Review of 'Node-Based Optimal Routing in Wireless Networks'

Stephan Günther supervised by Maximilian Riemensberger

1. Introduction

This abstract outlines the fundamental ideas of a distributed routing algorithm to minimize the overall delay in wireless networks suitable for static and quasi-static scenarios [1]. In contrast to centralized approaches each node is capable of making optimal routing decisions on its own based on periodic updates from adjacent nodes. This reduces protocol overhead and removes a single point of failure.

2. Network model

Let a wireless network be represented by a directed acyclic[1] graph $N = (\mathcal{V}, \mathcal{A})$ with a set of nodes $i, k \in \mathcal{V}$ and a set of arcs $(i,k) \in \mathcal{A}$. For a link (i,k) we call k *downstream* of i if k is located closer to the sink. We define $\mathcal{N}(i) := \{k \mid (i,k) \in \mathcal{A}\}$ as the *neighborhood* of i. The flow rate of traffic over link (i,k) destined for j is denoted by $f_{ik}(j)$. At node i, the total traffic

$$t_i(j) = r_i(j) + \sum_{(k,i)\in\mathcal{A}} f_{ki}(j), \ \forall i,j \in \mathcal{V}, \ i \neq j \quad (1)$$

towards j is comprised of the sum of link flows entering i and additional flow $r_i(j)$ generated at i. Conservation of flow is implicitly ensured if node i forwards all traffic not destined for itself. The total link flow rates

$$f_{ik} = \sum_{j\in\mathcal{V}} f_{ik}(j) \leq c_{ik}, \quad \forall (i,k) \in \mathcal{A} \quad (2)$$

are limited by positive link capacities c_{ik}. These conventions are similar to a classic multi-commodity flow problem except that each node is considered as potential source and sink. We now modify this model by introducing non-negative *routing variables*

$$\phi_{ik}(j) = \frac{f_{ik}(j)}{t_i(j)}, \quad \forall i,j,k \in \mathcal{A}, \ i \neq j. \quad (3)$$

These determine the fraction of the total traffic $t_i(j)$ which is routed over link (i,k). Let $\mathcal{P}_{ij} \subseteq \mathcal{A}$ denote a path from i to j containing the intermediate links. A feasible set ϕ of routing variables has to fulfill the following conditions for all pairs of nodes $\{i,j\} \subseteq \mathcal{V}$:

$$\phi_{ik}(i) = 0, \quad \forall (i,k) \in \mathcal{A}, \quad (4a)$$

$$\sum_{k\in\mathcal{N}(i)} \phi_{ik}(j) = 1, \quad \forall i,j \in \mathcal{V}, \ i \neq j, \quad (4b)$$

$$\exists \mathcal{P}_{ij} \mid \phi_{lm}(j) > 0, \quad \forall (l,m) \in \mathcal{P}_{ij}. \quad (4c)$$

[1] For simplicity we assume a loop-free network. However, the algorithm presented here can be adapted to allow for loops.

Constraint (4a) demands that traffic does not re-enter the network once its destination is reached while (4b) asserts that the whole traffic destined for j is routed by intermediate nodes. The existence of a routing path from i to j is ensured by (4c). It is shown by Gallager in [1] that, given an input set r and a set of routing variables ϕ, (1) has a unique solution for t and that $t_i(j)$ is non-negative and continuously differentiable in r and ϕ.

The expected delay on a link $(i,k) \in \mathcal{A}$ is denoted as function $D_{ik}(f_{ik})$ which is assumed to be convex and increasing in f_{ik}. The problem of minimizing the total cost function D_T with respect to the set of routing variables ϕ is then given as

$$\min_\phi D_T = \min_\phi \sum_{(i,k)\in\mathcal{A}} D_{ik}(f_{ik}), \quad (5)$$
$$\text{s.t. (1), (2), (3) and (4).}$$

We now turn to necessary and sufficient conditions for optimality before the algorithm itself is explained.

3. Optimality conditions

Changes in the network delay D_T are caused by either changes in the input rates r or routing variables ϕ. At first we consider the network delay with respect to input rates as depicted in Figure 1. Here, a small increment ϵ in the input rate $r_i(j)$ at node $i \neq j$ is distributed to its neighbors according to the current values of the routing variables $\phi_{ik}(j)$. This causes an increment in the link delays $D_{ik}(f_{ik})$. Once the fraction $\epsilon \cdot \phi_{ik}(j)$ of additional traffic has reached a neighbor $k \in \mathcal{N}(i)$ it can be equivalently considered as an increase in its input rate $r_k(j)$. This in turn is distributed to neighbors of k until sink j is reached. This leads to a recursive definition of the partial derivatives of D_T with respect to $r_i(j)$:

$$\frac{\partial D_T}{\partial r_i(j)} = \sum_{k\in\mathcal{N}(i)} \phi_{ik}(j) \left[\frac{dD_{ik}}{df_{ik}} + \frac{\partial D_T}{\partial r_k(j)}\right]. \quad (6)$$

We declare $\partial D_T / \partial r_j(j) = 0$ because traffic does not re-enter the network once its destination is reached.

Since the routing algorithm does not control input rates but only routing variables, we next consider a small increment ϵ of the routing variable $\phi_{ik}(j)$ at node $i \neq j$ without respect to (4b). Consequently, traffic on (i,k) is augmented by $\epsilon \cdot t_i(j)$ leading to an increase in both the link delay on (i,k) and the delay induced from node k onwards which is governed by

$$\frac{\partial D_T}{\partial \phi_{ik}(j)} = t_i(j) \cdot \left[\frac{dD_{ik}}{df_{ik}} + \frac{\partial D_T}{\partial r_k(j)}\right]. \quad (7)$$

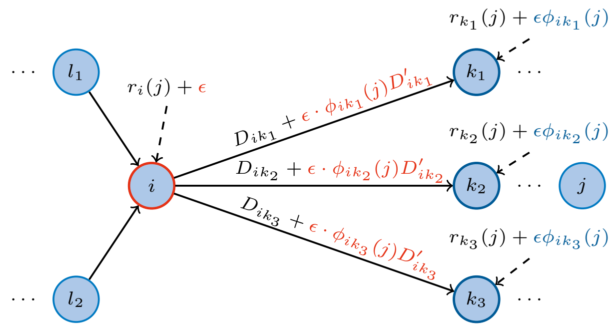

Figure 1: An increment ϵ of the input rate $r_i(j)$ adds a marginal delay to outgoing links and can be modeled from downstream neighbors onwards by a proportional increment of their input rate.

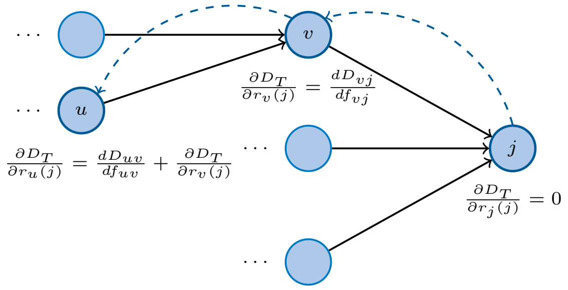

Figure 2: Starting at sink j the derivative $\partial D_T/\partial r_i(j)$ is sent recursively to upstream neighbors enabling them to calculate the gradient of the total delay with respect to a change of input rates.

Theorem: Let D_{ik} be convex and continuously differentiable in f_{ik}. Then, for a given set of input rates r and routing variables ϕ Equation (8) is necessary and (9) is sufficient for optimality at all nodes $i \neq j$.

$$\text{Necessary:} \quad \frac{\partial D_T}{\partial \phi_{ik}(j)} \begin{cases} = \lambda_{ij}, & \phi_{ik}(j) > 0 \\ \geq \lambda_{ij}, & \phi_{ik}(j) = 0 \end{cases} \tag{8}$$

$$\text{Sufficient:} \quad \frac{\partial D_T}{\partial r_i(j)} \leq \frac{dD_{ik}}{df_{ik}} + \frac{\partial D_T}{\partial r_k(j)} \tag{9}$$

The necessary condition follows by incorporating the constraints (4b) and $\phi_{ik}(j) \geq 0$ into (7) by means of Lagrangian multipliers. The insight is that at an optimal point the cost in terms of delay for sending additional traffic must be equal on all outgoing links of i if the corresponding routing variable on that link is positive. If, however, the routing variable is zero, the increase in cost might be arbitrary large but must not be smaller than on any other link originating at i. Otherwise a better solution could be found by shifting traffic to this link. Unfortunately this condition is not sufficient since it is automatically fulfilled for nodes where $t_i(j) = 0$.

For this reason the sufficient condition which can be derived from (6) using (4b) does not depend on $t_i(j)$. It states that the solution is optimal if the marginal delays are the same for all outgoing links which carry traffic for a specific sink ($\phi_{ik}(j) > 0$). This part of the theorem is proofed in [1].

4. Routing algorithm

The iterative algorithm presented here can be divided into two phases. For phase 1, assume a situation with a single sink j as depicted in Figure 2. Each node can average the flow f_{ik} on its outgoing links and therefore estimate both the per link delay D_{ik} and its derivative dD_{ik}/df_{ik}. The algorithm starts at node j by sending a null message to its upstream neighbors since $\partial D_T/\partial r_j(j)$ is zero as agreed upon in Section 3. Let v be an arbitrary upstream neighbor of j. According to the update message received from j, node v can conclude that $\partial D_T/\partial r_v(j) = dD_{vj}/df_{vj}$ which is in turn sent to the upstream neighbors of v. Let u denote such a neighbor. This enables u to calculate $\partial D_T/\partial r_u(j)$ which is sent to its upstream neighbors and so on. This process continues until all nodes received updates from their downstream neighbors which completes phase 1. The extension to multiple sinks follows through superposition of individual flows destined for different destinations.

Phase 2 updates the set of routing variables at each node i. Let

$$m_i := \arg \min_{k \in \mathcal{N}(i)} \left[\frac{dD_{ik}}{df_{ik}} + \frac{\partial D_T}{\partial r_k(j)} \right].$$

On link (i, m_i) an increase of traffic has the smallest influence on the delay of all links leading to neighbors of i. It is therefore the candidate to take over traffic from other links. For all $k \in \mathcal{N}(i)$ we define

$$\begin{aligned} a_{ik}(j) &:= \frac{dD_{ik}}{df_{ik}} + \frac{\partial D_T}{\partial r_k(j)} - \frac{dD_{im_i}}{df_{im_i}} - \frac{\partial D_T}{\partial r_{m_i}(j)}, \\ \Delta_{ik}(j) &:= \min \left(\phi_{ik}(j), \frac{a_{ik}(j)}{t_i(j)} \right). \end{aligned} \tag{10}$$

The set of routing variables $\phi^{(n)}$ in the n-th iteration of the algorithm is then given by

$$\phi_{ik}^{(n)}(j) = \begin{cases} \phi_{ik}^{(n-1)}(j) - \Delta_{ik}(j), & k \neq m_i, \\ \phi_{ik}^{(n-1)}(j) + \sum\limits_{k \neq m_i} \Delta_{ik}(j), & k = m_i. \end{cases} \tag{11}$$

According to (11), flow on link (i, m_i) is increased by the sum by which flows over the other links originating at i are decreased. The decrement $\Delta_{ik}(j)$ is determined by the difference of the delay derivatives of the best and the current link relative to the expected traffic at i. Taking the minimum between this and the current value of $\phi_{ik}(j)$ asserts that the result is non-negative. This algorithm is generalized and extended to include power control and speedup of convergence by Xi and Yeh in [2].

References

[1] Robert Gallager, "A Minimum Delay Routing Algorihtm Using Distributed Computation," in *IEEE Transactions on Communcations*, 1977, vol. 25, pp. 73–85.

[2] Yufang Xi and Admund Yeh, "Node-Based Optimal Power Control, Routing, and Congestion Control in Wireless Networks," in *IEEE Transactions on Information Theory*, 2008, vol. 54, pp. 4081–4106.