

The Cross-layer Design Approach for Multi-hop Wireless Networks with Elastic and Inelastic Traffic

*Phuong Luu Vo, *Nguyen H. Tran, **Jin Woong Cho, **Lee Jang Yeon, *Choong Seon Hong

*Networking Lab, Department of Computer Engineering, Kyung Hee University, **Korea Electronics Technology Institute

Email: {phuongvo, nguyenth}@khu.ac.kr

{chojw, jylee136}@keti.re.kr and cshong@khu.ac.kr

Abstract—We consider the network with two kinds of traffic: inelastic and elastic. The inelastic traffic requires fixed throughput, higher priority while the elastic traffic has the rate that can be controlled and lower priority. Given 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.

We apply the cross-layer design framework in the multi-hop wireless environment. By using node-centric formulation, the congestion control, dynamic routing and scheduling is implemented.

I. INTRODUCTION

In the last several years, optimization is very useful technique for researching on network, especially in resource allocation and cross-layer design. Two main kinds of formulation in cross-layer design framework are node-centric and link-centric [1]–[4]. In the node-centric formulation, network utility is maximized such that all queues are stable. On the other hand, the constraint of link-centric formulation is: total of load of all flows on a link must be less than the capacity of that link. While link-centric formulation needs a routing matrix as a priori, the node-centric doesn't; so, the dynamic routing is also solved in node-centric formulation.

Recent year, some authors have applied the above frameworks for the network with heterogeneous traffic: elastic and inelastic [5]–[7]. Most of them used the link-centric formulation, so the routing is not a part in the framework. Our paper would like to apply the node-centric formulation to solve the problem, hence, the dynamic routing is integrated naturally. We don't cover the end-to-end delay constraint in the scope. Instead, the priority of inelastic traffic in using the links in wireless environment is considered. The rate of inelastic traffic is fixed, sometimes it is called *source demand*. We want to inject the elastic traffic such as maximizing the utility of elastic traffic subject to all the queues in network keep stable.

The following is the structure of the paper: The formulation and optimization problem are described at section 2; the section 3 presents the algorithm that gives optimal solution; section 4 briefly introduces the distributed scheduling; and finally, the simulation results are in section 5.

II. PROBLEM FORMULATION

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

Define \mathcal{F}_i is the set of inelastic flows. Each inelastic flow f_i maps a pair of two nodes: source node and destination node. Define \mathcal{S}_i and \mathcal{D}_i are 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. In this paper, 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 are 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 destination nodes of inelastic and elastic flows are disjoint.

Each elastic flow is associated with a utility function $U(\cdot)$, which is concave, 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 have maximum total of utility.

Link-Rate Region

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 $h \in \mathcal{H}$: the set of all scheduling policies, is 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. Define the *link-rate region* Π is 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 called *feasible* if and only if it is in the link-rate region Π .

$$\mathbf{f} \in \Pi \quad (1)$$

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

Flow conservation

In this paper, we use node-centric formulation, where each node maintains a separate queue for each destination. Denote $f_{ij}^{d_i}$ is 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 $f_{ij}^{(i)} = \sum_{d_i \in \mathcal{D}_i} f_{ij}^{d_i} \cdot \sum_{j: (i,j) \in \mathcal{L}} f_{ij}^{d_i}$ 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 d_i 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.

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

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 inequation:

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

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's exist a link-rate vector \mathbf{f} that satisfying (1), (2), and (3). It's easily to see that Λ is a convex set.

Primal problem

Our objective is maximizing the total utility of elastic traffic such that $\mathbf{x} \in \Lambda$.

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

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:

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

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

$$\min_{\boldsymbol{\lambda} \geq \mathbf{0}} D_1(\boldsymbol{\lambda}) \quad (6)$$

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

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, [8]. By using sub-gradient algorithm with constant step-size, we can find the optimal solution of dual problem. It's easily to check that $(x_e + \sum_{j:(j,e) \in \mathcal{L}} f_{je}^{d_e} - \sum_{j:(e,j) \in \mathcal{L}} f_{ej}^{d_e})$, 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 sub-gradients of objective function of Lagrange function.

Price update:

$$\begin{aligned} \lambda_e^{d_e}(t+1) &= \left[\lambda_e^{d_e}(t) + k_e \left(x_e^* \right. \right. \\ &\quad \left. \left. + \sum_{j:(j,e) \in \mathcal{L}} f_{je}^{d_e*} - \sum_{j:(e,j) \in \mathcal{L}} f_{ej}^{d_e*} \right) \right]^+ \end{aligned} \quad (7)$$

$$\begin{aligned} \lambda_i^{d_i}(t+1) &= \left[\lambda_i^{d_i}(t) + k_i \left(x_i \right. \right. \\ &\quad \left. \left. + \sum_{j:(j,i) \in \mathcal{L}} f_{ji}^{d_i*} - \sum_{j:(i,j) \in \mathcal{L}} f_{ij}^{d_i*} \right) \right]^+ \end{aligned} \quad (8)$$

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 $\lambda(t)$ representing the size of all queues at the time t , $\mathbf{q}(t) = \lambda(t)/\text{stepsize}$.

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

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

$$\begin{aligned} &+ \arg \max_{\mathbf{f} \in \Pi} \left[\sum_{e \in \mathcal{N}, d_e \in \mathcal{D}_e} \lambda_e^{d_e} \left(\sum_{j: (e,j) \in \mathcal{L}} f_{ej}^{d_e} - \sum_{j: (j,e) \in \mathcal{L}} f_{je}^{d_e} \right) \right. \\ &\left. + \sum_{i \in \mathcal{N}, d_i \in \mathcal{D}_i} \lambda_i^{d_i} \left(\sum_{j: (i,j) \in \mathcal{L}} f_{ij}^{d_i} - \sum_{j: (j,i) \in \mathcal{L}} f_{ji}^{d_i} \right) \right] \end{aligned} \quad (10)$$

Solving the first sub-problem (9) yielding the **Rate Control** scheme:

$$x_e^* = U'^{-1}(\lambda_e^{d_e}) \quad \forall e \in \mathcal{S}_e \quad (11)$$

The second sub-problem (10) relates to the scheduling and routing. The solution of (10) 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 (10) can be rewritten:

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

Note that

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

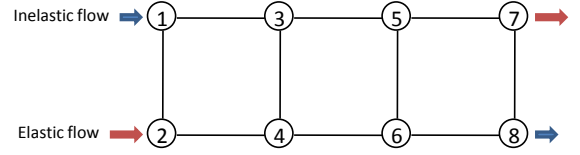


Fig. 1. The simulation network with two flows. All links are bidirectional.

yielding :

$$\begin{aligned} f^* &= \arg \max_{\mathbf{f} \in \Pi} \left(\sum_{(i,j) \in \mathcal{L}, d \in \mathcal{D}} f_{ij}^d (\lambda_i^d - \lambda_j^d) \right) \\ &= \arg \max_{\mathbf{f} \in \Pi} \left(\sum_{(i,j) \in \mathcal{L}} f_{ij}^d \max_{d \in \mathcal{D}} (\lambda_i^d - \lambda_j^d) \right) \end{aligned} \quad (12)$$

Define 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:

$$\mathbf{f}^* = \arg \max_{\mathbf{f} \in \Pi} \sum_{l \in \mathcal{L}} f_l w_l \quad (13)$$

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

IV. DISTRIBUTED ALGORITHM

The rate control is implemented in distributed sense because it 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 weights of all the 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 weight of links will overhead the network.

We would like to implement the algorithm that just bases on some local information. The following distributed scheduling is utilized from [9].

- 1) One node chooses the neighbor with the maximum weight, and active the link connecting them.
- 2) Removing two nodes, the links connecting them and all the interference links of two above nodes from graph.
- 3) Choosing another node and repeat 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 needs is the weights of its neighbors. So that weights 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 flows. 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 bi-directional. 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, and the utility function for the elastic flow is $U(x) = \log(x)$.

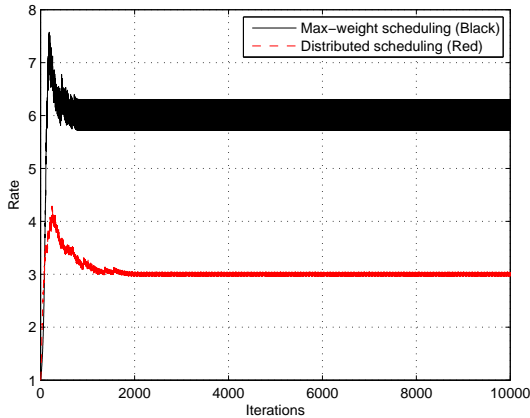


Fig. 2. Elastic rate ($x_i = 4$).

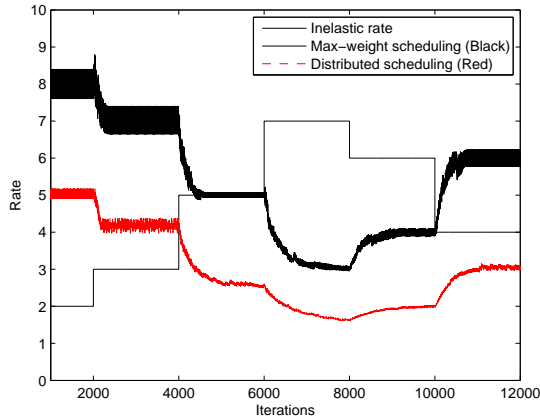


Fig. 3. The elastic rate when changing the source demand

The Figure 2 shows the plots of the rate of elastic flow 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 varies from time-slot to time-slot. The reason is that we used the sub-gradient method with constant step size for the algorithm. The smaller of the step size, the closer of the rate or price to the optimal value. We can see from the Figure 3 that the elastic rate increase when decrease the inelastic rate and vice versa.

VI. CONCLUSIONS

In this paper, we have presented the frame-work 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 is not only maximum the utilizing of the capacity, but also guarantees the fairness of the elastic flows. The simulation has shown the impact of the inelastic rate demand on the elastic traffic.

ACKNOWLEDGMENT

This research is supported by UCN Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy(MKE) in Korea as a result of UCN's subproject 10C2-C2-21T.

Dr. CS Hong is the corresponding author.

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] S. Shakkottai and R. Srikant, "Network optimization and control," *Found. Trends Netw.*, vol. 2, no. 3, pp. 271–379, 2007.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
- [9] 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.