Price/Utility-based Optimized Resource Allocation in Wireless Ad hoc Networks

Calin Curescu and Simin Nadjm-Tehrani
Department of Computer and Information Science
Linköping University
[calcu/simin@ida.liu.se]

***Abstract*— This paper proposes a scheme for bandwidth allocation in wireless ad hoc networks. The Quality of Service (QoS) levels for each end-to-end flow are expressed using *resource-utility* functions, and our algorithms aim to maximise the aggregated utility of the flows. The scheme differentiates between applications with flexible resource requirements and rigid (real-time) requirements. As an abstract notion of resource, we use *maximal cliques* of mutual interfering links.**

Using concave piece-wise linear utility functions we present a linear programming (LP) formulation of the problem that can serve as an optimal though unrealistic solution. Then we replace this centralised approach with a distributed low complexity solution. A key concept, borrowed from the dual of the optimal allocation problem, is the *shadow price* of a resource.

The contributions of the paper are twofold: (1) a distributed algorithm that allocates the bandwidth based on bids that are calculated using the shadow price of the resources and the flow's utility function, (2) a utility-aware on-demand "shortest" path routing algorithm in which the shadow prices are used a natural distance metric.

We compare the performance of the distributed allocation scheme with the centralised, optimal linear programming solution. We also compare with a non-utility-based QoS allocation scheme, that uses hop-based shortest path routing followed by highest possible bandwidth accommodation of the flow.

I. Introduction

Mobile ad-hoc networks are formed by wireless nodes that move freely and have no fixed infrastructure. Each node in the network may act as a router for other nodes, and flows follow a multi-hop path from a source to a destination.

The infrastructure-less flexibility makes ad-hoc networks a strong complement to cellular networks, making them ideal for many novel scenarios, including cooperative information sharing, defence applications and disaster management. Mobile ad hoc networks will aim to provide a wide range of services in which soft real-time (multimedia), and high priority critical data, should seamlessly integrate. These could further be integrated into core communication networks and Internet. As society becomes dependable on provision of such services the availability of these services under overloads becomes a critical issue.

Wireless networks will always be more resource constrained compared to their wire-line counterparts. In comparison with wire-line networks, wireless multi-hop networks present two fundamental differences. The first major issue is the shared channel with spatial reuse capabilities. Neighbouring nodes could interfere and cannot transmit independently. The second major difference is the mobility of the nodes and its effect on the established paths. This means that paths are constantly created and destroyed (requiring flow rerouting in the latter case). Network resources such as bandwidth and power have to be dealt with in fundamentally different ways compared to the wireline or centralised cellular networks. Resource availability can quickly change, and therefore continuous resource reallocation is needed to provide graceful degradation during overloads.

An ad hoc network that is designed for adaptive and autonomic reaction to failures and overloads should take advantage of the flexibility in the nature of a service that it provides. If every service is associated with multiple levels of acceptable quality, the current flows in the network can be regularly adapted to achieve optimised quality of service (QoS) in the light of resource changes. Our paper presents resource allocation algorithms that provide differentiation among flows and enforce resource assurance for each flow (subject to system-wide optimisation).

Our approach is based on *utility* functions that can be used to capture the value assigned by the client to a flow's different resource allocation levels. This allows for flexible allocations, without the use of online negotiations. Utility functions provide the means for the network to revise its allocation decisions on the fly and optimise resource usage, for instance, by choosing an allocation that maximises the aggregated utility of the flows in the network [1], [2]. This has been shown to be a powerful mechanism for optimising resource allocation instantaneously, but also in a time-aware context – i.e. over the age of a given flow [2]. Reallocations take account of the nature of a flow in terms of flexibility. Best-effort connections are considered to tolerate any resource reallocation whereas real-time flows might require a fixed allocation, otherwise the so far accrued utility will be lost.

In order to deal with the problem of shared resources we use the concept *clique-resource* [3], [4], that allows gathering mutually interfering links in partially overlapping maximal cliques. These deterministically account for bandwidth capacity and interference.

The contributions of the paper are as follows. We present a combined routing, admission control and resource allocation scheme that aims to maximise the aggregated utility over flows and time. For such a scheme to be feasible in a distributed ad hoc setting we use the concept of "shadow price" for a resource, based on the dual of the linear programming (LP) formulation of the problem. Two novel, utility-based algorithms are presented. The first one is a distributed QoS-aware algorithm that allocates bandwidth, at each clique-resource on the end-to-end path of the flow, using only local and flow-related information. Secondly, we present a complementary routing algorithm for choosing the most advantageous path for the flow.

For experimental evaluation we have extended a subset of the J-sim network simulator developed at Ohio State University [5]. We compare the performance of the distributed allocation algorithms with a baseline allocation scheme (shortest path first routing, followed by maximum possible bandwidth allocation). We further study the effects of mobility at different speeds. To provide a comparison with the best possible solution, we also formulate the utility-based optimal allocation problem as an LP problem. Such an approach is infeasible for an online allocation problem, since it needs global knowledge and is too computationally demanding. However, it provides an excellent measure of the upper bound to the performance. Our distributed algorithms are shown to provide an accrued aggregate utility that is merely 10% lower than the optimum achievable by the (centralised) LP solution.

The paper is organised as follows: Section II discusses related work and Section III presents utility functions, network model and the LP formulation of the problem. Our distributed utility-based routing and allocation scheme is described in Section IV. Section V presents the experimental results and Section VI concludes the paper and outlines future work.

II. Related work

Work in resource allocation for ad-hoc wireless networks has been addressed either at the MAC-level, as an extension to routing, or at an optimisation policy level.

For ad hoc networks bandwidth availability can be either precomputed [3], [4], [6] or measured at MAC level [3]. Xue and Ganz [6] compute the available bandwidth at a node as the channel bandwidth minus the bandwidth consumed by the traffic at all neighbours. While easy to implement, this is too pessimistic, and better models can be created when interference structures are built based on link interference [3], [4]. In this work, we use the contention model based on maximal cliques of contending links [3].

If no global optimisation is sought, resource allocation can be attempted independently at every node by appropriate MAC layer design. Luo et al [4] present a packet scheduling approach to ensure a minimum weighted-fair scheduling combined with maximising spatial reuse of channel.

Routing is an important component in multihop wireless networks, and resource allocation/reservation is has been treated as an extension of the routing protocol. For instance, Chen and Nahrstedt [7] propose an on-demand distributed routing algorithm that aims to avoid flooding the network. They consider delay-constrained least cost and bandwidth constrained least cost problems. The feature of the "bandwidth routing" [8] protocol is that link-layer scheduling is directly considered in the protocol. To calculate available bandwidth, both free bandwidth must be known, and a schedule of the free TDMA slots must be constructed.

QoS routing however, is usually not directly aimed at optimal resource allocation but at finding the shortest path that satisfies at least some minimum QoS requirements, or the path that gives the largest margins for a QoS constraint. Karaki et al provide a nice survey on QoS routing problems [9]. In this work however, the routing algorithm is part of the global allocation optimisation scheme.

A seminal work concerning optimal resource allocation and usage of quantised utility functions is presented by Lee et al [1]. Among others, the authors propose an algorithm that uses the convex hull of the utility functions, and yields good results despite computational simplicity. In our work we adopt the same discrete utility function model.

A (re)allocation optimisation scheme for single-hop cellular networks called TARA, has been proposed in earlier work [10], [2]. By taking into consideration the flexibility of different application types to resource reallocations TARA can consistently treat both real-time and best effort connections.

Several other works describe utility-based approaches to resource allocation in multihop wireless networks [11], [12], [13]. Liao et al [12] provide a utility fair max-min allocation for wireless networks. A distributed allocation scheme is used, and periodical reallocations keep the consistency. We believe however, that aiming for equal utility can be counterproductive during overloads, as it will degrade all flows to a lowest acceptable level.

A system that addresses resource allocation in a wireless/wireline access network is the "TIMELY Architecture" proposed by Bharghavan et al [13]. Maximising the revenue based on max-min fairness is one of the criteria used during allocation and adaptation. They employ a 4-tuple revenue model (revenue function, termination credit, adaptation credit and an admission fee), where the same instance of the 4-tuple is used globally. While simplifying allocation, this prevents an accurate differentiation between flows.

Xue, Li and Nahrstedt [11], address a problem that is very similar to ours. They too use concave utility functions that represent user's utility and aim to maximise the aggregated utility of the flows in the network. Both solutions use "shadow prices" of bandwidth resources on the end-to-end path of the flow for steering allocation. However, there are some fundamental differences between the two approaches. Xue et al use non-linear functions while we use piecewise linear ones. Their work builds upon a previous problem formulation by Kelly [14], for wireline networks. The allocation problem is split in two, where a) the network adapts to the rate of flows by changing resource price, b) the flow adapts to the new price by modifying the transmission rate. The iterative algorithm continuously changes the flow's allocation until it reaches an optimal value, and this requires several hundreds of iterations. While this works for long-life flexible flows, it will be unacceptable for inflexible flows needing resource guarantees. In addition, due to mobility and arrivals of new flows, the system could spend little time in an optimal state and flows would suffer frequent oscillations in their allocation.

In our allocation scheme, (re)allocation is considered periodically, and also takes into consideration the effects of a reallocation on the flow's accrued utility. At every decision pont we aim to allocate close to the optimum in one try. Then this allocation will be kept unchanged for at least one period.

In an extension, Xue et al [3] consider a mobile environment and use AODV as a routing algorithm. AODV routes over the shortest path (number of hops) and this might overload inner network paths while resources will go unused towards the marginal areas. Therefore, we use the price-based approach also for routing decisions.

III. Problem formulation

In this section we first layout the network model, followed by a LP formulation of the allocation problem. We then go to present the notion of shadow price and some properties of the optimal solution that will be used when constructing the distributed algorithm.

A. Network model

We consider a wireless ad-hoc network with n nodes. Nodes a and b that are in transmission range of each other are connected by a wireless link, l_{ab}. Nodes communicate with each other by means of multi-hop bidirectional end-to-end flows (f_i) between an originator (source) node and a destination node.

In ad hoc wireless networks, we have a location-dependent contention between the transmissions on the wireless links. Transmissions over a link can be bidirectional, thus two links contend with each other if one of the end-nodes of a link is within the transmission range of an end-node of the other link [11], [4]. A link contention graph can be constructed, where vertices represent links, and an edge connects two vertices if the corresponding links contend with each other. Each maximal clique[1] in such a graph represents a distinct maximal set of mutual contending links.

A necessary condition for a *feasible* bandwidth allocation is that for each maximal clique the bandwidth allocated over all links forming the clique is less than or equal to channel capacity. The channel capacity gives only an upper bound, as in practice, the choice of transmission scheduling algorithm, and even the topology of the cliques [15] can impose a tighter bound. That is,

$$\forall j, \quad \sum_{l \in r_j} lb_l \leq B_j^{max} \tag{1}$$

where lb_l is the allocated bandwidth over wireless link l, r_j is a given maximal clique, and B_j^{max} is the achievable clique capacity, that is less or equal to the wireless channel capacity.

Hence, each maximal clique can be regarded as an independent *clique-resource* with capacity B_j^{max}. Since only links close to each other contend for the same bandwidth, local information is sufficient for constructing the cliques [2] that a certain link belongs to (details in Section IV-E). More description and proofs are given by Xue et al [11], [3].

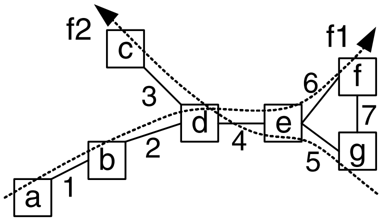

Fig. 1. Network example

In Figure 1 we present an example of a network topology (the mobile nodes are represented as squares) and two ongoing flows using this network. Figure 2 presents the link contention graph, where vertices represent the links (identified by corresponding numbers) of the network in Figure 1. We can identify three maximal cliques representing resources. Note that a single flow can span over several links belonging to the same clique-resource. Let q_{ij} represent how many links of clique r_j are used by flow f_i. Transmissions over the links in a clique are mutually exclusive, and this means that the "effective" bandwidth used by the flow is q_{ij} times higher than its nominal rate. Let m be the total number of flows, and b_i a certain allocation to flow f_i. Finally we can rewrite the constraints in Equation 1 in relation to the bandwidth allocated to the flows:

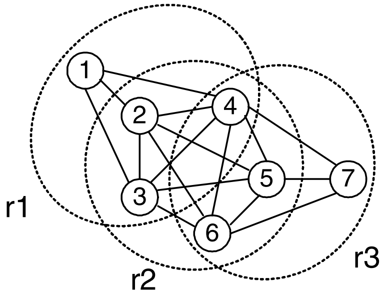

Fig. 2. Link contention graph for network example

$$\forall j \quad \sum_{i=0}^{m} q_{ij} \times b_i \leq B_j^{max} \tag{2}$$

Table I presents the values of q_{ij} for the example in Figures 1 and 2.

[1] A maximal clique is a subset of vertices, each pair of which defines an edge, that cannot be enlarged by adding any additional vertex.

[2] We assume that the communication range is the same as the transmission range. Otherwise, bandwidth estimation has to be used, since two nodes could interfere but not be able to communicate.

TABLE I

FLOW-RESOURCE USAGE FOR NETWORK EXAMPLE

q_{ij}	r_1	r_2	r_3
f_1	3	3	2
f_2	2	3	2

B. Utility functions

The utility of an application represents the value assigned by the user to the quality of the application's results, and may relate to willingness to pay for the service. Utility is therefore measured with an abstract unit that can be mapped to other units according to billing models. We assume that the utility of an application's flow is set by its user and not just an internal system parameter for steering allocation. In this paper, we use quantised bandwidth-utility functions. Thus we can represent the function as a list of bandwidth utility pairs, $u_i = \left(\binom{U_i^1}{B_i^1}, ..., \binom{U_i^k}{B_i^k} \right)$ where k is the number of utility levels of flow i. For an allocated bandwidth b_i, the momentary accrued utility is denoted by $u_i(b_i)$. Utility functions could take any shape, which makes the optimal allocation problem NP-complete even in a single resource case. Nevertheless Lee et al [16] obtained results very close to the optimum when approximating general utility functions with their convex hull frontier. We also use convex hull frontier approximation and thus, for the rest of the paper we assume utility functions to be concave and piecewise linear.

Utilty functions do not inherently reflect application flexibility with respect to resource reallocations. Nevertheless, our system can treat rigid applications (e.g. firm real-time) differently from flexible applications (e.g. file transfer). This is done by changing the user-provided utility functions at runtime (e.g. for a real-time flow its utility function will "grow" with age, expressing the importance of not losing invested resources). Note that these changes are orthogonal to the allocation algoritm presented in this paper. The illustration of the time and reallocation-aware approach (TARA) in a cellular single hop setting, can be found in earlier work [10], [2].

C. Linear programming form of the optimisation problem

For the wireless multihop network, having computed all the clique-resources, and assuming for now that for every flow the path between source and destination is set (routing done), at any (re)allocation moment we can formulate the following problem. Let m be the number of flows, u_i the utility function and x_i is the allocation to be determined for flow i. Let, p be the number of clique-resources and q_{ij} the usage counter of clique-resource j by flow i. Then the optimal allocation for all x_i over all cliques j can be obtained from:

$$Maximise \quad \sum_{i=1}^{m} u_i(x_i) \tag{3}$$

$$subject\ to \quad \sum_{i=1}^{m} q_{ij} \times x_i \leq B_j^{max} \tag{4}$$

$$x_i \geq 0 \tag{5}$$

where B_j^{max} is the maximum bandwidth available for clique j.

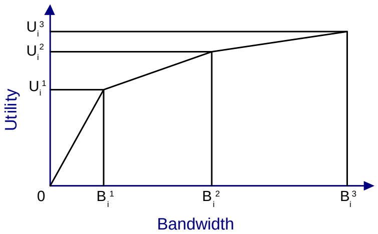

Fig. 3. R-U function example

Due to the use of convex hull, the resource-utility functions in Equation 3 are piece-wise linear. To entirely linearise the problem we conceptually split a flow in several parallel subflows (same source, destination, and path), each corresponding to a linear part of the utility function. For a subflow k of flow i the allocation is constrained as follows, $b_i^k \leq B_i^k - B_i^{k-1}$. The *utility efficiency* of the subflow (utility/bit) is $\lambda_i^k = \frac{U_i^k - U_i^{k-1}}{B_i^k - B_i^{k-1}}$. In Figure 3 we highlight the boundaries of the 3 subflows. Then given s segments in the utility function, for allocations b_i^k to subflows of flow i, we have:

$$u_i(b_i) = \sum_{k=1}^{s} \lambda_i^k \times b_i^k \tag{6}$$

where $b_i = \sum_{k=1}^{s} b_i^k$. However, not any allocation to the subflows is consistent. In order to use the right side of Equation 6 as a function, we have to add two constraints:

(C_1) Every k-th subflow has a maximum allocation limit, that is any $b_i^k \leq b_i^{k\ max}$ where $b_i^{k\ max} = B_i^k - B_i^{k-1}$.

(C_2) The order of the segments in the R-U function must be respected when allocating (i.e. "gaps" are not allowed). That is, if $b_i^k > 0$ then for all $l < k$, $b_i^l = b_i^{l\ max}$.

Constraint (C_1) is linear and can be directly used in the problem formulation. Due to concavity, the linear segments of the utility function are ordered highest efficiency first, and therefore an optimal allocation to the subflows will automatically respect (C_2). We will show this by first formulating the subflow optimal allocation problem and then go on to prove that any solution to the problem satisfies (C_2).

Proposition 3.1: Consider the following formulation of the maximisation problem, where x_i^k are the allocation variables:

$$Maximise \quad \sum_{i=1,k=1}^{m,s} \lambda_i^k \times x_i^k \tag{7}$$

$$subject\ to \quad \sum_{i=1,k=1}^{m,s} q_{ij} \times x_i^k \leq B_j^{max} \tag{8}$$

$$0 \leq x_i^k \leq b_i^{k\ max} \tag{9}$$

Then, the results of the maximisation problem satisfy constraint (C_2).

Proof: Let's assume the opposite, which means that there are two subflows $l < k$ of a flow i where $b_i^l < b_i^{l\ max}$

and $b_i^k > 0$, where b_i^k is the value allocated to x_i^k, etc. Let $\gamma = min(b_i^{l\,max} - b_i^l, b_i^k)$. We denote the utility generated by subflows l and k by $U_{l+k} = \lambda_i^l \times b_i^l + \lambda_i^k \times b_i^k$. Since both subflows belong to the same flow, there is no restriction to subtract γ from subflow k and to allocate it to subflow l. Let $U'_{l+k} = \lambda_i^l \times (b_i^l + \gamma) - \lambda_i^k \times (b_i^k - \gamma)$. Then $U'_{l+k} > U_{l+k}$ because $\lambda_i^l > \lambda_i^k$ for a concave function. The allocations for other subflows being equal, this means that b_i^l, b_i^k are not optimal. Contradiction. ■

While solvable with LP, the following drawbacks make the LP solution infeasible for online allocation in an open, dynamic ad hoc network.

- Centralised allocation. The algorithm needs to centralise and disseminate back information about all nodes and all flows. This includes keeping track of the changing topology and dealing with the generated signalling overhead.
- Time complexity. The computational overhead created by the LP solver is prohibitive for an online allocation.

We aim for a distributed, low complexity allocation scheme. Nevertheless, we will use the LP allocation as an "upper bound" baseline to evaluate the performance of the distributed algorithm.

The distributed scheme uses the *shadow price* concept so we first present the dual formulation of the above LP problem together with some useful characteristics of the optimal solution.

D. Dual formulation and characteristics of optimal solution

In economic terms the dual optimisation problem can be explained as follows. Assume somebody would like to buy a very small amount of a resource. The *shadow* (or marginal) *price* should be the minimum price the owner should accept. Obviously for offers lower than the shadow price the owner would gain more using the resource in his own production. Thus, the shadow price is a measure of resource contention and shows the marginal increase in utility if more resource would be available. The following is the dual of the previous LP problem (Equations 7-9):

$$Minimise \sum_{j=1}^{p} B_j^{max} \times y_j + \sum_{i=1,k=1}^{m,s} b_i^{k\,max} \times v_i^k \tag{10}$$

$$subject\,to \sum_{j=1}^{p} q_{ij} \times y_j + v_i^k \geq \lambda_i^k \tag{11}$$

$$0 \leq y_j,\ 0 \leq v_i^k \tag{12}$$

The shadow price of clique-resource j is denoted as y_j, the number of all clique-resource being p. Note that the shadow price is the price per unit of resource (i.e. price/bit). Since subflows are by default constrained to a maximum bandwidth $b_i^{k\,max}$ this is modelled as a limited resource, which is used only by the respective subflow, and v_i^k denotes the shadow price of the "artificial" resource.

We now define the slack variable w_j as the amount of unused capacity of clique-resource j. For a given flow f_i the slack variable z_i^k represents "loss per unit", i.e. the difference between "benefit per bit", λ_i^k, and the sum of the shadow prices of used clique-resources. Using the slack variables, the inequalities of the primal and dual problem become:

$$\sum_{i=1,k=1}^{m,s} q_{ij} \times x_i^k + w_j = B_j^{max} \tag{13}$$

$$\sum_{j=1}^{p} q_{ij} \times y_j + v_i^k - z_i^k = \lambda_i^k \tag{14}$$

According to LP theory, the optimal solutions for the primal and dual problems fulfil the following constraints [17] (constraint 17 is similar to 16, but applies the above mentioned "artificial" resources):

$$x_i^k \times z_i^k = 0 \tag{15}$$

$$y_j \times w_j = 0 \tag{16}$$

$$v_i^k \times (b_i^{k\,max} - x_i^k) = 0 \tag{17}$$

From equations 13-17 we can identify the following characteristics of the optimal solution:

(O_1) If a resource is underutilised ($w_j > 0$) then its shadow price is zero ($y_j = 0$), otherwise its price is greater than zero.

(O_2) For subflows where $z_i^k > 0$, we have $\sum_j q_{ij} \times y_j > \lambda_i^k$, meaning that the accumulated price is higher than the subflow utility efficiency. Thus $x_i^k = 0$, and so $v_i^k = 0$.

(O_3) For subflows where $z_i^k = 0$, and $v_i^k = 0$, we have $\sum_j q_{ij} \times y_j = \lambda_i^k$. This means that the subflow is at the allocation edge given the resources it uses.

(O_4) For subflows where $v_i^k > 0$, we have $\sum_j q_{ij} \times y_j < \lambda_i^k$. Also, $x_i^k = b_i^{k\,max}$ and $z_i^k = 0$. Parameter $v_i^k > 0$ represents a "pricing slack", i.e. the amount by which the accumulated prices of the used resources could increase, and the flow still be profitable.

IV. Distributed routing and resource reallocation

The ad hoc network considered in this work is an open dynamic system where resource requests and availability are always changing. Thus, our scheme employs periodic reallocations to keep the resource usage optimised. As end-to-end connections span several nodes and clique-resources, it is important that (re)allocations are well coordinated along the path. Also, reallocations imply a "mode" change for applications so their number should be strictly controlled. The use of periodic, synchronised allocation rounds guarantees that flows will enjoy an allocation for at least one period. It also puts a bound on the reallocation rate in the system, even if the rate of events (traffic and topology changes) is much higher. The only disadvantage is that new and rerouted flows must wait until the next allocation time to receive new resources. Choosing an appropriate period will imply tradeoff between a) utility optimisation and reducing the delay of path establishment and b) the computation and signalling overhead of allocation rounds.

The algorithm will be referred as adhoc-TARA in the rest of the paper. Assume that a route for a flow is given (we will come back to how this route is found shortly). Conceptually, at each period the (re)allocation will proceed like this:

- Every flow calculates a bid for all clique-resources it traverses, based on their associated *shadow prices*.
- Each clique-resource independently evaluates the bids, proposes a certain bandwidth allocation to the flow and recalculates its shadow price.
- The flow chooses the lowest bandwidth proposal from all the cliques it traverses as the new bandwidth for the new period.

Since a flow is constrained by the lowest available bandwidth on its path, the allocations must be performed synchronised at all clique-resources.

A. Bid construction

For each flow, the shadow prices (determined at the previous allocation round) of all clique-resources on its end-to-end path are used to calculate its path-price, $pp_i = \sum_j q_{ij} \times y_j$. Intuitively, the utility efficiency (λ_i^k) represents the maximum "budget" that could be "paid" to the traversed resources keeping the flow still "profitable". Note that both λ_i^k and pp_i measure utility and price per bit. Thus we can now compute the *price slack* of every subflow as the difference between the utility efficiency and the path price, $slk_i^k = \lambda_i^k - pp_i$. After the new allocation, the price could increase for any of the used resources, so we uniformly divide the slack among all the cliques it traverses (cc_i is the clique counter for flow i, $cc_i = \sum_j q_{ij}$). Thus, for each clique-resource we create the following bid:

$$bid_{ij}^k = y_j + \frac{\lambda_i^k - pp_i}{cc_i} = y_j + \frac{\lambda_i^k - \sum_j q_{ij} \times y_j}{\sum_j q_{ij}} \tag{18}$$

where bid_{ij}^k is the bid of flow i, to resource j for subflow k. The sum of a subflow's bids amounts to its maximum "budget" (λ_i^k). Thus, if all bids are accepted, the subflow is accepted, and corresponds to either category (O_3) or (O_4) in Section III-D.

B. Independent allocation

After all the bids have been placed, every clique-resource will independently allocate the bandwidth, to the subflows in decreasing order of bids, until bandwidth is depleted. Then the *new shadow price* of the resource is set to the price of the lowest bid among the accepted subflows. Note that all the bandwidth is reallocated, and some subflows might get this time an allocation different from last period.

If the contention at a certain resource is greater than that at the previous allocation, its price will increase. If the bid could not accommodate this increase, the subflow will be rejected. If the contention decreases, the price of a resource will decrease. This means that subflows that bid less than the previous price (i.e. have a negative price slack) are accepted, bringing the price down accordingly. If a resource does not allocate all bandwidth, it becomes underloaded and its shadow price becomes "0" (case (O_1) in Section III-D).

C. Discussions

Note that if we could use the solutions of the dual problem as shadow prices when constructing bids, any subflow, at all the cliques that it traverses, would consistently be accepted or rejected. As we do not know the new shadow price beforehand, we use the shadow price from the last allocation as an estimate. If at a clique-resource the contention level has increased, the price could increase more than the bid by some flows (resource price was underestimated at bid construction). In such a case, some flows that could have offered a proper bid (with hindsight) are rejected. Conversely, overestimating a resource unnecessarily increases its bid to the detriment of others.

As a consequence of over/underestimation, for a flow, bandwidth could be allocated in different amounts at different clique-resources, and the flow can use only the minimum allocation over the end-to-end path. As a remark, we want to mention that in such cases one might use the algorithm iteratively, to better balance the bids. Nevertheless, as the algorithm is intended for online allocation, we do not iterate and any mis-allocated bandwidth will remain unused. Since in an optimal allocation, the amount of this mis-allocated bandwidth would be zero, we will use it in our experiments as another measure of how close to the optimal allocation our algorithm performs.

Figure 4 presents a pseudocode of the distributed algorithm that is run synchronised for every clique-resource and respectively for every flow. For every clique-resource a clique-leader node, which is used for performing the (re)allocation computations, is determined at clique-construction time (see Section IV-E). Whenever a flow starts/stops using a wireless link, one of the end-nodes of the link registers/deregisters the flow (i.e. its source node) with the clique-leaders of all the cliques containing that link. The clique-leader gathers information about the flows using the clique-resource and runs the allocation algorithm. The natural place for running the flow-part of the algorithm (and changing the transmission rate) is at the flow's source node.

Note that the signalling information (between clique-resources and the flows' source nodes) is sent only along established flows, and thus can be piggy-backed on existing packets. The clique-leader can be chosen such that the distance to the end-nodes of the links belonging to the clique is at most 2 hops (see also Section IV-E). Therefore, inside-clique signalling could use the MAC layer signalling (e.g. piggyback RTS,CTS,ACK packets). As reallocations occur seldom (compared to flow dynamics) we envision a small signalling overhead.

The flow's source node must receive the new bandwidth decision from all the clique-resources on the end-to-end path of the flow and choose the minimum allocated. The larger the synchronisation error between the clocks of the clique-leaders, the more the source-node has to wait until it can set the new

```
Allocation Algorithm run at every clique-leader $j$,
at every period $T$:
  Let $F_j$ be the set of flows using resource $j$
  $awb_j = B_j^{max}$      //initialise available bandwidth
  $\forall$ subflows $f_i^k \in F_j$
      $bid_{ij}^k = y_j + (\lambda_i^k - pp_i)/cc_i$      //compute bid
  while $F_j \neq \emptyset$      //allocate for highest bidder first:
      select $f_i^k \in F_j$ with highest bid
      if $awb_j > q_{ij}^k \times b_i^{k\ max}$
          $x_{ij}^k = b_i^{k\ max}$
          $awb_j = awb_j - q_{ij}^k \times x_{ij}^k$
      else
          $x_{ij}^k = 0$
      $F_j = F_j - f_i^k$
  $y_j = min(bid_{ij}^k \mid x_{ij}^k > 0)$      //recompute resource price
  $\forall i$ where $f_i \in F_j$
      $x_{ij} = \sum_k x_{ij}^k$
      send $x_{ij}$ and price $y_j$ to source of $f_i$

Flow adaptation algorithm run at every node $n$,
at every period $T$:
  $\forall$ flows $f_i$ sourced at node $n$
      $\forall$ resources $j$ that $f_i$ traverses
          wait for allocation $x_{ij}$ and price $y_j$
      $x_i = min(x_{ij})$      //set bandwidth of $f_i$
      $pp_i = \sum_j q_{ij} \times y_j$      //recompute its path price
      $\forall$ resources $j$ that $f_i$ traverses
          send $pp_i$ to clique-leader of resource $j$
```

Fig. 4. The distributed allocation algorithm

rate of the flow. As a consequence a flow might increase its rate before another decreases it, leading to short-lived congestions at certain points. Regarding clock-synchronisation protocols in wireless (sensor) ad hoc networks, Römer et al. [18] give precision results of less than $100\mu sec$ for nodes five hops away. These clock skews are small compared to the envisioned reallocation periods, and thus we assume these congestions to be easily mitigated. Nevertheless, synchronised allocation generate bursty signalling in the network, and our ongoing work studies methods that potentially remove the need for synchronised allocation rounds among the clique-resources.

D. *QoS routing*

Traditional QoS routing algorithms typically use either shortest path (respecting minimum constraints), or widest path (allowing a better QoS for that flow). However, these are two extreme cases and do not optimise global utility. Shortest path might overload some routes. Widest path may produce too long routes, increasing total network load. Therefore, as part of adhoc-TARA we propose a new routing algorithm based on the shadow price of resources introduced above. Used with the allocation algorithm presented in Section IV-C the best chance for the highest QoS is along a path with the lowest path price. So we use an on-demand shortest path first (SPF) routing algorithm that uses the path price as distance metric (i.e. it chooses a path that yields a minimal $pp_i = \sum_j q_{ij} \times y_j$)). The lowest path price comes from both less contended links (lower link prices) and shorter topological paths (lower number of links).

Once chosen, keeping a route fixed is important for deterministic resource allocation. For this we employ a source-routing algorithm. In source routing the source specifies the hops to destination, and routing tables are not needed. Besides providing load balancing capabilities, it prevents load oscillations when e.g. the shortest path changes.

Rerouting is performed for flows only when a link in the end-to-end path breaks due to mobility. There are two reasons why not to perform rerouting in the case of a decrease in allocation. First, this would create an oscillating allocation pattern where flows constantly chase a better route. Second, rerouting implies a big signalling overhead, and should be used only if necessary. Routing or rerouting is asynchronous to allocation, however the shortest paths are valid only until next allocation round due to price change.

E. *Mobility and clique construction*

Due to mobility, a node might enter or exit the communication range of another one, thus creating a new wireless link, or alternatively breaking one. Handling topology changes can be implemented either event based (when MAC feedback is used) or periodically (when hello messages are broadcast), and should be independent of the allocation algorithm. For clique (re)computation, new and broken links should be reported to all nodes as far as 3 hops away. This will enable all nodes to completely determine all the clique-resources containing any of its adjacent links [11], [3]. Thus, the signalling overhead involved in clique (re)computation greatly depends on network mobility.

If a link breaks, all the flows that used this link should be re-routed. Some old clique-resources will disappear and some new ones will be created. The new cliques will have an initial 0 shadow price. However, to set a better starting price, we perform a "dry allocation" (no bandwidth is actually reallocated) at the new clique-resources, based on the inherited flows. After a topology change all the affected flows must update their path price, to be used in the next allocation round.

In Figure 5 the clique-reconstruction algorithm is presented. All the nodes identify all the relevant clique-resources independently. Then, for each clique-resource the "clique-leader" node is determined (e.g. the node in the clique that has the highest identifier, and is adjacent to at least two links belonging to the clique). The algorithm implies waiting for results from other nodes, and thus runs synchronised with a period T'.

V. Simulation and results

A. *Evaluation setup*

To evaluate the behaviour of our resource allocation scheme we use a traffic simulating applications used in the Internet and representative for a future mobile communication network [19], [2]. Table II summarises their characteristics. To create a diverse traffic mix, the maximum required bandwidth

```
**Clique reconstruction algorithm at every node $n$,
at every period $T'$:**
  local (one-hop) broadcast “hello” message
  timeout-wait
  if the neighbour set changed
      notify sources of broken flows to reroute
      three-hop broadcast the *new* and *broken* links
  timeout-wait
  (re)construct clique-resources due to *new* and *broken* links
  $\forall$ new resources $j$ where $n$ is clique-leader
      establish initial $y_j$
      $\forall$ flows $f_i$ traversing resource $j$
          send $q_{ij}$ and $y_j$ to source of flow $f_i$
  timeout-wait
  $\forall$ flows $f_i$ sourced at node $n$
      if $f_i$ is broken then reroute
      else
          $pp_i = \sum_j q_{ij} \times y_j$      //recompute its path price
          $cc_i = \sum_j q_{ij}$      //recompute its clique-counter
          $\forall$ resources $j$ that $f_i$ traverses
              send $pp_i$ and $cc_i$ to clique-leader of $j$
```

Fig. 5. The clique reconstruction algorithm

follows a geometric distribution with the given minimum, maximum and mean values (columns 2 and 3).

The second column from the right shows the flexibility class, which depends on the flexibility of the applications to bandwidth reallocation. Next we shortly present the classification. [10], [2].

- Class I represents rigid applications, e.g., for a real-time application once the “mode” is set by the initial allocation any allocation increase is useless and decrease fatal. That is the utility accumulated in time for this flow is completely lost if resources are decreased.
- Class II is semi-rigid, where the lowest allocation point is used to compute the utility for the whole duration (i.e. if resource is decreased a proportional chunk of already accumulated utility will be lost). Examples could be sensor flows with different accuracy, or sensitive multimedia.
- Class III represents fully flexible applications that have no problem to adapt (for every new allocation period the accumulated utility of the flow grows with the utility of the given allocation). Examples are non-real-time data transfers.

The shapes of the utility functions are more or less similar, and not presented due to space considerations. The “relative importance” factor is important, and shown in the rightmost column of Table II. It represents the utility per bit associated with the maximum required bandwidth, and scales the shape of the utility function accordingly. For example, even though one might be ready to pay roughly three times more for a video-phone conversation (bandwidth demand of 256 Kbps), the utility per bit is almost three times higher for an audio-phone application (which requires only 30 Kbps).

TABLE II

TRAFFIC MIX USED IN THE EXPERIMENTS

Applic. Group	Max. Bandwidth Requirement (Kbps)			Connection Duration (sec)			Examples	Class	Utility Scaling Factor
	min	Max	avg	min	max	avg			
1	30	30	30	60	600	180	Voice Service & Audio Phone	I	1
2	64	256	128	60	1800	300	Video-phone & Video-conference	II	1/3
3	200	1000	500	300	7200	600	Interact. Multimedia & Video on Demand	II	1/10
4	10	30	20	10	120	30	E-Mail, Paging, & Fax	III	1/2
5	64	512	256	30	7200	180	Remote Login & Data on Demand	III	1/5
6	128	2000	512	30	1200	120	File Transfer & Retrieval Service	III	1/7

A simulator was built on top of the j-sim component platform [5], however packet level simulation was not considered at this stage. The experiments use $1500 \times 1500m^2$ area where 60 mobile stations are uniformly, randomly deployed. The communication range is $250m$ and considered equal to the interference range. Environmental perturbations are not considered, and every clique-resource has the $4Mb/s$ channel bandwidth at their disposition.

Mobility is implemented using the random way-point model, with a random speed between 0 and max_speed. To ensure a nice connectivity and keep the mobile nodes from clumping together, we made nodes move away from each other when they come closer than a third of the communication distance. We believe this is a reasonable model in urban connectivity. New flows arrive following an exponentially distributed inter-arrival time. All the 6 application groups arrive with equal probability. To solve the linear programming part, we have used a Java package from the operation research objects collection (OR-Objects) [20].

B. Comparison of allocation schemes

In our experiments we compare the behaviour of the following routing and allocation schemes for different load and mobility scenarios.

- As a baseline algorithm we use a non-utility routing and allocation scheme denoted by *simple* in the experiments. The routing is on-demand shortest path first (hop-based). After a route is chosen, the minimum of the bandwidth available at all clique-resources on the end-to-end path is allocated to the flow. Actually if not enough bandwidth is available to accommodate the minimum then the flow is rejected. If the path breaks, the flow is rerouted, and new bandwidth allocated. If a clique-resource becomes overloaded due to mobility, flows will be dropped on a last-accepted first-rejected basis.
- To represent best possible solution, we use a *LP* solver to optimally solve the global allocation problem as defined in Section III-C. The formulation of the LP problem does not include routing, so we use the price routing algorithm described in Section IV-D. This serves to compare our distributed allocation algorithm with the optimal allocation.

- Next we show the results of runs for the *adhoc-TARA* scheme. It uses the distributed allocation algorithm described in Sections IV-A to IV-C and the routing algorithm from Section IV-D.
- Finally, we compare with a variant of our distributed allocation scheme, denoted *altbid*, where a different formula is used to construct the bids. In this alternative the "budget" (λ_i^k), is proportionally divided based on shadow prices. Thus, $bid_{ij}^k = \frac{y_j \times \lambda_i^k}{\sum_j q_{ij} \times y_j}$. The intuition is that resources with higher shadow prices are more disputed, and thus have a higher chance of getting even more disputed. However, bids for low priced resources become very small, and in the case of a 0 priced resource, all $bid_{ij}^k = 0$. In this case, ties are broken by $slk_i^k = \lambda_i^k - \sum_j q_{ij} \times y_j$.

For all the four schemes, allocation is separated from utility accounting. Utility accounting is performed using the extended utility model briefly described in Section V-A. According to the model, a class I flow is dropped if the initial bandwidth cannot be maintained. A class II flow is dropped if the minimum bandwidth cannot be maintained. A class III flow is never dropped (unless there is a network partition) as it can recover after zero allocation. If a flow is dropped, no utility is gained for the flow, and the bandwidth invested during its lifetime is wasted.

C. Experimental results

As utility is our main performance metric we will first show how the total utility of the four schemes behaves when subjected to scenarios with different mobility. Thus in Figures 6 and 7, on the X-axis we have the average speed of the nodes (m/s), and on the Y-axis the time-accumulated system utility. Every point represents an average of 3 different experiments. Each experiment was run over a period of $600sec$, with a (re)allocation period of 2 seconds. All the experiments were run with a moderate overload (average inter-arrival rate of $1/200s$).

The experiments in the two figures are differentiated by the type of the applications used. In Figure 6 ("rt-mix" scenario) we consider a mix of rigid and flexible application groups as presented in in Table II (see the "class" column). In Figure 7 we consider that all the 6 application groups are fully flexible (their class is set to class III). In this case, no flows will be dropped due to zero allocation.

We can see that the results of adhoc-TARA come surprisingly close to the optimal LP allocation. Even at the lowest point, the distributed allocation algorithm is at almost 90% of the optimal allocation (note the LP algorithm uses the same routing as adhoc-TARA). Both the "flexible" and the "rt-mix" scenarios suffer from mobility in similar ways (i.e. the "rt-mix" scenario is not affected more). The simple scheme cannot differentiate properly between flows and is trailing at around half of the utility of the LP algorithm. The altbid algorithm performs constantly below adhoc-TARA (at worst 72% of the LP). This is because the bid is too biased towards high-priced resources, while low priced resources can also quickly increase their prices. Adhoc-TARA, on the other hand, creates a more evenly distributed bid.

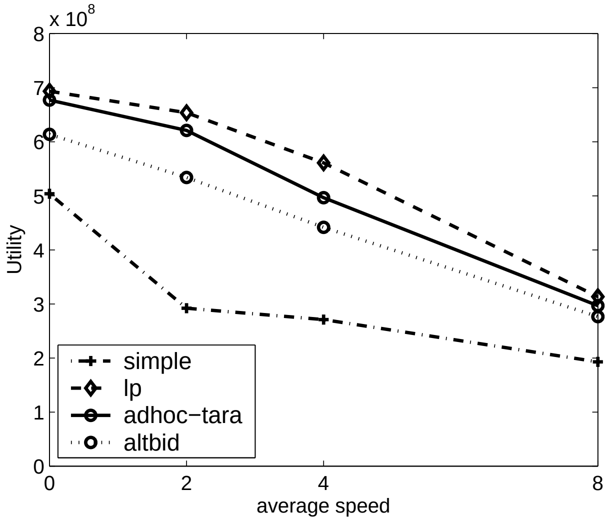

Fig. 6. Utility for rt-mix traffic

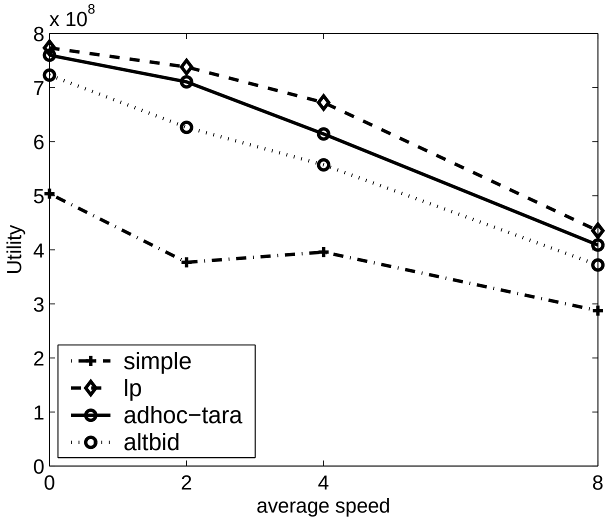

Fig. 7. Utility for flexible traffic

In Table III we present the bandwidth utilisation of adhoc-TARA as compared to the utilisation of the LP algorithm. The mobility row shows the average speed of nodes (m/s). Our distributed algorithm independently allocates bandwidth at the clique-resources along the flow's path. If allocations are different, some bandwidth is wasted. The LP solution is using global knowledge and thus has no such problem. Nevertheless, the difference in bandwidth usage (as an average over all clique-resources) between LP and adhoc-TARA is only around 15% as presented in Table III.

A more detailed view on the QoS for the 6 application groups (from Table II) is given by the experiments presented in Table IV. The two cases have the same offered load (the

TABLE III

BANDWIDTH USAGE OF ADHOC-TARA COMPARED TO LP

mobility	0	2	4	8
usage(%)	92	84	80	87

number of arriving flows is around 190), and are using adhoc-TARA scheme, and “rt-mix” traffic. We show the number of flows that were blocked and dropped, and also the average allocation (level) granted to a flow as a percentage of the maximum it requested. As expected, flows with lower general utility (groups 3 and 6, see last column in Table II) fare worst. Also, the mobile case (second row) creates more congestion and mis-allocations than the static case (first row), and performance drops in general. If utility is high enough, and mobility is not excessive, the QoS of rigid real-time flows is well preserved (e.g. application group 1).

TABLE IV

STATISTICS AT APPLICATION GROUP LEVEL

	appGroup	1	2	3	4	5	6
mobility	blocked	0	0	5	0	0	4
0	dropped	0	1	13	0	0	0
	level(%)	100	84	54	98	72	68
mobility	blocked	0	2	11	0	6	6
4	dropped	0	7	16	0	2	0
	level(%)	100	76	50	97	76	70

The next set of experiments, Figure 8, show how utility depends on the offered load. In all experiments, average speed is 4. On the X-axis, we plot the average inter-arrival rate for a light (1/400), moderate (1/200) and heavy (1/100) offered load. We can observe that utility of all 3 schemes follows similar trends and increases almost proportionally with load, preserving the superior performance of adhoc-TARA.

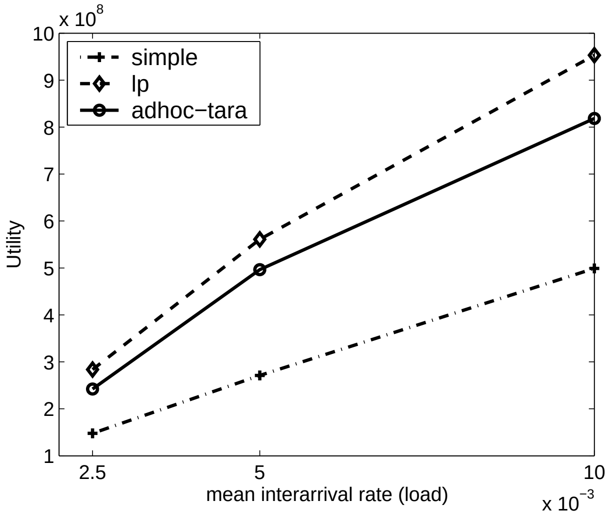

Fig. 8. Utility as function of traffic load

Besides signalling overhead, computational complexity is a big drawback of the LP solution and one of the reasons to choose adhoc-TARA. Table V gives a comparison of the average time (seconds, on a 1GHz P III) needed to reach an allocation decision, as offered load is increased.

TABLE V

AVERAGE RUNNING TIME OF ALLOCATION DECISION

inter-arrival rate	1/400	1/200	1/100
LP	54.1	173.8	890.9
adhoc-TARA	0.179	0.265	0.476

The above experiments gave a good indication of relative merits of the the distributed allocation algorithm of adhoc-TARA compared to optimal LP, as in both cases we use the price-based routing algorithm. Current work involves isolating the characteristics of the price-based routing. Surprisingly, preliminary results, using networks with no hot-spots, show no important utility improvements by using the price-based routing instead of the hop-based shortest path first (SPF). However, more work needs to be done to study price-based routing in congested networks.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a novel utility/price-based bandwidth allocation scheme for wireless networks, together with a complementary price-based routing algorithm.

We first show that we can use utility functions together with techniques derived from linear programming for optimising resource allocation in multihop ad hoc networks. We then propose a distributed allocation algorithm that bids for resources depending on their “shadow prices”, and the “utility efficiency” of the flow. Simulations show a superior behaviour of the distributed allocation algorithm, which comes close to the optimal linear programming allocation, and has a much lower overhead.

As future work we intend to study the convergence properties of the algorithm and the benefits and overheads of using several iterations at each allocation point in order to further improve the allocation decision. The behaviour of the routing algorithm in topologies with hot-spots configurations is also interesting to follow.

ACKNOWLEDGEMENT

This work was supported by CUGS (the National Graduate School in Computer Science, Sweden).

REFERENCES

[1] C. Lee, J. Lehoczky, R. Rajkumar, and D. Siewiorek, “On quality of service optimization with discrete qos options,” in *Proceedings of the IEEE Real-time Technology and Applications Symposium*, June 1999.

[2] C. Curescu and S. Nadjm-Tehrani, “Time-aware utility-based resource allocation in wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 7, pp. 624–636, July 2005.

[3] Y. Xue, B. Li, and K. Nahrstedt, “Optimal resource allocation in wireless ad hoc networks: A price-based approach,” *IEEE Transactions on Mobile Computing*, 2005, to appear.

[4] H. Luo, S. Lu, V. Bharghavan, J. Cheng, and G. Zhong, “A packet scheduling approach to qos support in multihop wireless networks,” *Mobile Networks and Applications*, vol. 9, no. 3, pp. 193–206, 2004.

[5] http://www.j sim.org/, “J-sim homepage.”

[6] Q. Xue and A. Ganz, "Ad hoc qos on-demand routing (aqor) in mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 63, no. 2, pp. 154–165, 2003.

[7] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad-hoc networks," *IEEE Journal on Special Areas in Communications*, vol. 17, no. 8, pp. 1–18, aug 1999.

[8] C. R. Lin and J.-S. Liu, "Qos routing in ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1426–38, 1999.

[9] J. N. Al-Karaki and A. Kamal, *Quality of Service Routing in Mobile Ad hoc Networks: Current and Future Trends*. CRC Publishers, 2004, ch. Mobile Computing Handbook.

[10] C. Curescu and S. Nadjm-Tehrani, "Time-aware utility-based qos optimisation," in *Proceedings of the 15th Euromicro Conference on Real-time Systems*. IEEE Computer Society, July 2003, pp. 83–93.

[11] Y. Xue, B. Li, and K. Nahrstedt, "Price-based resource allocation in wireless ad hoc networks," in *Proceedings. of the 11th International Workshop on Quality of Service (IWQoS), also in Lecture Notes in Computer Science*, vol. 2707. ACM Springer-Verlag, jun 2003, pp. 79–96.

[12] R. R.-F. Liao and A. T. Campbell, "A utility-based approach for quantitative adaptation in wireless packet networks," *Wireless Networks*, vol. 7, pp. 541–557, Sept. 2001.

[13] V. Bharghavan, K.-W. Lee, S. Lu, S. Ha, J.-R. Li, and D. Dwyer, "The timely adaptive resource management architecture," *Personal Communications, IEEE*, vol. 5, no. 4, pp. 20–31, aug 1998.

[14] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, 1998.

[15] Z. Fang and B. Bensaou, "Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks." in *INFOCOM*, 2004.

[16] C. Lee, "On quality of service management," Ph.D. dissertation, Carnegie Mellon University, Aug. 1999, technical Report CMU-CS-99-165.

[17] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*. Boston: Kluwer Academic Publishers, 1996.

[18] K. Römer, P. Blum, and L. Meier, "Time synchronization and calibration in wireless sensor networks," in *Handbook of Sensor Networks: Algorithms and Architectures*, I. Stojmenovic, Ed. John Wiley, Sept. 2005, to appear.

[19] C. Oliveira, J. B. Kim, and T. Suda, "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 858–878, Aug. 1998.

[20] http://opsresearch.com/OR Objects/index.html, "Or-objects homepage."