the literature [18], [21], [22], [34]. These techniques can be broadly grouped into two camps, integrated services (IntServ)-based and differentiated services (DiffServ)-based. IntServ-based mechanisms aim to assure each flow with its specified QoS along its *specific route* with *per-flow resource reservation* at each node along the route. DiffServ-based mechanisms, on the other hand, do not perform resource reservation and per-flow operations. Instead, a number of service classes are provisioned with certain resources to provide different per hop behaviors (PHBs)—QoS assurance at each node, such as the expedited forwarding (EF) PHB [13] and the assured forwarding (AF) PHB [9]. To provide end-to-end assurances, per domain behaviors (PDBs) ought to be defined [22]. Each application then chooses to be serviced in any of these classes.

Existing IntServ and DiffServ proposals are, however, difficult to implement in a multihop wireless network. Difficulties with IntServ solutions are centered around resource reservation. As the available bandwidth for each node varies with time, it is necessary but hard for a node to estimate its bandwidth for resource reservation, and since the medium is shared, resource reservation must be done with global coordination. Once reservations are made, violations may occur due to bandwidth fluctuations and route changes, as such reservations are pinned to a route [34]. DiffServ assurances also depend on nodes' available resources. The EF PHB assures *low queueing delays at a node* if its service rate is no less than the EF traffic arrival rate. The AF PHB assures a *specified throughput at a node* with packet priority marking and selective queue management. Packets marked

with high priorities are always serviced before those marked with low priorities. If the marking rate does not exceed the service rate, a throughput equal to the marking rate is assured. AF alone, however, does not address end-to-end throughput assurances. Both EF and AF face a difficult resource provisioning problem in a dynamic multihop network.

Numerous recent studies have proposed QoS provision solutions for mobile and ad hoc networks. QoS-aware routing protocols identify *acceptable paths* in an ad hoc network for application flows with end-to-end QoS constraints [19], [20], [28], [38], [41]. To provide IntServ like QoS guarantees, resource reservation schemes are proposed for ad hoc networks [36], [39]. To maintain and restore resource reservation in mobile networks, mobility management and fast handoff schemes are proposed [5], [26]. To provide DiffServ like prioritized service to a small number of real-time traffic classes over best-effort traffics, differentiated scheduling and medium access algorithms are proposed [4], [27], [37], [40]. In [16], a multihop coordinated scheduling scheme is proposed to meet end-to-end delay bounds by servicing more urgent packets prior to others at each node. Nevertheless, these solutions either still face the overheads and uncertainty in resource reservation, provide only limited service differentiation, or are sensitive to end-to-end route changes. In this paper, our objective is to propose an end-to-end QoS assurance framework that is independent of end-to-end route dynamics, free from explicit resource reservation and class provisioning, and able to provide a wide range of assurances.

In [32], we proposed the DiffServ-based service called neighborhood proportional delay differentiation (NPDD) in a multihop wireless network. In the NPDD model, the network supports multiple service classes. The PHB at each node assures that the ratio of average packet delays in two different classes equals a ratio preset by the network service provider. Attractively, this PHB requirement of proportional delays holds at all times independent of a node's dynamic bandwidth and traffic arrival. It also holds consistent *across nodes* in the same network. While the model is not tied to a specific link-layer protocol, we have developed a set of mechanisms that realize NPDD in IEEE 802.11-based ad hoc networks in the presence of node mobility, traffic variations, and network disturbances due to topology-dependent contention (hidden and exposed nodes), fluctuating queueing delays, and packet buffer losses [31]. Based on the model, we showed that an application can effectively choose a class for each packet with dynamic class selection (DCS) algorithms to achieve an end-to-end delay requirement [32] or an end-to-end throughput requirement [33].

While NPDD assures consistent proportional delay differentiation at all times, the *absolute class delays vary* according to the traffic class distribution. As applications independently adapt their classes, queueing delays fluctuate at each node. Essentially, applications compete with each other to enhance their QoS. When an application selects a higher class to enhance its delay or throughput, QoS of all others degrade. *Vice versa*, all applications benefit when an application lowers its class. We model such DCS interactions in a game theoretic framework where applications are considered as independent players in a resource constrained noncooperative game. For single-hop and multihop wireless networks, we analyze the *existence* of an

equilibrium where all applications *settle in their chosen classes*, the *feasibility* of an equilibrium such that all applications *meet their end-to-end QoS requirements*, and the *guaranteed convergence* to a feasible equilibrium when one exists. Earlier in a different setting, noncooperative games have been used to model DiffServ flows serviced by fixed-capacity routers in a wireline network [24].

The remainder of the paper is organized as follows. Section II describes the network model, the NPDD service model, and the end-to-end QoS assurance problem. Section III describes the proposed mechanisms. Section IV summarizes results of the simulation studies. Section V presents the game theoretic analyses. This paper concludes in Section VI.

II. QoS ASSURANCES IN WIRELESS AD HOC NETWORKS

A. The Network Model

We consider a wireless ad hoc network with nodes communicating over a shared medium. Each node has a fixed radio range and exchanges messages only with nodes within this range. Messages for nodes beyond the range are forwarded along multihop routes found with a routing protocol. Along multihop routes, messages are queued, forwarded, and possibly dropped at nodes with a full queue. For medium access, the IEEE 802.1e enhanced DCF (EDCF) with multiple priorities is assumed [29]. The IEEE 802.11 request-to-send (RTS) and clear-to-send (CTS) channel acquisition scheme is also assumed [11]. Mobile nodes may enter or leave the network at any time and may move around the network during active communication. In the case of an ad hoc network connected to the infrastructure, wireless gateways are considered as stationary nodes in the network.

B. NPDD Service Model

The NPDD service model supports N classes relatively ordered in per-hop packet queueing delays at any node. At node k , packets from class i experience smaller delays than class j for all $i < j$, $i, j \in S_{B,k}$, where $S_{B,k}$ is the set of backlogged classes at node k . The spacing between delays is tuned by the network designer with a set of class differentiation parameters. As its name suggests, the model not only holds at each node, it also holds across nodes in a certain neighborhood. Specifically, NPDD maintains delay proportionality at each node and among nodes in the same *contending set*. Two nodes k and q are said to be in the same contending set if there exists a route between them. The NPDD service model is defined as follows. Let $1 = \delta_1 > \delta_2 > \dots > \delta_N > 0$ be N delay differentiation parameters (DDPs) provisioned by the network designer. Let $\bar{d}_i^{(k)}$ denote the average queueing delay of class i packets at node k . The queueing delay is defined as the difference between the time a packet arrives at the node and the time the packet is transmitted again. The NPDD requirement is

$$\frac{\bar{d}_i^{(k)}}{\bar{d}_j^{(q)}} = \frac{\delta_i}{\delta_j} \quad (1)$$

for all classes i and j and between all pairs of nodes k and q such that k and q belong to the same contending set. We define the normalized average queueing delay $\hat{d}_i^{(k)}$ for class i at node k

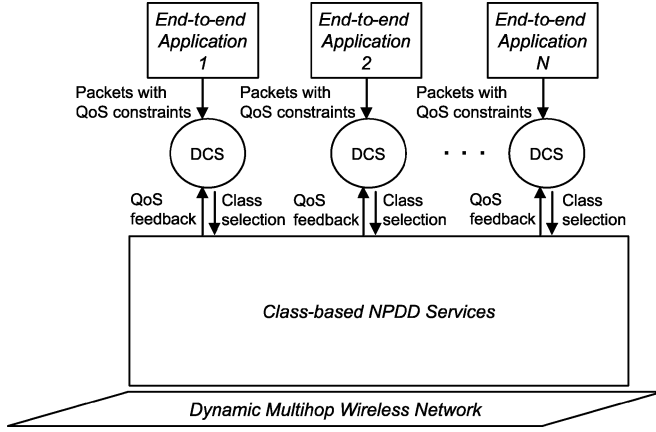


Fig. 1. NPDD-based end-to-end service assurance framework.

as $\hat{d}_i^{(k)} = \bar{d}_i^{(k)} / \delta_i$. If NPDD holds, all backlogged classes at all contending nodes have the same normalized average queuing delay. That is

$$\hat{d}_i^{(k)} = \hat{d}_j^{(q)} \quad \forall i, j \in \{1 \dots N\} \quad (2)$$

for any two nodes k, q in the same contending set.

Considering that a packet's end-to-end delay equals the sum of all per-hop delays along its route, packets of different classes have *proportional end-to-end delays* along the route. For a transmission control protocol (TCP) flow, the end-to-end throughput is inversely proportional to its round-trip time (RTT) [23]. With proportional end-to-end delays, TCP flows have *proportional RTTs* and *proportional throughputs* along any given route.

C. End-to-End QoS Assurance

Mobile nodes host applications with end-to-end flows between arbitrary pairs of nodes. Each flow has its specific QoS requirement. Specifically, an end-to-end delay-sensitive application may request a bounded maximum end-to-end delay, while a TCP-based application may request a bounded minimum end-to-end throughput. The end-to-end QoS assurance problem is formulated as follows. An application f at a node k forms either a delay-sensitive connection or a TCP connection with a remote destination. The application requests either an assured maximum end-to-end delay bound \hat{D}_f for all its packets, or an assured minimum end-to-end throughput \hat{T}_f over its connection duration. The network strives to meet this bound. The network, however, provides no hard guarantees.

III. PROPOSED MECHANISMS

Fig. 1 shows the NPDD-based end-to-end service assurance framework. With NPDD as the common basis, an application selects a class such that its QoS objective is met. For delay-sensitive applications, the delay-based DCS algorithm performs class selection based on end-to-end delay estimation. For applications with throughput requirements, the throughput-based DCS algorithm is proposed with end-to-end TCP throughput estimation. Thus, NPDD concurrently supports delay and throughput oriented applications with no distinction.

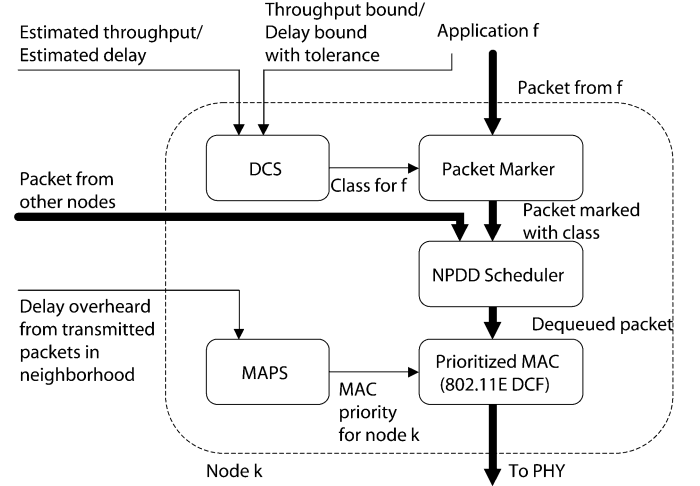


Fig. 2. End-to-end QoS assurance mechanisms based on DCS and NPDD.

Fig. 2 shows the implemented mechanisms at each node. As shown, each application packet is marked with the class determined by a DCS agent. Once marked by the sender, a packet is serviced in the same class along the entire route. At each node, the NPDD service is realized with the NPDD scheduler and the prioritized MAC, whose priority is determined by the medium access priority selection (MAPS) mechanism. The following describes the implementation of DCS, NPDD scheduler, and MAPS.

A. Dynamic Class Selection

Fig. 3 presents the pseudocodes for the delay-based and throughput-based DCS algorithms. Each application f is serviced by a DCS agent based on one of the algorithms. DCS makes periodic class selections every Γ seconds for the following period. At the k th period, four inputs are considered: the current class $c(k\Gamma)$, the current QoS estimate [estimated delay $D_f(k\Gamma)$ or estimated throughput $T_f(k\Gamma)$], the QoS bound (delay bound \hat{D}_f or throughput bound \hat{T}_f), and the corresponding tolerance (Δ_f or Φ_f). While $D_f(k\Gamma)$ is estimated with receiver feedbacks, $T_f(k\Gamma)$ is estimated with TCP ACK packets.

At the end of period k , the class for period $k + 1$ is determined. If the estimated QoS has violated the desired bound for K_I consecutive periods, DCS increases the class by 1. On the other hand, if it is *overly satisfied* by more than the tolerance for K_D consecutive periods, DCS will decrease the class by 1. Otherwise, the class remains the same.

B. NPDD Scheduler

The NPDD scheduler services packets in N classes and realizes proportional average per-hop delays among them locally at each node. The scheduler is work-conserving, and the waiting time priority (WTP) algorithm [7] is adopted. With WTP, each class is serviced with a separate first-in-first-out (FIFO) queue. The head-of-line packet of a class i is assigned a WTP $\tilde{w}_i(t)$ and the scheduler always schedules the highest priority head-of-line packet for transmission.

Every Γ seconds, DCS for application f computes class

$$c((k+1)\Gamma) = \mathcal{C}(c(k\Gamma), D_f(k\Gamma), \hat{D}_f, \Delta_f).$$

k : Index of current period.
 $D_f(k\Gamma)$: Current delay estimate.
 \hat{D}_f : Delay bound of f .
 Δ_f : Delay tolerance of f .
 N : Maximum NPDD class.

```

 $\mathcal{C}(c(k\Gamma), D_f(k\Gamma), \hat{D}_f, \Delta_f)$ 
{
   $c(0) = 0$ ;
  if  $D_f(k\Gamma) > \hat{D}_f$  for  $K_I$  consecutive periods,
     $c((k+1)\Gamma) = \min\{c(k\Gamma) + 1, N\}$ ;
  else if  $D_f(k\Gamma) < \hat{D}_f - \Delta_f$  for  $K_D$  consecutive periods,
     $c((k+1)\Gamma) = \max\{c(k\Gamma) - 1, 1\}$ ;
  else
     $c((k+1)\Gamma) = c(k\Gamma)$ ;
  return  $c((k+1)\Gamma)$ ;
}

```

(a)

Every Γ seconds, DCS for application f computes class

$$c((k+1)\Gamma) = \mathcal{C}(c(k\Gamma), T_f(k\Gamma), \hat{T}_f, \Phi_f).$$

k : Index of current period.
 $T_f(k\Gamma)$: Current throughput estimate.
 \hat{T}_f : Throughput bound of f .
 Φ_f : Throughput tolerance of f .
 N : Maximum NPDD class.

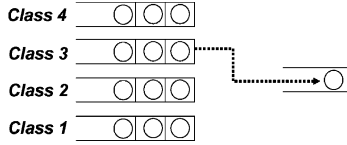
```

 $\mathcal{C}(c(k\Gamma), T_f(k\Gamma), \hat{T}_f, \Phi_f)$ 
{
   $c(0) = 0$ ;
  if  $T_f(k\Gamma) < \hat{T}_f$  for  $K_I$  consecutive periods,
     $c((k+1)\Gamma) = \min\{c(k\Gamma) + 1, N\}$ ;
  else if  $T_f(k\Gamma) > \hat{T}_f + \Phi_f$  for  $K_D$  consecutive periods,
     $c((k+1)\Gamma) = \max\{c(k\Gamma) - 1, 1\}$ ;
  else
     $c((k+1)\Gamma) = c(k\Gamma)$ ;
  return  $c((k+1)\Gamma)$ ;
}

```

(b)

Fig. 3. DCS algorithms. (a) Delay-based DCS. (b) Throughput-based DCS.



Head packet waiting time w_i	Delay differentiation parameter δ_i	Normalized waiting time \bar{w}_i	Class to deque
0.8	1/8	6.4	
2	1/4	8	V
1	1/2	2	
2	1	2	

Fig. 4. Example of the WTP algorithm.

Specifically, let $S_B(t)$ denote the set of classes that have packets waiting for transmission at time t . Let $w_i(t)$ denote the time the class i head-of-line packet has waited in its queue. The WTP $\tilde{w}_i(t)$ at time t is defined as

$$\tilde{w}_i(t) = \frac{w_i(t)}{\delta_i} \quad (3)$$

where δ_i is the DDP of class i . Whenever $S_B(t)$ is nonempty, the scheduler schedules a packet for transmission from class j such that

$$j = \arg \max_{i \in S_B(t)} \tilde{w}_i(t). \quad (4)$$

Fig. 4 shows an example of the WTP algorithm with four classes. In steady state, all packets of a node are transmitted with the same WTP and the NPDD proportionality is realized at this node.

C. Medium Access Priority Selection (MAPS)

NPDD requires packets of the same class to have the same average per-hop delay at nodes in the same contending set. This property, however, can not be realized with proportional scheduling alone. To achieve this property, bandwidth and medium access must be coordinated among nodes based on a fair delay objective. The basic idea of MAPS is to equalize the average normalized waiting time of packets transmitted at each node such that (2) is satisfied. Given multiple priorities in MAC where nodes with higher priorities have higher bandwidth and shorter queueing delays, MAPS adapts a node's priority by comparing the average normalized waiting time of packets transmitted locally and that of all packets transmitted in its radio range. To obtain the latter, waiting time estimates must be exchanged among nodes in each other's radio range.

Specifically, MAPS performs two tasks at a node k at time t . First, it estimates k 's average normalized waiting time $\bar{d}_k(t)$ and its contending set's average normalized waiting time $\bar{d}_{N,k}(t)$. Second, MAPS selects the MAC priority. $\bar{d}_k(t)$ is estimated as a running average of the normalized waiting time of each packet transmitted at node k

$$\bar{d}_k(t) = \alpha \tilde{w}(t) + (1 - \alpha) \bar{d}_k(t) \quad (5)$$

where $\tilde{w}(t)$ is the normalized waiting time of a packet transmitted at time t . To estimate the contending set's average $\bar{d}_{N,k}(t)$, each packet carries two pieces of information: its normalized waiting time $\tilde{w}(t)$ and the sending node q 's estimated average $\bar{d}_{N,q}(t)$. A node overhears all packets transmitted in its radio range. As node k overhears a packet being transmitted by node q , it updates its estimate

$$\bar{d}_{N,k}(t) = \gamma \tilde{w}(t) + \kappa \bar{d}_{N,q}(t) + (1 - \gamma - \kappa) \bar{d}_{N,k}(t) \quad (6)$$

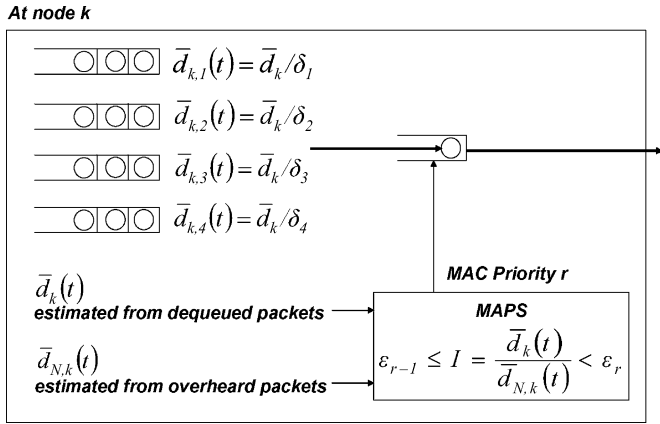


Fig. 5. MAPS mechanism.

where γ and κ are weighting factors and $\gamma + \kappa < 1$. The inclusion of $\bar{d}_{N,q}(t)$ in the estimation is to account for nodes in k 's contending set but out of its radio range. In this case, k may not overhear packets transmitted at nodes outside its radio range, but node q 's estimate potentially accounts for nodes that are neighbors of q but not k . As such, estimates of nodes propagate multiple hops to all nodes in the contending set. With $\bar{d}_k(t)$ and $\bar{d}_{N,k}(t)$, MAPS computes the index

$$I_k(t) = \frac{\bar{d}_k(t)}{\bar{d}_{N,k}(t)}. \quad (7)$$

Given P levels of MAC priorities, P thresholds are defined

$$0 < \epsilon_1 < \epsilon_2 < \dots < \epsilon_P = \infty. \quad (8)$$

MAPS assigns priority r to node k at time t if and only if $\epsilon_{r-1} \leq I_k(t) < \epsilon_r$, where $\epsilon_0 = 0$. Fig. 5 illustrates the MAPS algorithm in choosing a node's MAC priority. In steady state, $I_k(t)$ approaches 1 and NPDD holds consistent across all nodes in the network.

IV. SIMULATION STUDIES

With simulations, the integrated solution is evaluated and compared with two conventional approach, *best effort* and *strict priority*. Denoted as the Baseline scheme, the best effort service uses a FIFO scheduler and the IEEE 802.11 DCF at each node. The baseline performance represents the QoS perceived in existing IEEE 802.11-based ad hoc networks. The strict priority scheme is based on a strict priority scheduler with four classes, the IEEE 802.11e EDCF with three medium access priorities, and the proposed DCS algorithms. As a strict priority scheduler always schedules higher priority packets to be transmitted first, it provides consistently superior QoS to a higher class. The scheme demonstrates the QoS assurances achievable with DCS given consistent class ordering but without the NPDD proportionality. Strict priority classes are statically mapped to the MAC priorities (Class 1, 2 \rightarrow Priority 1, Class 3 \rightarrow Priority 2, Class 4 \rightarrow Priority 3). Finally, the proposed solution is referred to as the DCS-NPDD-MAPS scheme. All simulations are conducted using the network simulator *ns-2* [30] with its CMU mobile node extension. Table I summarizes all simulation parameters. To simulate IEEE 802.11a which is not currently

supported in *ns-2*, modifications are made to its physical-layer attributes as defined in [12]. The modified parameters are summarized in Table II. Dynamic source route [14] is the adopted ad hoc routing protocol.

The simulations evaluate end-to-end throughput and delay assurances for concurrent TCP and user datagram protocol (UDP) applications. We consider three multihop ad hoc network scenarios: a multihop public hotspot, a multihop hotspot with constantly moving nodes, and a peer-to-peer ad hoc network. In all simulations, nodes have a 250 m radio range.

A. Multihop Public Hotspot

In this scenario, we consider a multihop wireless local area network setting where mobile nodes distributed in a 1000 m \times 1000 m square region communicate with a wireless gateway located at the center. Node mobility is modeled with the public hotspot mobility (PHM) model [33], where random node arrivals and departures are modeled with Poisson processes and the number of nodes arriving and departing together is modeled with scenario-dependent distributions. In this paper, we consider individual arrivals and departures with mean arrival and departure rates of 1 node per minute. Around 20 to 30 concurrent flows are present in the network most of the time. An arriving node chooses to stay at a random location with direct or multihop connectivity to the gateway. When nodes leave, other nodes may have their multihop routes broken. Disconnected nodes then move to randomly selected new locations to restore connectivity.

Each node has a TCP flow and a UDP flow between itself and the gateway whenever it is in the network. Flows are randomly selected to be uplink or downlink with a uniform distribution. Each TCP flow randomly selects one of three throughput bounds (30, 60, and 90 kb/s), while each UDP flow randomly selects one of three end-to-end delay bounds (0.1, 0.4, and 0.7 s) with uniform distributions. TCP flows are infinitely backlogged with a maximum window of $W_{\max} = 50$ packets. UDP flows have exponentially distributed on/off intervals with mean duration 128 ms and mean on-time arrival rate 200 kb/s. All packets are 512 bytes in size.

For any TCP flow f with throughput bound \hat{T}_f and session throughput T_f , its throughput assurance is evaluated with its *throughput utility* $\hat{T}_f = \min(1, (T_f/\hat{T}_f))$. Upper bounded by 1, a user has full satisfaction with any throughput above its bound. A UDP flow's delay assurance is evaluated with its *in-time delivery ratio*, i.e., the percentage of packets delivered with end-to-end delays within its bound. Packets dropped en route are considered as packets with infinite delays.

Each simulation lasts for 1000 s. Fig. 6 shows the average throughput utilities grouped by their throughput bounds. Each bar in a group represents the average utility achieved with one scheme. The baseline exhibits the least and the most unpredictable utilities for the diverse TCP applications. The performance degradation is attributed to network topology changes and traffic variations. Apparently, at times of path changes and congestion, applications are unable to maintain desirable throughputs without QoS differentiation. Both the strict priority and the DCS-NPDD-MAPS schemes respond

TABLE I
PARAMETERS OF EVALUATED SERVICE SCHEMES

<i>Scheme</i>	<i>DCS-NPDD-MAPS</i>	<i>Strict Priority</i>	<i>Baseline</i>
TCP throughput tolerance $\Phi(x)$, x : throughput bound	$0.5x$	$0.5x$	<i>N/A</i>
UDP delay tolerance $\Delta(y)$, y : delay bound	$0.5y$	$0.5y$	<i>N/A</i>
DCS period (seconds)	2	2	<i>N/A</i>
DCS sensitivity parameters (K_I, K_D)	(1,1)	(1,1)	<i>N/A</i>
NPDD classes	4	4	<i>N/A</i>
DDP δ_i , $i \in 1, 2, 3, 4$	$[1 \frac{1}{2} \frac{1}{4} \frac{1}{8}]$	<i>N/A</i>	<i>N/A</i>
Per-class maximum queue size (packets)	600	600	2400
MAC priorities	3	3	1
MAC $CW_{\min,i}$, $i \in 1, 2, 3$	[255 127 31]	[255 127 31]	255
MAC $CW_{\max,i}$, $i \in 1, 2, 3$	1023	1023	1023
MAPS ϵ_i , $i \in 1, 2, 3$	[0.4 0.6 ∞]	<i>N/A</i>	<i>N/A</i>
MAPS $d_{N,k}$ average weights (γ, κ)	(0.1, 0.1)	<i>N/A</i>	<i>N/A</i>
MAPS d_k moving average weight α	0.9	<i>N/A</i>	<i>N/A</i>
802.11 modes	802.11e over 802.11a	802.11e over 802.11a	802.11a

TABLE II
IEEE 802.11a PARAMETERS UPDATED IN *ns-2*

<i>Scheme</i>	<i>Values</i>
aSlotTime	$9\mu s$
aCCATime	$4\mu s$
aRxFxTurnaroundTime	$2\mu s$
aSIFSTime	$16\mu s$
aPreambleLength	$20\mu s$ (120 bits @ 6Mbps)
aPLCPHeaderLength	$4\mu s$ (24 bits @ 6Mbps)
aPLCPDataRate	6Mbps
BasicRate	6Mbps
DataRate	54Mbps

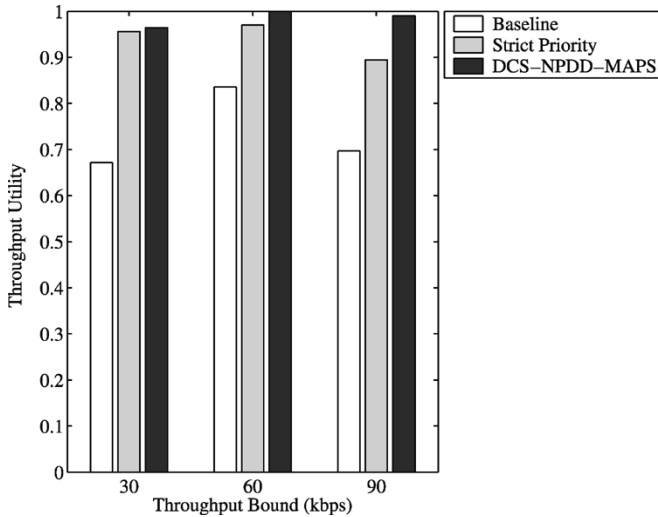


Fig. 6. Average throughput utilities in a PHM multihop hotspot.

to network changes by means of class adaptation. The strict priority does present consistently better utilities than the baseline, while DCS-NPDD-MAPS provides the highest utility for applications with all throughput requirements.

As is mentioned earlier, packets lost en route are considered as infinitely delayed. In a highly congested network, packet losses can severely degrade the overall delay assurance. Fig. 7(a) shows the overall packet delivery ratio, which is the percentage of *all packets delivered* from senders to receivers without worrying about the delay bounds. Fig. 7(b), on the other hand, shows

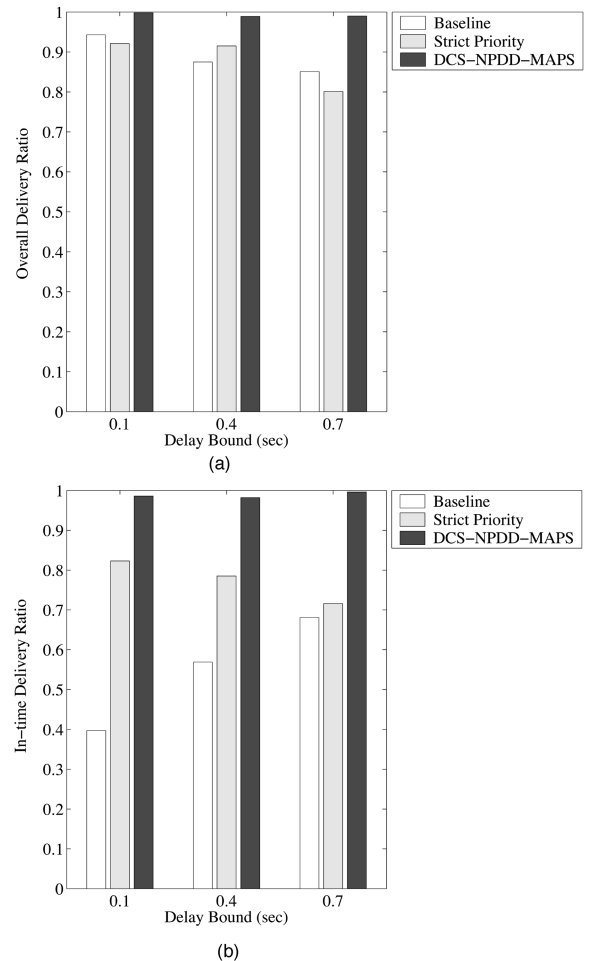


Fig. 7. Average delivery ratios in a PHM multihop WLAN. (a) Packets received over packets sent. (b) Packets received in time over packets sent.

the percentage of *packets delivered in time*. As is seen, DCS-NPDD-MAPS provides a higher delivery ratio with and without considering the delay constraints. While the baseline scheme shows a clear degradation trend in in-time delivery ratios with tighter delay bounds, the strict priority and DCS-NPDD-MAPS schemes provide more uniform performance for applications with diverse delay requirements.

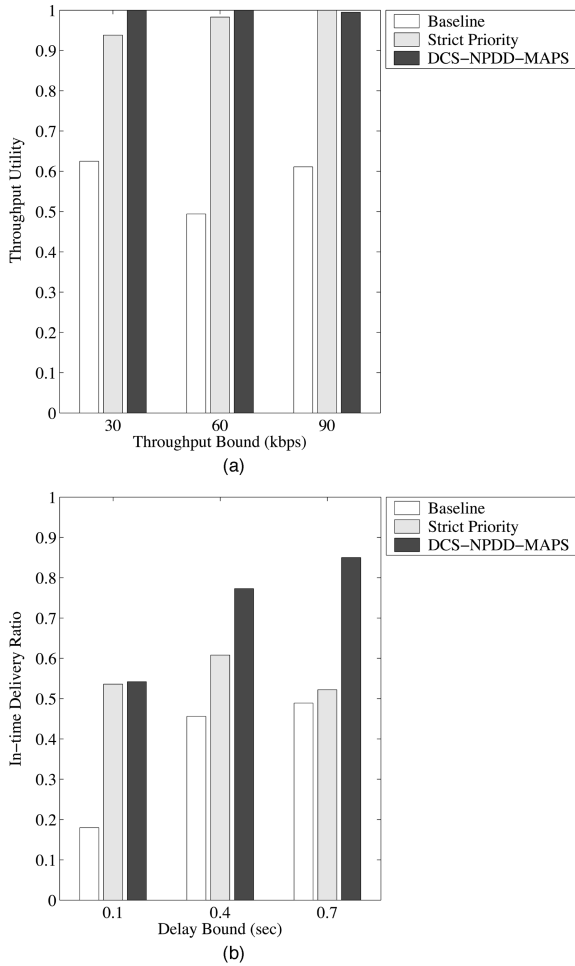


Fig. 8. Assurances in a multihop hotspot with constantly moving nodes. (a) Average throughput utilities. (b) Average in-time delivery ratios.

B. Multihop Hotspots With Moving Nodes

In this scenario, 20 nodes are placed in two circles around the gateway. Nodes in the outer circle are beyond the gateway's radio coverage. The constant movement is modeled with the inner nodes moving counterclockwise around the AP at 5 m/s. Such artificial node movement is rather regular, but it creates frequent topology and route changes for our evaluation purpose. Each simulation lasts for 1000 s.

Fig. 8(a) shows the achieved throughput utilities. In this highly mobile scenario, the baseline remains unsatisfactory, while the strict priority and DCS-NPDD-MAPS schemes achieve substantially high throughput utilities. The in-time delivery ratios are, however, obviously impaired by the constant node mobility. As seen in Fig. 8(b), DCS-NPDD-MAPS still poses significantly higher in-time delivery ratios than the other two for applications with 0.4- and 0.7-s delay bounds. For applications with a 0.1-s delay bound, strict priority and DCS-NPDD-MAPS are similar in performance. The frequent changes in network topology force a large number of packets to be held at nodes during periods of route breaks and route repairs. Such overheads are inevitable and cannot be fully compensated with either scheduling scheme. Moreover, route repairs constitute substantial traffic that induces further network congestion. It is observed that long nonoptimal routes are

exploited by DSR quite often in this highly mobile network. At times when route changes aggregate a substantial amount of bursty traffic to a node, long queues and bursty drops are seen as well.

C. Peer-to-Peer Ad Hoc Network

Peer-to-peer ad hoc networks assume end-to-end communications between arbitrary node pairs. On the one hand, peer-to-peer networks and hotspot networks are identical in the sense that all nodes service incoming traffic with exactly the same mechanisms. On the other hand, the traffic distribution and the expected end-to-end QoS may differ substantially among these networks. For example, in multihop hotspots, end-to-end flows gather at the gateway and create a higher traffic density at nodes closer to the gateway. In peer-to-peer networks, traffic density can be higher when flows happen to traverse routes in the same vicinity, or it can also be lower when flows are uniformly distributed in the network. We consider a mobile ad hoc network where 36 nodes are initially placed in a 6×6 grid over a $625 \text{ m} \times 625 \text{ m}$ square region, from where each node randomly moves according to the random way point model with a maximum speed of 2 m/s [15].

In this network, we simulate 30 UDP flows and 10 TCP flows between random node pairs. The source and destination nodes are chosen from the two sides of the region such that most flows traverse across the entire network. The flows are randomly set to be rightward or leftward with a uniform distribution. The flow parameters are the same as previous scenarios, except each TCP flow randomly selects one out of three possible throughput bounds (30, 60, and 90 kb/s) and each UDP flow randomly selects one out of seven possible end-to-end delay bounds (2–8 s) with uniform distributions. Each simulation lasts for 500 s.

Fig. 9 summarizes the throughput utilities of TCP flows and in-time delivery ratios of UDP flows. Similarly, DCS-NPDD-MAPS has been able to concurrently meet the diverse delay and throughput requirements with the highest satisfaction. While strict priority achieves throughput utilities similar to that of DCS-NPDD-MAPS, its UDP applications suffer from serious delay violations. The baseline assurance is certainly unacceptable for either UDP or TCP applications. While the relatively higher node density, more complex traffic patterns, and longer routes result in vastly different performances as compared with those in a hotspot, the superior performances achieved with DCS-NPDD-MAPS are significant. As depicted by the DCS algorithms, applications not meeting their QoS bounds eventually select the highest class. It is observed in simulations that, at times of congestion, a large number of applications indeed select the highest class simultaneously. Nevertheless, as congestion resolves, applications always return to lower classes and still meet their bounds. In the long run, the average performances presented do suggest the network is stable most of the time and achieves satisfactory QoS assurances.

V. DCS: A GAME THEORETIC PERSPECTIVE

Game theoretic concepts are used in this paper to model the competitive nature of multiple DCS applications in an NPDD

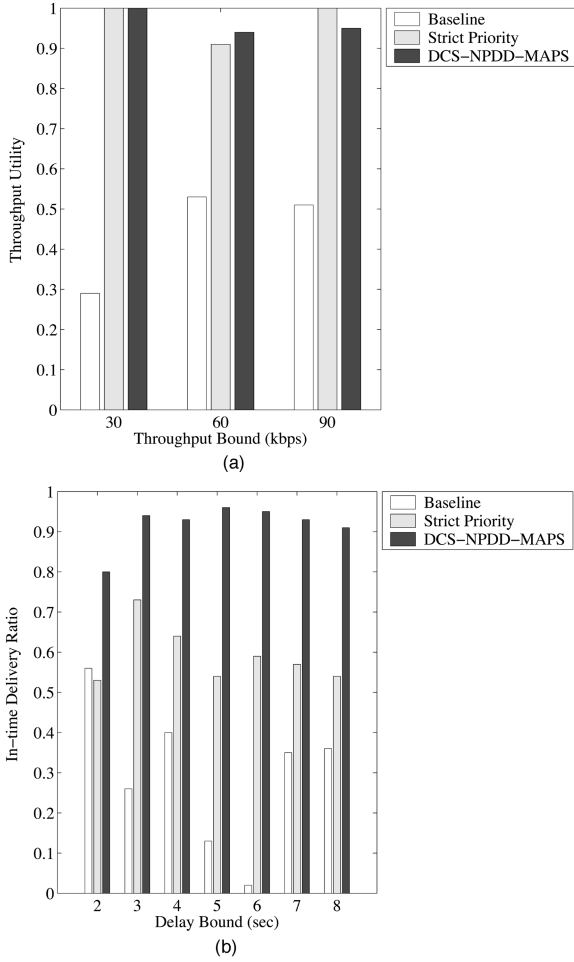


Fig. 9. Assurances in an ad hoc network. (a) Average throughput utilities. (b) Average in-time delivery ratios.

network. Specifically, we consider the *multiagent noncooperative game* of DCS applications with *end-to-end delay requirements* over a generalized *continuous NPDD service model*. In the following, we define the continuous NPDD service model, the noncooperative game, and prove the *existence of an equilibrium*, the *feasibility of an equilibrium*, and the *guaranteed convergence to a feasible equilibrium when one exists*. The proven theorems apply to single-hop NPDD networks, as well as multihop NPDD networks under the work-conserving assumption for both networks.

A. Continuous NPDD Service Model

The continuous NPDD service model has an infinite number of classes denoted as real numbers on the closed line segment $S_C = [0, K]$. The continuous NPDD service model is then defined as follows.

Let $\delta(c)$ be the DDP of a class $c \in S_C$. Let $d(c)$ denote the average queueing delay of class c packets at any node. $d(c)$ is defined if and only if there has been at least one packet serviced in class c . In the same contending set, all nodes have the same average queueing delay for the same class as long as they have serviced packet(s) in that class. Thus, the NPDD requirement

states that for all classes c and l such that the set of packets serviced in them is nonempty

$$\frac{d(c)}{d(l)} = \frac{\delta(c)}{\delta(l)}. \quad (9)$$

For any class $c \in S_C$, its DDP is determined with the following *continuous* and *monotonically decreasing* function

$$\delta(c) = P^{\frac{c}{K}} \quad \forall c \in S_C \quad (10)$$

where P is a design parameter that controls the ratio between the delay of class K , $d(K)$, and the delay of class 0, $d(0)$, such that at any given time when there are packets in both classes

$$d(K) = P \cdot d(0), \quad 0 < P < 1. \quad (11)$$

Note that the choice of the DDP function is not unique. Nevertheless, any chosen function must be continuous and monotonically decreasing to guarantee existence of an equilibrium of the game.

According to [17], the *conservation law* states that for any work-conserving scheduler servicing N traffic classes

$$\sum_{c=1}^N \lambda(c)d(c) = \lambda d_{ag} = q_{ag} \quad (12)$$

where $\lambda(c)$ is the arrival rate in class c . d_{ag} and q_{ag} are the average delay and the average backlog, both of which are constant, if the same aggregate arrival is applied to a FIFO scheduler with the same capacity. Subjected to the continuous NPDD constraint, delay of class c is

$$d(c) = \frac{\delta(c)q_{ag}}{\int_{\zeta=0}^K \delta(\zeta)\lambda(\zeta)d\zeta}. \quad (13)$$

In further analyses, it is essential that the work-conserving assumption applies to single-hop and multihop wireless networks. In a single-hop wireless network where all nodes are within each other's carrier sensing range, we assume all nodes share the same bandwidth with stationary random backoff and collision overheads. In a *large* multihop network spanning an area much larger than a node's radio range with *active and uniformly distributed nodes*, all nodes share the same bandwidth with nodes in each other's radio range and perceive a similar degree of contention. Note that, this assumption inevitably precludes situations where networks are nonuniform in node density, traffic distribution, and topology-dependent contentions.

For a simpler presentation, analysis of a single-hop network is first presented in the following. The same analysis procedure applies to a multihop wireless network [31], whose conclusion is summarized at the end of this section. Consider M applications in a single-hop network. Each application flow f has a class c_f , traffic rate r_f , and single-hop delay D_f . From the class delay expression in (13), flow f 's delay is found to be

$$D_f = d(c_f) = \frac{\delta(c_f)q_{ag}}{\sum_{g=1}^M \delta(c_g)r_g}. \quad (14)$$

Given a set of flows and their class selections, a class increase in one flow results in increased delays for all other flows. *Vice versa*, a class decrease in one flow results in reduced delays for all other flows.

B. The Game

1) *The Ideal Game*: Given constrained network resources, each application competes for a better delay by increasing its class, while such class changes may degrade delays of others and trigger subsequent class changes of others. In an economic setting, we model each application as a selfish player trying to maximize its own utility. Consider M DCS applications. Each application f has a traffic rate r_f and an end-to-end delay bound \hat{D}_f . Let c_f denote f 's chosen class, D_f denote its end-to-end delay, and \bar{c}_f denote the vector containing the classes of all applications other than f . At any given time, f selects a class c_f that maximizes its utility subject to \bar{c}_f

$$c_f = \arg \max_{c \in S_C} U_f(c | \bar{c}_f) \quad (15)$$

where U_f is the utility function of f . That is, given that *each application knows the classes selected by all others*, the application selects a class that unilaterally maximizes its utility.

2) *Utility Functions*: To faithfully reflect a DCS application's desire to meet its delay bound with the least possible class, we define a utility function that depends on both the delay and class of an application conditioned on its current delay. For an application f , its utility function is defined as

$$U_f(c_f, \tilde{D}_f) = \begin{cases} U_{D,f}(\tilde{D}_f), & \text{if } \tilde{D}_f > 1 \\ U_{C,f}(c_f), & \text{if } \tilde{D}_f \leq 1 \end{cases} \quad (16)$$

where $U_{D,f}$ is its *delay utility function*, $U_{C,f}$ is its *class utility function*, and $\tilde{D}_f = D_f / \hat{D}_f$ is its *normalized end-to-end delay* with respect to its delay bound. Equation (16) correctly models the envisioned application preference if the following properties are satisfied.

- For $\tilde{D}_f > 1$, $U_{D,f}(\tilde{D}_f)$ decreases monotonically as \tilde{D}_f increases. For $0 \leq \tilde{D}_f \leq 1$, $U_{D,f}(\tilde{D}_f)$ remains flat.
- $U_{C,f}(c_f)$ decreases monotonically as class increases.

3) *An Approximate DCS Strategy*: From (14), it is seen that flow f 's delay depends on arrival rates and class selection of all other flows. Without this complete knowledge, it is difficult to determine f 's delay in some class c before it is actually in class c . As a result, it is difficult to precisely estimate the optimal class and the corresponding optimal delay of a flow in the ideal game of (15).

At the same time, observe that $\sum_{g=1}^M \delta(c_g) r_g \gg \delta(c_f) r_f$ when the aggregate flow rate is much higher than that of any individual flow f . Assume that in a short time frame, most flows incur none or minor class changes only, $\sum_{g=1}^M \delta(c_g) r_g$ remains relatively constant regardless of the individual class changes. At time t , given f 's current class $c_f(t)$, current delay $D_f(t)$, and chosen new class $c_f(t + \Delta t)$, its delay for the next Δt period is

$$D_f(t + \Delta t) \approx \frac{\delta(c_f(t + \Delta t)) q_{ag}}{\sum_{g=1}^M \delta(c_g(t)) r_g} = \frac{\delta(c_f(t + \Delta t))}{\delta(c_f(t))} D_f(t). \quad (17)$$

A DCS strategy that approximates the ideal game is thereby defined as follows. Periodically, DCS chooses a class at time t for the next time period $[t, t + \Delta t]$

$$c_f(t + \Delta t) = \max \{0, \min \{K, c_{f,est}(t)\}\} \quad (18)$$

where $c_{f,est}(t)$ is the *estimated optimal class* such that

$$\frac{\delta(c_{f,est}(t))}{\delta(c_f(t))} = P^{\frac{c_{f,est}(t) - c_f(t)}{K}} = \frac{\hat{D}_f}{D_f(t)} \quad (19)$$

i.e., $c_{f,est}(t) = c_f(t) + (\ln \hat{D}_f - \ln D_f(t)) K / \ln P$.

It is possible that $c_{f,est}(t)$ may fall beyond the range of supported classes. When this happens, the selected class is clamped at either the highest class K or the lowest class 0. Defined as such, the strategy has the following properties.

- *DCS Property 1*: If an unsatisfied application chooses to remain in a class, the class must be the highest class available.
- *DCS Property 2*: If an overly satisfied application chooses to remain in a class, the class must be the least class available.
- *DCS Property 3*: If a satisfied application chooses to remain in a class other than the least class, its delay bound must have been met exactly.
- *DCS Property 4*: In a single-hop work-conserving network, two applications f and g with one-hop delay requirements \hat{D}_f and \hat{D}_g always have $(\delta(c_{f,est}(t)) / \delta(c_{g,est}(t))) = (\hat{D}_f / \hat{D}_g)$. From (9), (14), and (19),

$$\frac{\delta(c_{f,est}(t))}{\delta(c_{g,est}(t))} = \frac{\hat{D}_f}{\hat{D}_g} \cdot \frac{\frac{\delta(c_f(t))}{d(c_f(t))}}{\frac{\delta(c_g(t))}{d(c_g(t))}} = \frac{\hat{D}_f}{\hat{D}_g}. \quad (20)$$

C. DCS Equilibrium

1) *Existence of Equilibrium*: With the ideal game of (15), the existence of an equilibrium is proved following the equilibrium existence theorem of Debreu's [6]. As long as all applications select their classes from a continuous set S_C and receive a utility satisfying properties given in Section V-B2, there exists an equilibrium for the game [31].

2) *Feasibility of Equilibrium*: While there is always an equilibrium, all applications may not always meet their delay bounds in any equilibrium even at the highest class K . An equilibrium is *feasible* if and only if all applications meet their delay bounds; otherwise, it is *infeasible*. We define a class selection vector (CSV) for M applications as $\mathbf{C} = [c_1, \dots, c_M]$.

Theorem 5.1: If there is a feasible equilibrium for a set of DCS applications serviced by nodes in a single-hop work-conserving NPDD wireless network, then there are no infeasible equilibria for the game.

Proof: Since there is a feasible equilibrium, there is at least one CSV \mathbf{C} such that $D_f \leq \hat{D}_f, \forall f \in \{1, \dots, M\}$. Suppose there exists another equilibrium \mathbf{C}' , where a subset of applications $S_{\text{infeasible}}$ do not meet their delay requirements such that

$$D'_f > \hat{D}_f, \quad \forall f \in S_{\text{infeasible}} \quad (21)$$

while all remaining flows are satisfied with both \mathbf{C} and \mathbf{C}' . Let S_{feasible} denote the set of satisfied flows in \mathbf{C}' . Let c_{max} be the highest class used in \mathbf{C} and π be the corresponding DDP. From (21), DCS Property 1, and the conservation law

$$d(K)' > d(c_{\text{max}}) = \pi \cdot d(0). \quad (22)$$

On the other hand, the conservation law also suggests that $S_{\text{feasible}} \neq \emptyset$ and $\sum_{f \in S_{\text{feasible}}} r_f D'_f < \sum_{f \in S_{\text{feasible}}} r_f D_f$. Hence, one or more flows in S_{feasible} must have had lower delays with C' than with C . From DCS Property 2, $d(0)' < d(0)$, while from (11) and (22), $P \cdot d(0)' = d(K)' > d(c_{\text{max}}) = \pi \cdot d(0) \geq P \cdot d(0)$ and, hence, $d(0)' > d(0)$. The theorem follows by contradiction. ■

3) Convergence to a Feasible Equilibrium:

Theorem 5.2: If there is a feasible equilibrium for a set of DCS applications serviced by nodes in a single-hop work-conserving NPDD wireless network, the game always converges to a feasible equilibrium using the approximate DCS strategy of (18), where all applications change classes concurrently in a synchronized fashion.

Proof: Starting at time 0, each application f determines its class for the next Δt time period according to (18). Without loss of generality, we assume $\Delta t = 1$. At time t , if $c_{f,\text{est}}(t) \in [0, K]$, $c_f(t+1) = c_{f,\text{est}}(t)$. Otherwise, $c_f(t+1)$ is clamped at class 0 or K . Thus, we partition the applications into three sets: $S_{\ominus}(t)$, the set of applications with $c_{f,\text{est}}(t) < 0$, $S_{\oplus}(t)$, the set of applications with $c_{f,\text{est}}(t) > K$, and $S_{\circ}(t)$, the set of applications with $0 \leq c_{f,\text{est}}(t) \leq K$. After i iterations of class changes, the delay of f can be found from (14), shown in (23) at the bottom of the page. By the conservation law, a feasible equilibrium, i.e. $(D_g \leq \hat{D}_g, \forall g \in \{1, \dots, M\})$, has

$$q_{\text{ag}} = \sum_{g=1}^M r_g D_g \leq \sum_{g=1}^M r_g \hat{D}_g. \quad (24)$$

Case 1) $S_{\ominus}(0) = S_{\oplus}(0) = \emptyset$, from (17) and (23), $f \in S_{\circ}(i-1)$ has

$$D_f(i) = \frac{\hat{D}_f \sum_{g=1}^M \delta(c_g(i-1)) r_g}{\sum_{g=1}^M \frac{\hat{D}_g}{D_g(i-1)} \delta(c_g(i-1)) r_g}. \quad (25)$$

Substituting $D_g(i-1)$ with (14)

$$D_f(i) = \hat{D}_f \cdot \frac{q_{\text{ag}}}{\sum_{g=1}^M r_g \hat{D}_g} = \beta \cdot \hat{D}_f \quad (26)$$

where $\beta = q_{\text{ag}} / \sum_{g=1}^M r_g \hat{D}_g$ and $0 < \beta \leq 1$. If $\beta = 1$, all applications meet their bounds exactly in one iteration and the game converges. Otherwise, all applications continue reducing their classes until some application is clamped at class 0.

To continue, we first show that $S_{\ominus}(i-1) \neq \emptyset$ implies $S_{\oplus}(i-1) = \emptyset$ and *vice versa*. If there exist $f \in S_{\ominus}(i-1)$ and $g \in S_{\oplus}(i-1)$ at the same time, then $\delta(c_{f,\text{est}}(i-1)) > \delta(0) = 1$ and $\delta(c_{g,\text{est}}(i-1)) < \delta(K) = P$. From DCS Property 4, $\delta(c_{f,\text{est}}(i-1)) / \delta(c_{g,\text{est}}(i-1)) = (\hat{D}_f / \hat{D}_g) > (1/P)$. Given that there exists a feasible equilibrium, there cannot be applications with delay bounds ratioed larger

than the maximum NPDD spacing. By contradiction, $S_{\ominus}(i-1)$ and $S_{\oplus}(i-1)$ can never be nonempty at the same time.

Case 2) $S_{\ominus}(0) \neq \emptyset$ and $S_{\oplus}(0) = \emptyset$, $f \in S_{\circ}(i-1)$ has

$$D_f(i) = \frac{\gamma \delta(c_f(i-1)) q_{\text{ag}}}{\sum_{g \in S_{\circ}(i-1)} \gamma \delta(c_g(i-1)) r_g + \sum_{g \in S_{\ominus}(i-1)} \delta(0) r_g}$$

where $\gamma = \hat{D}_f / D_f(i)$. If $\gamma > 1$, f is overly satisfied and

$$\hat{D}_f > D_f(i) \geq \frac{\delta(c_f(i-1)) q_{\text{ag}}}{\sum_{g \in S_{\circ}(i-1)} \delta(c_g(i-1)) r_g + \sum_{g \in S_{\ominus}(i-1)} \delta(0) r_g} = D_f(i-1). \quad (27)$$

Similarly, if $\gamma < 1$

$$\hat{D}_f \leq D_f(i) \leq D_f(i-1). \quad (28)$$

If $\gamma = 1$, $D_f(i) = \hat{D}_f$. In all cases, $\{D_f(i)\}$ forms a bounded monotonic sequence that is guaranteed to converge [25]. At convergence (in i iterations), applications in $S_{\circ}(i)$ meet their delay bounds exactly, applications in $S_{\ominus}(i)$ are overly satisfied at class 0, and $S_{\oplus}(i) = \emptyset \forall i \geq 0$, i.e., all applications converge in a feasible equilibrium.

Case 3) $S_{\oplus}(0) \neq \emptyset$ and $S_{\ominus}(0) = \emptyset$, $f \in S_{\circ}(i-1)$ has

$$D_f(i) = \frac{\delta(c_{f,\text{est}}(i-1)) q_{\text{ag}}}{\sum_{g \in S_{\circ}(i-1)} \delta(c_{g,\text{est}}(i-1)) r_g + \sum_{g \in S_{\oplus}(i-1)} \delta(K) r_g} < \beta \cdot \hat{D}_f$$

i.e., applications in $S_{\circ}(i-1)$ are always overly satisfied and

$$D_f(i) = \frac{\gamma \delta(c_f(i-1)) q_{\text{ag}}}{\sum_{g \in S_{\circ}(i-1)} \gamma \delta(c_g(i-1)) r_g + \sum_{g \in S_{\oplus}(i-1)} \delta(K) r_g}.$$

Since f is overly satisfied, $\gamma > 1$ and

$$\hat{D}_f > D_f(i) \geq \frac{\delta(c_f(i-1)) q_{\text{ag}}}{\sum_{g \in S_{\circ}(i-1)} \delta(c_g(i-1)) r_g + \sum_{g \in S_{\oplus}(i-1)} \delta(0) r_g} = D_f(i-1). \quad (29)$$

$\{D_f(i)\}$ forms a bounded monotonic sequence that is guaranteed to converge [25]. By Theorem 5.1, at convergence (in i iterations), applications in $S_{\circ}(i)$ meet their delay bounds exactly, applications in $S_{\ominus}(i)$ are overly satisfied at class 0, and $S_{\oplus}(i) = \emptyset$, i.e., all applications converge in a feasible equilibrium. The proof thereby completes. ■

For work-conserving multihop wireless networks with a given set of multihop flows along a certain set of multihop routes, each node services a stationary aggregate arrival. In a network of Q nodes and M multihop flows with end-to-end delay bounds \hat{D}_f , $f \in \{1, \dots, M\}$, let τ_{fq} be the per-hop

$$D_f(i) = \delta(c_f(i)) \cdot \frac{q_{\text{ag}}}{\left(\sum_{g \in S_{\ominus}(i-1)} \delta(0) r_g + \sum_{g \in S_{\oplus}(i-1)} \delta(K) r_g + \sum_{g \in S_{\circ}(i-1)} \delta(c_{g,\text{est}}(i-1)) r_g \right)} \quad (23)$$

queueing delay of f at node $q \in \{1, \dots, Q\}$ and $\tau_{fq} = 0$ if f is not routed through node q . Thus, the end-to-end delay of f is $D_f = \sum_{q=1}^Q \tau_{fq}$. Let S_q be the set of applications serviced by node q , the aggregate arrival at node q is $\Lambda_q = \sum_{f \in S_q} r_f$. We define the *average network queueing delay*

$$\mathbb{D} = \frac{\sum_{f=1}^M \left(\sum_{q=1}^Q \tau_{fq} \cdot r_f \right)}{\sum_{q=1}^Q \Lambda_q} = \frac{\sum_{f=1}^M D_f r_f}{\sum_{q=1}^Q \Lambda_q} \quad (30)$$

by averaging per-hop queueing delays of all applications at all nodes. The following conservation relationship, thus, holds:

$$\sum_{q=1}^Q \Lambda_q \cdot \mathbb{D} = \sum_{f=1}^M D_f r_f = Q_{ag} \quad (31)$$

where Q_{ag} is a constant regardless of traffic and class distribution among nodes in the network. With the conservation relationship, the same feasibility and convergence proofs are proved with similar procedures [31].

As mentioned, the work-conserving assumption applies only to large and uniform multihop networks. Nevertheless, the results remain informative in understanding the steady-state behavior of DCS in a uniform NPDD network domain. Based on the conclusions, potential admission control strategies can be devised. Given that the aggregate backlog can be estimated for simple work-conserving queueing models [1], [3], the necessary condition for the existence of a feasible equilibrium, i.e., (24), can serve the admission control test. A flow is admitted on its arrival if the condition is met and there exists a feasible NPDD class assignment according to the network's current traffic profile. Best-effort flows can be admitted with the same procedure and be served with the least available class. If admission control is not exercised, however, it is understood that at times of congestion, all applications will select the same highest class and perceive the same QoS as others as in a best-effort network. The proven convergence further suggests that applications will always regain satisfactory QoS as congestion conditions are resolved.

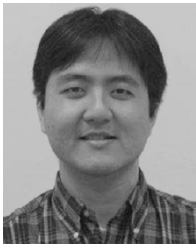
VI. CONCLUSION

This paper addresses the challenges of concurrently providing end-to-end throughput and delay assurances in a wireless ad hoc network. The proposed solution is based on class selection among multiple service classes in the NPDD service model. In a highly mobile multihop network, the model provides a set of classes with per-hop delays proportional to the predefined ratios and this proportionality holds across all nodes independent of network dynamics. With simulations, the proposed class selection mechanism is shown to effectively achieve diverse end-to-end throughput and delay assurances in ad hoc networks with random arrivals of mobile nodes. Using noncooperative game theory, we prove that: 1) there always exists an equilibrium for a DCS game in NPDD networks; 2) when there exists a feasible equilibrium, there exist no infeasible equilibria; and 3), when there exists a feasible equilibrium, a DCS game is guaranteed to converge to a feasible equilibrium.

REFERENCES

- [1] O. Allen, *Probability, Statistics, and Queueing Theory With Computer Science Applications*, 2nd ed. New York: Academic, 1990.
- [2] S. Bansal, R. Gupta, R. Shorey, I. Ali, A. Razdan, and A. Misra, "Energy efficiency and throughput for TCP traffic in multi-hop wireless networks," in *Proc. IEEE INFOCOM*, vol. 1, 2002, pp. 210–219.
- [3] G. Bloch, S. Greiner, H. Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains*. New York: Wiley, 1999.
- [4] H.-L. Chao, J.-C. Kuo, and W. Liao, "Fair scheduling with QoS support in ad hoc networks," in *Proc. IEEE LCN*, Nov. 2002, pp. 502–507.
- [5] Y. Cheng and W. Zhuang, "DiffServ resource allocation for fast handoff in wireless mobile Internet," *IEEE Commun. Mag.*, vol. 40, no. 5, pp. 130–136, May 2002.
- [6] G. Debreu, "A social equilibrium existence theorem," *Proc. Nat. Acad. Sci.*, vol. 38, no. 10, pp. 886–893, Oct. 1952.
- [7] C. Dovrolis, D. Stiliadis, and P. Ramanathan, "Proportional differentiated services: delay differentiation and packet scheduling," *IEEE/ACM Trans. Netw.*, vol. 10, no. 1, pp. 12–26, Feb. 2002.
- [8] M. Gerla, K. Tang, and R. Bagrodia, "TCP performance in wireless multi-hop networks," in *Proc. IEEE Workshop Mobile Comput. Syst. Appl.*, 1999, pp. 41–50.
- [9] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," IETF, RFC 2597, Jun. 1999.
- [10] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Proc. ACM MOBICOM*, August 1999, pp. 219–230.
- [11] *Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11, 1999.
- [12] *Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications: High-Speed Physical Layer in the 5 GHz Band*, IEEE Standard 802.11a, 1999.
- [13] V. Jacobsen, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB (Per-Hop Behavior)," IETF, RFC 2598, Feb. 1999.
- [14] D. B. Johnson, D. A. Maltz, and J. Broch, "The dynamic source routing protocol for multihop wireless ad hoc networks," *Ad Hoc Netw.*, pp. 139–172, 2001.
- [15] D. Johnson *et al.*, "The dynamic source routing protocol for mobile ad hoc networks," Internet Draft, Feb. 2002.
- [16] V. Kanodia, C. Li, A. Sabharwal, B. Sadeghi, and E. Knightly, "Distributed multi-hop scheduling and medium access with delay and throughput constraints," in *Proc. ACM MOBICOM*, 2001, pp. 200–209.
- [17] L. Kleinrock, *Queueing Systems, Volume II*. New York: Wiley, 1976.
- [18] S.-B. Lee, G.-S. Ahn, and A. T. Campbell, "Improving UDP and TCP performance in mobile ad hoc networks with INSIGNIA," *IEEE Commun. Mag.*, vol. 39, no. 6, pp. 156–165, Jun. 2001.
- [19] C. Lin, "On-demand QoS routing in multihop mobile networks," in *Proc. IEEE INFOCOM*, vol. 3, Apr. 2001, pp. 1735–1744.
- [20] C. R. Lin and J.-S. Liu, "QoS routing in ad hoc wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1426–1438, Aug. 1999.
- [21] H. Luo, S. Lu, and V. Bharghavan, "A new model for packet scheduling in multihop wireless networks," in *Proc. ACM MOBICOM*, Aug. 2000, pp. 76–86.
- [22] K. Nichols and B. Carpenter, "Definition of differentiated services per domain behaviors and rules for their specification," IETF, RFC 3086, Apr. 2001.
- [23] J. Padhye, V. Firoiu, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *Proc. ACM SIGCOMM*, vol. 28, Oct. 1998, pp. 303–314.
- [24] H. Ren and K. Park, "Toward a theory of differentiated services," in *Proc. IEEE/IFIP Int. Workshop Qual. Serv. 2000*, 2000, pp. 211–220.
- [25] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York: McGraw-Hill, 1976.
- [26] P. Sethi and G. Barua, "CRESQ: providing QoS and security in ad hoc networks," in *Proc. Euromicro Conf. Parallel, Distrib. Network-Based Process.*, Feb. 2003, pp. 544–550.
- [27] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 8, pp. 1353–1368, Aug. 1999.
- [28] H. Sun and H. D. Hughes, "Adaptive QoS routing based on prediction of local performance in ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, vol. 2, Mar. 2003, pp. 1191–1195.
- [29] The IEEE P802.11 Task Group E. The IEEE 802.11e. [Online]. Available: <http://grouper.ieee.org/groups/802/11/>
- [30] The VINT Project. (2000, Nov.) The Network Simulator—ns-2 v.2.1b7a. [Online]. Available: <http://www.isi.edu/nsnam/ns/>
- [31] K.-C. Wang, "Quality of service assurances in multihop wireless networks," Ph.D. dissertation, Univ. Wisconsin, Madison, WI, 2003.
- [32] K.-C. Wang and P. Ramanathan, "End-to-end delay assurances in multihop wireless local area networks," in *Proc. IEEE GLOBECOM*, 2003, pp. 2962–2966.

- [33] ———, "End-to-end throughput and delay assurances in multihop wireless hotspots," in *Proc. ACM Int. Workshop Wireless Mobile Appl. Serv. WLAN Hotspots*, 2003, pp. 93–102.
- [34] P. P. White, "RSVP and integrated services in the internet: a tutorial," *IEEE Commun. Mag.*, vol. 35, no. 5, pp. 100–106, May 1997.
- [35] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc network," *IEEE Commun. Mag.*, vol. 39, no. 6, pp. 130–137, Jun. 2001.
- [36] J. Xue, P. Stuedi, and G. Alonso, "ASAP: an adaptive QoS protocol for mobile ad hoc networks," in *Proc. IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Sep. 2003, pp. 2616–2620.
- [37] X. Yang and N. H. Vaidya, "Priority scheduling in wireless ad hoc networks," in *Proc. ACM MOBIHOC*, Jun. 2002.
- [38] C.-H. Yeh, "A geographically pinned QoS routing scheme for mobile ad hoc networks," in *Proc. IEEE Int. Conf. Netw.*, Aug. 2002, pp. 329–334.
- [39] C.-H. Yeh, H. T. Mouftah, and H. Hassanein, "Signaling and QoS guarantees in mobile ad hoc networks," in *Proc. IEEE Int. Conf. Commun.*, vol. 5, Apr. 2002, pp. 3284–3290.
- [40] Z. Ying, A. L. Ananda, and L. Jacob, "A QoS enabled MAC protocol for multi-hop ad hoc wireless networks," in *Proc. IEEE Int. Conf. Perform., Comput., Commun.*, Apr. 2003, pp. 149–156.
- [41] C. Zhu and M. S. Corson, "QoS routing for mobile ad hoc networks," in *Proc. IEEE INFOCOM*, vol. 2, Jun. 2002, pp. 958–967.



Kuang-Ching Wang (S'98–M'04) received the B.S. and M.S. degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1997 and 1999, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Wisconsin, Madison, in 2001 and 2003, respectively.

He is currently with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, as an Assistant Professor. From 2000 to 2003, he participated in the DARPA Sensor

Information Technology (SensIT) Program as the leading developer of its network protocols and collaborative signal processing applications. His research interests include wireless networks, mobile computing, distributed protocols, and embedded systems.

Dr. Wang is a member of the IEEE Computer, Communication, and Biomedical Engineering Societies and the Association for Computing Machinery.



Parameswaran Ramanathan (S'86–M'89–SM'02) received the B.Tech. degree from the Indian Institute of Technology, Bombay, India, in 1984, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1986 and 1989, respectively.

Since 1989, he has been a Faculty Member in the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, where he is presently a Full Professor. He leads research projects in the areas of sensor networks and next-generation cellular technology. From 1997 to 1998, he took a sabbatical leave to visit research groups at AT&T Laboratories and Telcordia Technologies. His research interests include wireless and wireline networking, real-time systems, fault-tolerant computing, and distributed systems.

Dr. Ramanathan is a member of the Association of Computing Machinery. He is presently an Associate Editor for the *IEEE TRANSACTIONS ON MOBILE COMPUTING* and the *Ad Hoc Networks Journal*. He served as an Associate Editor for the *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED COMPUTING* from 1996 to 1999. He has also served on program committees of conferences such as Mobicom, Mobihoc, International Conferences on Distributed Systems and Networks, Distributed Computing Systems, Fault-Tolerant Computing Symposium, Real-Time Systems Symposium, Conference on Local Computer Networks, and International Conference on Engineering Complex Computer Systems. He was the Finance and Registration Chair for the Fault-Tolerant Computing Symposium (1999). He was the Program Co-Chairman of the Workshop on Sensor Networks and Applications (2003), Broadband Wireless (2004), Workshop on Architectures for Real-Time Applications (1994), and the Program Vice-Chair for the International Workshop on Parallel and Distributed Real-Time Systems (1996).