A Joint Congestion Control, Routing, and Scheduling Algorithm in Multihop Wireless Networks with Heterogeneous Flows

Phuong Luu Vo, Nguyen H. Tran, Choong Seon Hong, *KiJoon Chae
Kyung Hee University, *Ewha Womans University, Korea
{phuongvo, nguyenth, cshong}@khu.ac.kr, *kjchae@ewha.ac.kr

***Abstract*—We consider the network with two kinds of traffic: inelastic and elastic traffic. The inelastic traffic requires fixed throughput, high priority while the elastic traffic has controllable rate and low priority. Giving the fixed rate of inelastic traffic, how to inject the elastic traffic into the network to achieve the maximum utility of elastic traffic is solved in this paper.**

The Lagrangian Duality method is applied to solve the optimization problem. We decompose the Lagrangian into sub-problems, and each sub-problem associates with each layer. The convexity of the primal problem guarantees the duality gap between primal and dual solutions is zero. The Lagrange multipliers, which are indeed the queue length on nodes for every destinations, implicitly update according to subgradient algorithm. The joint algorithm for rate control, routing, and scheduling is proposed. However, the scheduling is Max-weight scheduling and centralized algorithm actually. The Greedy distributed scheduling is introduced to implement scheduling in a distributed sense.

I. INTRODUCTION

Network optimization and control is an active research area [1]–[4]. There are two main kinds of network formulations: node-centric and link-centric [3]. The node-centric formulation maximizes the utility such that all queues are stable: total transmitting rate and incoming flow of a queue must less than the outgoing flow. On the other hand, the link-centric formulation uses the capacity constraint: total of load of all flows on a link must be less than the capacity of the link. Whereas link-centric formulation need a routing matrix in the formulation, the node-centric does not; therefore, the dynamic routing is also solved in the node-centric formulation.

Recent papers apply the frameworks for networks with heterogeneous flows: elastic and inelastic flows [5]–[8]. By using the link-centric formulation, giving the inelastic rate, the authors in [6] maximize the utility of elastic traffic while load-balancing the inelastic traffic on some predefine routes. Also using the link-centric formulation, the optimization problem is solved in [7] with the additional constraint: the probability of missing the deadline packets less than a threshold. All these papers use link-centric formulation, therefore, the routing matrix must be a priori.

This work was supported by the IT R&D program of MKE/KEIT [10035245: Study on Architecture of Future Internet to Support Mobile Environments and Network Diversity].

Dr. CS Hong is the corresponding author.

Our paper applies the node-centric formulation to solve the problem, so the dynamic routing is integrated naturally. We don't cover the end-to-end delay constraint in the scope, but the priority of inelastic traffic in using the links in wireless environment is considered. The rate of inelastic traffic is fixed (the source demand). We want to inject the elastic traffic such that maximizing the utility of elastic traffic while all the queues in the network keep stable. Our contributions in this paper are:

1) Applying the node-centric formulation in the cross-layer design for the multihop wireless network with the requirement of fixed the inelastic rate (source demand) and optimal utility of elastic traffic.
2) Proposing the rate control for elastic traffic, the distributed routing and scheduling for both kinds of traffic.
3) Providing the simulation of the impact of higher priority of inelastic rate demand on the lower priority elastic traffic.

II. PROBLEM FORMULATION

The network is modeled by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes, and $\mathcal{L}$ is the set of links. The network has two kinds of flows: inelastic flows and elastic flows.

Define $\mathcal{F}_i$ as the set of inelastic flows. Each inelastic flow f_i maps a pair of two nodes: source node and destination node. Let $\mathcal{S}_i$ and $\mathcal{D}_i$ be the sets of inelastic source nodes and destination nodes respectively. We have $\mathcal{S}_i \subset \mathcal{N}$ and $\mathcal{D}_i \subset \mathcal{N}$. Set x_i is the rate of flow f_i. x_i just depends on the demand of multimedia service, and it is a constant. We assume that the inelastic rate is always *admissible* by the network.

Define $\mathcal{F}_e$ is the set of elastic flows. Each elastic flow f_e maps a pair of two nodes: source node and destination node. Define $\mathcal{S}_e$ and $\mathcal{D}_e$ as the sets of elastic source nodes and destination nodes respectively. We also have $\mathcal{S}_e \subset \mathcal{N}$, $\mathcal{D}_e \subset \mathcal{N}$. In this paper, we assume that the sets of of destination nodes of inelastic and elastic flows are disjoint.

Each elastic flow is associated with a utility function $U(.)$, which is concave, twice-differentiable, and non-decreasing. For example, $\log(x)$ where x is the rate of the flow is a utility function. x_e is the rate of flow f_e, and $\mathbf{x}^{(e)}$ is the rate vector of all elastic flows. Here we want to control the rate x_e to archive the maximum total of utility.

Link-Rate Region

978-1-61284-663-7/11/$26.00 ©2011 IEEE

A scheduling policy is a set of links that can be activated simultaneously in a wireless network. A link rate vector is called *obtainable* if it can be achieved simultaneously. Define the Γ_h, where $h \in \mathcal{H}$ is the set of all scheduling policies, as the set of all link rate vectors $\mathbf{f}$ under a scheduling policy h. Γ_h and the union of all Γ_h, $h \in \mathcal{H}$ can be non-convex. Denote the *link-rate region* Π as the convex-hull of all Γ_h

$$\Pi = \left\{ \mathbf{f} : \mathbf{f} = \sum_{h \in \mathcal{H}} a_h \mathbf{r}_h, \forall a_h \geq 0, \mathbf{r}_h \in \Gamma_h, \text{and } \sum_s a_s = 1 \right\}.$$

A link-rate vector $\mathbf{f}$ is *feasible* if and only if it is in the link-rate region Π.

$$\mathbf{f} \in \Pi \tag{1}$$

Note that a feasible link-rate vector can be achieved by time-sharing technique.

Flow conservation

We use node-centric formulation, which each node maintains a separate queue for each destination. Denote $f_{ij}^{d_i}$ be the rate allocated on link (i, j) for the inelastic destination d_i. So total rate allocated on one link for all inelastic destination is $f_{ij}^{(i)} = \sum_{d_i \in \mathcal{D}_i} f_{ij}^{d_i}$. $\sum_{j:(i,j)\in\mathcal{L}} f_{ij}^{d_i}$ is total outgoing packet for destination d_i at node i in one time-slot. $\sum_{j:(j,i)\in\mathcal{L}} f_{ji}^{d_i}$ is total incoming packet for destination di at node i in one time-slot.

In order to each queue of each node stables, the total number of incoming traffic and number of packets injected in to the network must less than total number of outgoing traffic in one time-slot for every queues on every nodes

$$x_i + \sum_{j:(j,i)\in\mathcal{L}} f_{ji}^{d_i} \leq \sum_{j:(i,j)\in\mathcal{L}} f_{ij}^{d_i} \quad \forall i \in \mathcal{N}, d_i \in \mathcal{D}_i. \tag{2}$$

(If node i is in $\mathcal{S}_i$ then x_i is a positive constant, else x_i equals zero.)

Similarly, the constraint for stability of queues of elastic traffic is given by the inequality

$$x_e + \sum_{j:(j,e)\in\mathcal{L}} f_{je}^{d_e} \leq \sum_{j:(e,j)\in\mathcal{L}} f_{ej}^{d_e} \quad \forall e \in \mathcal{N}, d_e \in \mathcal{D}_e. \tag{3}$$

If node e is in $\mathcal{S}_e$ then x_e is positive, else x_e equals zero.

Note that the link-rate $\mathbf{f}$ and user-rate vector $\mathbf{x}$ are difference. Link rate is the transmitting rate that is allocated on the links for endogenous traffic, while user-rate is the rate of the exogenous traffic injected to the system at nodes.

The network *capacity region* Λ is the set of all user-rate vector $\mathbf{x}$, such that there exists a link-rate vector $\mathbf{f}$ that satisfying (1), (2), and (3). It's easily to see that Λ is a convex set.

Time-varying channel

In the case of time-varying channel, the capacity of the links changes from time-slot to time-slot because of the fading or the mobility of nodes. Assume that the number of channel states is finite. The problem will be solved in a similar way to the fixed channel state problem, but in this case, we redefine the sets Γ and Π by the average of them over all channel states

$$\overline{\Gamma} = \sum_{c \in \mathcal{C}} p_c \Gamma_c \quad \text{and} \quad \overline{\Pi} = \sum_{c \in \mathcal{C}} \mathrm{p_c} \Pi_\mathrm{c},$$

where $\mathcal{C}$ is the set of all channel states and p_c is the probability of channel state c.

Just for simplicity in writing, we use only Π to denote the link-rate region for both fixed channel case and time-varying channel case. In the time-varying channel case, we implicitly understand that it is the average of the regions over all states.

Primal problem

Our objective is maximizing the total utility of elastic traffic such that guarantee the flow conservation constraints (2),(3) and the schedulability (1)

$$\begin{aligned}
\text{Max.} \; & \sum_{e \in \mathcal{S}_e} U(x_e) \\
\text{s.t.} \; & x_i + \sum_{j:(j,i)\in\mathcal{L}} f_{ji}^{d_i} \leq \sum_{j:(i,j)\in\mathcal{L}} f_{ij}^{d_i}, \; \forall i \in \mathcal{N}, d_i \in \mathcal{D}_i \\
& x_e + \sum_{j:(j,e)\in\mathcal{L}} f_{je}^{d_e} \leq \sum_{j:(e,j)\in\mathcal{L}} f_{ej}^{d_e}, \; \forall e \in \mathcal{N}, d_e \in \mathcal{D}_e \\
& \mathbf{f} \in \Pi.
\end{aligned} \tag{4}$$

The primal problem is a convex problem because we maximize a concave function with the feasible region is a convex set ((2) and (3) are affine, and Π is convex).

III. Solution analysis

It is difficult to solve the primal problem directly. We use the Lagrange dual method and decompose the problem into sub-problems. The decomposition also helps to breakdown the problem into function of layers and implement the distributed algorithm.

Lagrangian:

$$\begin{aligned}
& L(\mathbf{x}^{(e)}, \mathbf{f}, \boldsymbol{\lambda}) = \sum_{e \in \mathcal{S}_e} U(x_e) \\
& - \sum_{i \in \mathcal{N}, d_i \in \mathcal{D}_i} \lambda_i^{d_i} \left(x_i + \sum_{j:(j,i)\in\mathcal{L}} f_{ji}^{d_i} - \sum_{j:(i,j)\in\mathcal{L}} f_{ij}^{d_i} \right) \\
& - \sum_{e \in \mathcal{N}, d_e \in \mathcal{D}_e} \lambda_e^{d_e} \left(x_e + \sum_{j:(j,e)\in\mathcal{L}} f_{je}^{d_e} - \sum_{j:(e,j)\in\mathcal{L}} f_{ej}^{d_e} \right) \\
= & \sum_{e \in \mathcal{S}_e} [U(x_e) - \lambda_e^{d_e} x_e] \\
& + \sum_{e \in \mathcal{N}, d_e \in \mathcal{D}_e} \lambda_e^{d_e} \left(\sum_{j:(e,j)\in\mathcal{L}} f_{ej}^{d_e} - \sum_{j:(j,e)\in\mathcal{L}} f_{je}^{d_e} \right) \\
& + \sum_{i \in \mathcal{N}, d_i \in \mathcal{D}_i} \lambda_i^{d_i} \left(\sum_{j:(i,j)\in\mathcal{L}} f_{ij}^{d_i} - \sum_{j:(j,i)\in\mathcal{L}} f_{ji}^{d_i} \right) \\
& - \sum_{i \in \mathcal{S}_i, d_i \in \mathcal{D}_i} \lambda_i^{d_i} x_i,
\end{aligned} \tag{5}$$

where $\lambda_e^{d_e}$ and $\lambda_i^{d_i}$ are Lagrange multipliers associated with each destination on each node, and $\boldsymbol{\lambda}$ is the vector of all multipliers. We can interpret the Lagrange multiplier as the price the user must pay if they want to inject a traffic flow into the network.

Dual problem

$$\min_{\boldsymbol{\lambda}\succeq \mathbf{0}} D(\boldsymbol{\lambda}), \tag{6}$$

where $D(\boldsymbol{\lambda}) = \max_{\mathbf{x}^{(e)}\succeq 0, \mathbf{f}\in\Pi} L(\mathbf{x}^{(e)}, \mathbf{f}, \boldsymbol{\lambda})$.

The dual problem is always a convex problem. Because of the convexity of the primal problem, the duality gap between the primal and dual problem is zero, i.e. the optimal solution of dual co-insides with the optimal solution of primal [9]. By using sub-gradient algorithm with constant step-size, we can find the optimal solution of dual problem. It's easily to check that $(x_e + \sum_{j:(j,e)\in\mathcal{L}} f_{je}^{d_e} - \sum_{j:(e,j)\in\mathcal{L}} f_{je}^{d_e})$, and $(x_i + \sum_{j:(j,i)\in\mathcal{L}} f_{ji}^{d_i} - \sum_{j:(i,j)\in\mathcal{L}} f_{ij}^{d_i})$ are subgradients of Lagrangian.

Price update is given by

$$\lambda_e^{d_e}(t+1) = \Bigg[\lambda_e^{d_e}(t) + k_e\Bigg(x_e^* + \sum_{j:(j,e)\in\mathcal{L}} f_{je}^{d_e\,*} - \sum_{j:(e,j)\in\mathcal{L}} f_{je}^{d_e\,*}\Bigg)\Bigg]^+ \tag{7}$$

$$\lambda_i^{d_i}(t+1) = \Bigg[\lambda_i^{d_i}(t) + k_i\Bigg(x_i + \sum_{j:(j,i)\in\mathcal{L}} f_{ji}^{d_i\,*} - \sum_{j:(i,j)\in\mathcal{L}} f_{ji}^{d_i\,*}\Bigg)\Bigg]^+, \tag{8}$$

where k_i and k_e are positive constant step-sizes that are small enough to have the convergence of the algorithm; $(a)^+ = \max(a, 0)$. We can consider λ_i^d as the queue size on node i for destination d. We can see that the evolution of the multipliers in each step are proportional to the queue evolution in each time slot. If all queues in the network are empty initially, then we can think of the multipliers vector $\boldsymbol{\lambda}(t)$ representing the size of all queues at the time t, $\mathbf{q}(t)$. Actually, $\mathbf{q}(t) = \boldsymbol{\lambda}(t)/\text{stepsize}$.

The optimal values $(\mathbf{f}^*, \mathbf{x}^{(e)*}) = \arg\max L(\mathbf{x}^{(e)}, \mathbf{f})$ given λ in each iteration

$$\begin{aligned}(\mathbf{x}^{(e)*}, \mathbf{f}^*) = \arg\max\nolimits_{\mathbf{x}^{(e)}\succeq 0, \mathbf{f}\in\Pi}\Bigg[&\sum_{e\in\mathcal{S}_e}(U_e(x_e) - \lambda_e^{d_e}x_e)\\ &+ \sum_{e\in\mathcal{N}, d_e\in\mathcal{D}_e}\lambda_e^{d_e}\Bigg(\sum_{j:(e,j)\in\mathcal{L}} f_{ej}^{d_e} - \sum_{j:(j,e)\in\mathcal{L}} f_{je}^{d_e}\Bigg)\\ &+ \sum_{i\in\mathcal{N}, d_i\in\mathcal{D}_i}\lambda_i^{d_i}\Bigg(\sum_{j:(i,j)\in\mathcal{L}} f_{ij}^{d_i} - \sum_{j:(j,i)\in\mathcal{L}} f_{ji}^{d_i}\Bigg)\\ &- \sum_{i\in N, d_i\in\mathcal{D}_i}\lambda_i^{d_i}x_i\Bigg]\end{aligned} \tag{9}$$

$$= \sum_{e\in S_e}\arg\max_{x_e\geq 0}[U_e(x_e) - \lambda_e^{d_e}x_e] \tag{10}$$

$$+ \arg\max_{f\in\Pi}\Bigg[\sum_{e\in N, d_e\in\mathcal{D}_e}\lambda_e^{d_e}\Bigg(\sum_{j:(e,j)\in\mathcal{L}} f_{ej}^{d_e} - \sum_{j:(j,e)\in\mathcal{L}} f_{je}^{d_e}\Bigg) + \sum_{i\in\mathcal{N}, d_i\in\mathcal{D}_i}\lambda_i^{d_i}\Bigg(\sum_{j:(i,j)\in\mathcal{L}} f_{ij}^{d_i} - \sum_{j:(j,i)\in\mathcal{L}} f_{ji}^{d_i}\Bigg)\Bigg]. \tag{11}$$

Solving the first sub-problem (10) yielding the **Rate Control** scheme is

$$x_e^* = U'^{-1}(\lambda_e^{d_e}) \quad \forall e\in\mathcal{S}_e. \tag{12}$$

The second subproblem (11) relates to the scheduling and routing. The solution of (11) helps to determine the optimal scheduling policy and the optimal destination to transmit the packets to in each time-slot.

Because of the assumption $\mathcal{D}_i$ and $\mathcal{D}_e$ are disjoint, we can define $\mathcal{D} = \mathcal{D}_i + \mathcal{D}_e$ as the set of destination nodes for both kinds of flows. The scheduling and routing problem (11) can be rewritten

$$\mathbf{f}^* = \arg\max_{\mathbf{f}\in\Pi}\sum_{k\in\mathcal{N}, d\in\mathcal{D}}\lambda_k^d\Bigg(\sum_{j:(k,j)\in\mathcal{L}} f_{kj}^d - \sum_{j:(j,k)\in\mathcal{L}} f_{jk}^d\Bigg)$$

Note that

$$\sum_{k\in\mathcal{N}, d\in\mathcal{D}}\lambda_k^d\Bigg(\sum_{j:(k,j)\in\mathcal{L}} f_{kj}^d - \sum_{j:(j,k)\in\mathcal{L}} f_{jk}^d\Bigg) = \sum_{(i,j)\in\mathcal{L}, d\in\mathcal{D}} f_{ij}^d(\lambda_i^d - \lambda_j^d)$$

Hence,

$$\begin{aligned}\mathbf{f}^* &= \arg\max_{\mathbf{f}\in\Pi}\Bigg(\sum_{(i,j)\in\mathcal{L}, d\in\mathcal{D}} f_{ij}^d(\lambda_i^d - \lambda_j^d)\Bigg)\\ &= \arg\max_{\mathbf{f}\in\Pi}\Bigg(\sum_{(i,j)\in\mathcal{L}} f_{ij}^d\max_{d\in\mathcal{D}}(\lambda_i^d - \lambda_j^d)\Bigg)\end{aligned} \tag{13}$$

Defining the weight of the link $w_{(i,j)} = \max_{d\in\mathcal{D}}(\lambda_i^d - \lambda_j^d) = \lambda_i^{d^*} - \lambda_j^{d^*}$, where $d^* = \arg\max_{d\in\mathcal{D}}(\lambda_i^d - \lambda_j^d)$ as the maximum of the differential queue size over all the destination of each link (i, j).

The **scheduling** problem becomes the Max-weight scheduling: choosing the scheduling policy that has the maximum total of weight of all active links:

$$\mathbf{f}^* = \arg\max_{\mathbf{f}\in\Pi}\sum_{l\in\mathcal{L}} f_l w_l \tag{14}$$

Routing: over link l, send an amount of bits for destination d^* with the maximum rate.

IV. Performance analysis

A. Convergence analysis

***Proposition** 1:* The duality gap between Primal and Dual is zero. Moreover, the optimal solutions of Primal and Dual $(\boldsymbol{\lambda}^{(opt)}, \mathbf{x}^{(e)(opt)})$ are also global optimal and unique .

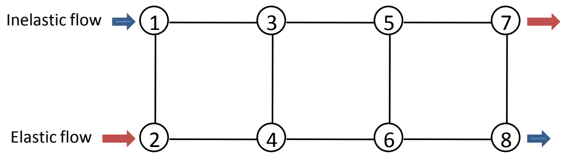

Fig. 1. The simulation network with two flows. All links are bidirectional.

Proof: Because the primal problem is a convex optimization problem (maximizing the strictly concave function with the feasible set Λ is a convex set), the Slater's condition holds. So we have the Proposition 1. ■

Proposition 2: The price evolution described by (7) and (8) are positive recurrent Markov chains. As the result, queues in the network are stable.

Proof: The interested readers can find the proof in previous works [1], [10]. ■

The rate $\mathbf{x}^{(e)}$ is determined by $\boldsymbol{\lambda}$, (12). As the result of the proposition 2, we also have the elastic rate evolution is a positive recurrent Markov chain; therefore, it is stable.

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} E[x_e(t)] = x_e^{(opt)} + O(k_e),$$

where $x_e^{(opt)}$ is the optimal solution of the Primal, k_e is the step-size of the subgradient algorithm.

B. Distributed algorithm

The rate control is implemented in distributed sense because the user rate is calculated by the queue size at the source node to update the transmission rate. However, the Max-weight scheduling requires the knowledge of not only weight of all links in network, but also all the possible scheduling policies in each calculation at each time-slot. Hence, the network need the centralized computation and the updates of the link weight information will overhead the network.

We would like to implement the algorithm that just based on some local information. The following distributed scheduling is utilized from a result in the paper of Preis [11].

1) One node chooses the neighbor with the maximum weight, and active the link connecting them.
2) Remove two nodes, the links connecting them, and all the interference links of two above nodes from graph.
3) Choosing another node and repeating the steps 1 and 2 until all the nodes are removed. The new link is activated only if the total weight of all activated links is increased.

The above algorithm is actually implemented in the distributed sense. All the information the node need is the weight of its neighbors. So that weight of all nodes is sent to its neighbors instead of broadcasting to all the networks.

V. Simulation Results

The purpose of the simulation is to understand the relation between the inelastic flows and elastic flow. How the source demand impacts on the rate of elastic flow.

We use the grid topology for our simulation. The graph includes 8 nodes and 10 links (Figure 1). All the links are bidirectional. We use the Node Exclusive Interference model: if two links share a same node, they cannot active at once time. We consider two flows in the network. The flow 1 from node 1 to node 8 is inelastic flow with the rate is constant. The flow 2 from node 2 to node 7 is elastic flow with the rate can be control. For simplicity, we use the capacity of all the links are constant and equal to 10 Mbps. The utility function for the elastic flow is $U(x) = \log(x)$.

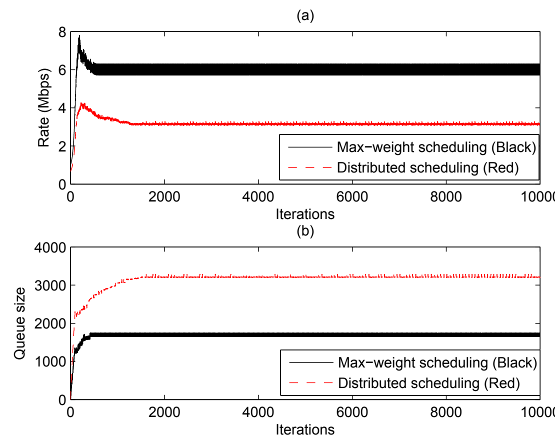

Fig. 2. The convergence of rate and queues when $x_i = 4$, (a) Inelastic rate (b) Total queue size

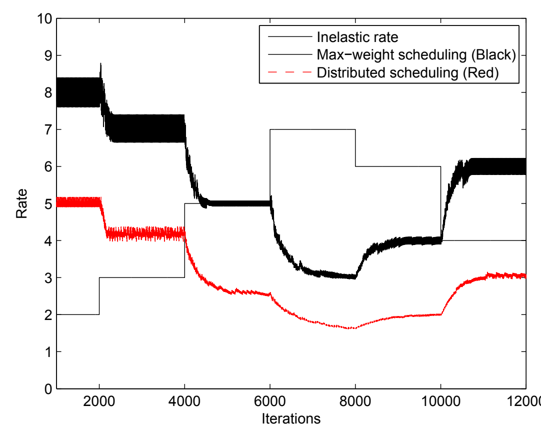

Fig. 3. The elastic rate when changing the source demand

The Figure 2 is the plots of the elastic rate and total size of all queue with respect to Max-weight scheduling and Distributed scheduling. The number of iterations is 10000. We can see that the rate become convergence in the long time. The distributed scheduling always yields the lower rate than the Max-weight scheduling because the capacity region of the distributed scheduling is always smaller than the capacity region of Max-weight scheduling, and it is also not the optimal rate. We can see the rate and the price vary from time-slot to time-slot. The reason is we use the sub-gradient method with constant step-size for the algorithm. The smaller of the step-size, the closer of the rate to the optimal value.

TABLE II

THE AVERAGE LINK RATES OF FLOW 2 ($x_i = 4$).

Links	Max-weight	Distributed	Links	Max-weight	Distributed
(1,3)	3	3	(3,1)	0	0
(3,5)	2.9999	2.9999	(5,3)	0	0
(5,7)	3	2.9999	(7,5)	0	0
(1,2)	0	0	(2,1)	3.0002	3.0003
(3,4)	0	0	(4,3)	0	0
(5,6)	0	0	(6,5)	0.0001	0.0002
(7,8)	0	0	(8,7)	3.0002	0.0001
(2,4)	3.0005	0.0008	(4,2)	0	0
(4,6)	3.0004	0.0006	(6,4)	0	0
(6,8)	3.0003	0.0002	(8,6)	0	0

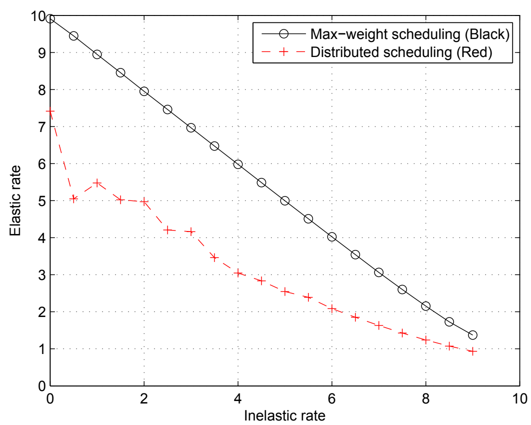

Fig. 4. The elastic rate v.s. inelastic rate.

TABLE I

THE AVERAGE LINK RATES OF FLOW 1 ($x_i = 4$).

Links	Max-weight	Distributed	Links	Max-weight	Distributed
(1,3)	2.0001	1.9986	(3,1)	0	0
(3,5)	2	1.9982	(5,3)	0	0
(5,7)	1.9999	0.0009	(7,5)	0	0
(1,2)	1.9998	2.0011	(2,1)	0	0
(3,4)	0	0.0001	(4,3)	0	0
(5,6)	0.0001	1.9971	(6,5)	0	0
(7,8)	1.9998	0.0008	(8,7)	0	0
(2,4)	1.9996	2.0008	(4,2)	0	0
(4,6)	1.9995	2.0007	(6,4)	0	0
(6,8)	1.9996	3.9978	(8,6)	0	0

The elastic rate decreases when the inelastic rate increase (Figure 4). The reason is the inelastic and elastic flows share some same links in difference time-slot (tables I and II), so the performance of elastic flow will be impact when increase the source demand. The inelastic always has the higher priority than the elastic traffic.

VI. CONCLUSION

We have presented the framework of cross-layer design using node-centric formulation for multi-hop wireless networks with both inelastic and elastic traffic. By using the Duality method, the rate control, routing and scheduling problems are decomposed. Our solution not only maximizes the utilization of the capacity, but also guarantees the fairness of the elastic flows. The simulation results show the impact of the inelastic rate demand on the elastic traffic.

REFERENCES

[1] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, april 2006, pp. 1 –13.

[2] X. Lin and N. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 2, 14-17 2004, pp. 1484 – 1489 Vol.2.

[3] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and mac for stability and fairness in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 8, pp. 1514 –1524, aug. 2006.

[4] S. Shakkottai and R. Srikant, "Network optimization and control," *Found. Trends Netw.*, vol. 2, no. 3, pp. 271–379, 2007.

[5] Y. Li, M. Chiang, A. Calderbank, and S. Diggavi, "Optimal rate-reliability-delay tradeoff in networks with composite links," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 6-12 2007, pp. 526 –534.

[6] R. Li, L. Ying, A. Eryilmaz, and N. Shroff, "A unified approach to optimizing performance in networks serving heterogeneous flows," in *INFOCOM 2009, IEEE*, 19-25 2009, pp. 253 –261.

[7] J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," in *INFOCOM, 2010 Proceedings IEEE*, 14-19 2010, pp. 1 –9.

[8] U. Javed, M. Suchara, J. He, and J. Rexford, "Multipath protocol for delay-sensitive traffic," in *Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International*, 5-10 2009, pp. 1 –8.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[10] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 396–409, 2008.

[11] R. Preis, "Linear time 1/2 -approximation algorithm for maximum weighted matching in general graphs," in *STACS'99: Proceedings of the 16th annual conference on Theoretical aspects of computer science*. Berlin, Heidelberg: Springer-Verlag, 1999, pp. 259–269.