View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by HKU Scholars Hub

The HKU Scholars Hub The University of Hong Kong 香港大學學術庫

Title	Optimal rate assignment strategy to minimize average waiting time in wireless networks
Author(s)	Zeng, H; Hou, R; Lui, KS
Citation	The 2011 IEEE Vehicular Technology Conference (VTC Fall), San Francisco, CA., 5-8 September 2011. In IEEEVTS Vehicular Technology Conference Proceedings, 2011, p. 1-5
Issued Date	2011
URL	http://hdl.handle.net/10722/158761
Rights	IEEEVTS Vehicular Technology Conference Proceedings. Copyright © IEEE.

Optimal Rate Assignment Strategy to Minimize Average Waiting Time in Wireless Networks

Hongfei Zeng[a], Ronghui Hou[b], King-Shan Lui[a]
[a]Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong
[b]State Key Lab of Integrated Service Networks, Xidian University, China
Email: hfzeng@eee.hku.hk,rhhou@xidian.edu.cn,kslui@eee.hku.hk

Abstract*—In a wireless network that supports multiple flows, allocation of bandwidth resource among the flows is one of the critical problems. Different allocation strategies have been developed based on different optimization objectives. Unfortunately, these objectives may not reflect directly the time needed for a flow to transmit what it wants. In this paper, we define a new objective,** *average waiting time, that reflects the average time needed for the flows to finish their transmissions. For small networks, we develop an optimal scheme that minimizes the average waiting time. We extend the mechanism for general networks, and simulation results show that it can significantly reduce the average waiting time when compared with other existing mechanisms.**

I. Introduction

When there are multiple flows in a wireless network, due to wireless interference, they compete for network bandwidth. A rate allocation scheme governs how the flow sessions share network bandwidth by specifying the rate of each session. Many rate assignment strategies have been developed [1], [2], [3] based on different optimization objectives.

The classical max-min rate allocation has been widely regarded as an optimal rate allocation policy [4]. It aims to maximize the minimum flow rate of all sessions in network. Let $\{x_1, x_2, x_3, ..., x_n\}$ be the flow rates of all sessions. Max-min allocation means if we increase x_i to x_i', then there definitely exists a x_j, that $x_j \leq x_i$, and x_j is decreased. Proportional fairness [5], on the other hand, tries to maximize total network throughput while providing all users a certain level of fairness. This is done by assigning each session a data rate that is inversely proportional to its anticipated resource consumption. Another strategy is to maximize λ [6] [7] in $f(a) = \lambda t(a)\ \forall a$ where $t(a)$ refers to the traffic demand and $f(a)$ refers to the flow rate of session a.

All these assignment strategies assume all sessions last forever. Nevertheless, such assumptions cannot reflect the increasing popularity of file sharing applications that a session ends when the file download is complete. In this work, we study the flow rate assignment issue when there are two types of sessions: streaming sessions and non-streaming sessions. A streaming session is a long-lived flow that requires a minimum flow rate at all time [8]. Non-streaming sessions reflect file downloading application sessions. These sessions need to deliver a known amount of data, and once all the data have been delivered, the session ends. As long as the flow rate of a streaming session is above or equal to a certain threshold, the user will be satisfied. On the other hand, users of non-streaming sessions would like the downloading to be finished as soon as possible. We thus define our objective as to minimize the average waiting time of non-streaming sessions while the requested minimum data rates are guaranteed for streaming sessions. Our objective directly reflects user experience. We develop a rate allocation algorithm in small wireless networks to achieve this objective and formally prove its optimality. In general networks, we develop a heuristic algorithm and conduct simulations. The results show our algorithm is effective. To the best of our knowledge, we are the first to propose this new objective and perform analytical studies.

The rest of this paper is organized as follows. We define our problem formally in Section II. Our solutions in small and general networks are discussed respectively in Section III and Section IV. Performance evaluation is given in Section V, and we conclude the paper in Section VI.

II. System Model

We model the wireless network as a directed graph $G = (V, E)$, where V is the set of nodes, and E is the set of links. We denote by D_T and D_I the transmission range and interference range, respectively. This paper adopts physical interference model. Let $h(u, v)$ be the Euclidean distance between nodes u and v. When $h(u, v) \leq D_T$, u and v can communicate directly. Thus we say an edge $e = (u, v) \in E$. The capacity of e is $C(e)$. Let $E^-(v)$ and $E^+(v)$ be the sets of incoming and outgoing links in E of node v with $v \in V$. If $D_T < h(u, v) \leq D_I$, u and v cannot communicate directly, but can interfere with each other meaning u has to be silent when v is receiving or sending data or vice versa.

Links $e_1 = (u_1, u_2)$ and $e_2 = (v_1, v_2)$ do not interfere with each other if and only if u_i does not interfere with $v_j, i, j = 1, 2$. We denote by $E^C(e)$ the *set of conflict links of link e*. That is, $E^C(e) = \{e' | e'$ interferes with $e, e' \in E\}$ where $e \in E$.

The network needs to support a number of sessions. We denote the set of all sessions as S. They are divided into Q and A. Q is the set of streaming sessions. A is the set of non-streaming sessions. $S = Q \bigcup A$ and $Q \bigcap A = \emptyset$.

For a session $a \in S$, let $s(a)$ and $d(a)$ be the source node and destination node, respectively. Packets of a go through only a single path from $s(a)$ to $d(a)$. Each session $a \in S$ may go through one-hop or multiple hops. A session $a \in Q$

978-1-4244-8327-3/11/$26.00 ©2011 IEEE

is characterized by a tuple $< s(a), d(a), f_{min}(a) >$, where $f_{min}(a)$ is the minimum flow rate required by a. We assume a streaming session is a long-lived session that lasts forever, and the minimum flow rate must be maintained at all time. A session $a \in A$ is denoted as $< s(a), d(a), k(a) >$, where $k(a)$ refers to the number of bits the session needs to transmit. Unlike a streaming session, a session in A finishes when all its data have been transmitted. In other words, a session in A has end time. We define waiting time of a non-streaming session as the time between request arises and session finishes. We assume all requests arrive at time zero. Then, the waiting time of a session is its end time. We further assume the paths of the sessions are given. Let P_a be the path of session a. We denote by $e \in a$ that link e is a link on path P_a.

We are going to discuss how to determine the flow rates of the sessions at different time such that the average waiting time of the sessions in A is minimized. Formally, let $f(e,t)$ be the flow rate over link e at time t. $f(e,t) \leq C(e), \forall t \in (0,\infty)$. We further let $f(a,t)$ be the flow rate of session a at time t. $f(e,t), f(a,t) \geq 0, \forall t \in (0,\infty)$. We should guarantee

$$f(a,t) \geq f_{min}(a), \forall a \in Q, \forall t \in (0,\infty) \tag{1}$$

For session $a \in A$, let $T_s(a)$ be the time when session a starts sending data and $T_e(a)$ be the time when session a finishes its transmission. We denote the number of sessions in A by N_A. We define

$$T_{end} = \max_{a \in A} T_e(a)$$

and

$$T_{wait} = \frac{\sum_{a \in A} T_e(a)}{N_A}$$

Our objective is to minimize T_{wait} while guaranteeing (1).

We now use an example to illustrate different flow rate assignments would lead to different T_{wait}'s. Consider the simple network in Figure 1 where there are three sessions a, b, and c going from α to β. $C(e = (\alpha, \beta)) = 10$. $Q = \{a\}$ and $A = \{b, c\}$. $a =< \alpha, \beta, 1 >$, $b =< \alpha, \beta, 80 >$ and $c =< \alpha, \beta, 10 >$. We need to guarantee $f(a,t) \geq 1, \forall t \in (0,\infty)$. On the other hand, $f(a,t) + f(b,t) + f(c,t) \leq 10, \forall t \in (0,\infty)$. Note that it is not necessary to assign more rate to a than what it needs. Therefore, we should set $f(a,t)$ to be $f_{min}(a)$ at all time, while assigning the remaining link capacity to other sessions. We now consider three different ways to allocate to b and c and show that T_{wait} would be different in the end.

- In the first way, we allocate bandwidth according to their traffic demand and both b and c start at time 0. Then, $f(b,t) = 8$ and $f(c,t) = 1$. $T_e(b) = \frac{80}{8} = 10s$ and $T_e(c) = \frac{10}{1} = 10s$ as well. It leads to $T_{wait} = 10s$.
- In the second way, we transmit b and then c. b starts at time 0 and have all 9 units for its traffic. It takes $\frac{80}{9}s$ to transmit its data. c starts after b at time $\frac{80}{9}s$. Thus, it will end at $\frac{80}{9}s + \frac{10}{9}s = 10s$. T_{wait} is around $9.5s$.
- In the third way, we transmit c and then b. It is easy to compute that $T_{wait} \simeq 5.5s$.

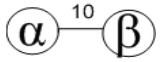

Fig. 1. $C(e = (\alpha, \beta)) = 10. Q = \{a\}. A = \{b, c\}. a =< \alpha, \beta, 1 >, b =< \alpha, \beta, 80 >, c =< \alpha, \beta, 10 >$.

III. Optimal Allocation in small networks

In this section, we are going to describe how to determine the optimal $f(a,t)$ for all $a \in S$ in a small network that all links interfere with each other. Formally, $e_1 \in E^C(e_2), \forall e_1, e_2 \in E$.

We denote $\lambda(t)$ as the total fraction of capacity in the network being used at time t. Since all the links interfere with each other, $\lambda(t) = \sum_{e \in E} \frac{f(e,t)}{C(e)}, \forall t \in (0,\infty)$. It is obvious that

$$\lambda(t) \leq 1, \quad t \in (0,\infty).$$

Note that $f(e,t)$ equals 0 if e does not lie on any path. If e lies on at least one path:

$$f(e,t) = \sum_{a \in S, e \in a} f(a,t)$$

It implies that

$$\begin{aligned}\lambda(t) &= \sum_{e \in E} \frac{f(e,t)}{C(e)} = \sum_{e \in E} \frac{\sum_{a \in S, e \in a} f(a,t)}{C(e)} \\ &= \sum_{a \in S} \left(\sum_{e \in E, e \in a} \frac{1}{C(e)} \right) f(a,t)\end{aligned}$$

Define

$$g(a) = \sum_{e \in E, e \in a} \frac{1}{C(e)}$$

Actually $g(a)$ is the inverse of the maximal flow rate of a when there are no other sessions in the network. Then we get

$$\begin{aligned}\lambda(t) &= \sum_{a \in S} g(a) f(a,t) \\ &= \sum_{a \in Q} g(a) f(a,t) + \sum_{a \in A} g(a) f(a,t).\end{aligned}$$

Define

$$\lambda_A(t) = \sum_{a \in A} g(a) f(a,t)$$

So

$$\begin{aligned}\lambda_A(t) &= \lambda(t) - \sum_{a \in Q} g(a) f(a,t) \\ &\leq 1 - \sum_{a \in Q} g(a) f_{min}(a), \quad \forall t \in (0, T_{end})\end{aligned}$$

The equality holds when

$$\begin{cases} \lambda(t) = 1, & \forall t \in (0, T_{end}) \\ f(a,t) = f_{min}(a), & \forall a \in Q, \forall t \in (0, T_{end}) \end{cases} \tag{2}$$

We know that,

$$\int_0^{T_{end}} \lambda_A(t)dt \le \int_0^{T_{end}} \left(1 - \sum_{a\in Q} g(a)f_{min}(a)\right) dt$$
$$\le T_{end}\left(1 - \sum_{a\in Q} g(a)f_{min}(a)\right)$$

On the other hand,

$$\int_0^{T_{end}} \lambda_A(t)dt = \int_0^{T_{end}} \left(\sum_{a\in A} g(a)f(a,t)\right) dt = \sum_{a\in A} g(a)k(a)$$

Then

$$T_{end} \ge \frac{\sum_{a\in A} g(a)k(a)}{1 - \sum_{a\in Q} g(a)f_{min}(a)}$$

This is the lower bound of T_{end}. The minimum T_{end} can be achieved when (2) holds. (2) holds means that bandwidth is fully utilized and sessions of Q get their required minimum flow rates.

We now study how to minimize T_{wait}. First we sort the sessions in A in non-decreasing order of $g(a)k(a)$, and number them from 1 to N_A. Formally:

$$g(a_i)k(a_i) \le g(a_{i+1})k(a_{i+1}), \quad 1 \le i \le N_A - 1. \quad (3)$$

We also introduce another virtual rank. Assuming we have known $T_e(a)$, we rank the sessions in non-decreasing order of $T_e(a)$. If $T_e(a) = T_e(b), a, b \in A$, a and b are ranked on a random basis. We denote by $N_t(a)$ as the rank number of session a of this ranking. By the definition, we get

$$\|\{a|T_e(a) \ge T_e(a_i), a \in A\}\| \ge N_A + 1 - N_t(a_i), 1 \le i \le N_A$$

The equality holds for all sessions in A if and only if

$$T_e(a_i) \ne T_e(a_j), 1 \le i, j \le N_A, i \ne j \quad (4)$$

Theorem 1: the minimum T_{wait} whose value is $\frac{\sum_{i=1}^{N_A}(N_A+1-i)g(a_i)k(a_i)}{N_A\sum_{a\in Q} g(a)f_{min}(a)}$ can be achieved if:

$$\begin{cases} T_s(a_1) = 0, \quad T_e(a_{N_A}) = T_{end}. & \\ T_s(a_{i+1}) = T_e(a_i), & 1 \le i \le N_A - 1. \\ \lambda(t) = 1, & t \in (0, T_{end}) \\ f(a,t) = f_{min}(a), & t \in (0, T_{end}), \forall a \in Q \end{cases}$$

Proof:

$$T_e(a_i) = \int_0^{T_e(a_i)} \frac{1 - \sum_{a\in Q} g(a)f_{min}(a)}{1 - \sum_{a\in Q} g(a)f_{min}(a)} dt$$
$$\ge \frac{1}{1 - \sum_{a\in Q} g(a)f_{min}(a)} \cdot \int_0^{T_e(a_i)} \lambda_A(t)dt$$

$$\int_0^{T_e(a_i)} \lambda_A(t)dt = \int_0^{T_e(a_i)} \left(\sum_{j=1}^{N_A} g(a_j)f(a_j,t)\right) dt$$
$$= \sum_{j=1}^{N_A} \left(\int_0^{T_e(a_i)} g(a_j)f(a_j,t)dt\right)$$
$$= \sum_{j=1}^{N_A} g(a_j)\left(\int_0^{T_e(a_i)} f(a_j,t)dt\right)$$
$$= \sum_{j=1,T_e(a_i)\ge T_e(a_j)}^{N_A} g(a_j)\int_0^{T_e(a_i)} f(a_j,t)dt +$$
$$\sum_{j=1,T_e(a_i)< T_e(a_j)}^{N_A} g(a_j)\int_0^{T_e(a_i)} f(a_j,t)dt$$
$$= \sum_{j=1,T_e(a_i)\ge T_e(a_j)}^{N_A} g(a_j)k(a_j) +$$
$$\sum_{j=1,T_e(a_i)< T_e(a_j)}^{N_A} g(a_j)\int_0^{T_e(a_i)} f(a_j,t)dt$$

So

$$T_e(a_i) \ge \frac{1}{1 - \sum_{a\in Q} g(a)f_{min}(a)} \cdot \left(\sum_{j=1,T_e(a_i)\ge T_e(a_j)}^{N_A} g(a_j)k(a_j) + \sum_{j=1,T_e(a_i)< T_e(a_j)}^{N_A} g(a_j)\int_0^{T_e(a_i)} f(a_j,t)dt\right) \quad (5)$$

The equality of (5) holds if and only if (2) holds. When the equality of (5) holds,

$$\Big(1 - \sum_{a\in Q} g(a)f_{min}(a)\Big)N_A T_{wait}$$
$$= \Big(1 - \sum_{a\in Q} g(a)f_{min}(a)\Big)\sum_{i=1}^{N_A} T_e(a_i)$$
$$= \sum_{i=1}^{N_A}\left(\sum_{j=1,T_e(a_i)\ge T_e(a_j)}^{N_A} g(a_j)k(a_j) + \sum_{j=1,T_e(a_i)< T_e(a_j)}^{N_A}\int_0^{T_e(a_i)} g(a_j)f(a_j,t)dt\right)$$
$$= \sum_{i=1}^{N_A}\left(\sum_{j=1,T_e(a_i)\ge T_e(a_j)}^{N_A} g(a_j)k(a_j)\right) + \sum_{i=1}^{N_A}\left(\sum_{j=1,T_e(a_i)< T_e(a_j)}^{N_A}\int_0^{T_e(a_i)} g(a_j)f(a_j,t)dt\right)$$
$$= \sum_{j=1}^{N_A}\|\{a_i|T_e(a_i) \ge T_e(a_j), a_i \in A\}\|g(a_j)k(a_j) + \sum_{i=1}^{N_A}\left(\sum_{j=1,T_e(a_i)< T_e(a_j)}^{N_A}\int_0^{T_e(a_i)} g(a_j)f(a_j,t)dt\right)$$
$$\ge \sum_{j=1}^{N_A}\Big(N_A + 1 - N_t(a_j)\Big)g(a_j)k(a_j) +$$

$$\sum_{i=1}^{N_A}\left(\sum_{T_e(a_i)<T_e(a_j)}\int_0^{T_e(a_i)} g(a_j)f(a_j,t)dt\right)$$

To minimize T_{wait}, we have to minimize

$$\sum_{j=1}^{N_A}\Big(N_A+1-N_t(a_j)\Big)g(a_j)k(a_j) \tag{6}$$

and

$$\sum_{i=1}^{N_A}\left(\sum_{T_e(a_i)<T_e(a_j)}\int_0^{T_e(a_i)} g(a_j)f(a_j,t)dt\right) \tag{7}$$

For (6), because:

$$\{N_A+1-N_t(a_j)|1\le j\le N_A\}=\{1,\cdots,N_A\}$$

We should multiply the largest element in $\{1,\cdots,N_A\}$ with the smallest element in $\{g(a_j)k(a_j)|1\le j\le N_A\}$ and vice versa. Combined with (3), (6) is minimized if

$$N_A+1-N_t(a_i)=N_A+1-i,\quad 1\le i\le N_A$$

i.e.

$$N_t(a_i)=i,\quad 1\le i\le N_A$$

This implies

$$T_e(a_{i-1})\le T_e(a_i),\quad 2\le i\le N_A \tag{8}$$

For (7),

$$\sum_{i=1}^{N_A}\left(\sum_{j=1,T_e(a_i)<T_e(a_j)}^{N_A}\int_0^{T_e(a_i)} g(a_j)f(a_j,t)dt\right)\ge 0$$

The equality holds if and only if

$$T_e(a_j)>T_e(a_i)\Rightarrow T_s(a_j)\ge T_e(a_i),\forall a_i,a_j\in A \tag{9}$$

It means if session a_j finishes later than session a_i, then session a_j should not start before session a_i finishes.

When all the above equalities hold, we get:

$$T_{wait}=\frac{\sum_{i=1}^{N_A}(N_A+1-i)g(a_i)k(a_i)}{N_A\left(1-\sum_{a\in Q}g(a)f_{min}(a)\right)}$$

However we have to verify whether those equalities can hold simultaneously. The conditions of all the above equalities are (2),(4),(8),(9). Consider (4) and (8), so $T_e(a_i)<T_e(a_{i+1}), 1\le i\le N_A-1$. Combined with (9), so $T_e(a_i)\le T_s(a_{i+1}), 1\le i\le N_A-1$. If $\exists i, T_e(a_i)<T_s(a_{i+1})$, then $\lambda_A(t)=0, t\in(T_e(a_i),T_s(a_{i+1}))$, which conflicts with (2). So $T_e(a_i)=T_s(a_{i+1}), 1\le i\le N_A-1$. As a result, the conditions are:

$$\begin{cases} T_s(a_1)=0,\quad T_e(a_{N_A})=T_{end}. & \\ T_s(a_{i+1})=T_e(a_i), & 1\le i\le N_A-1. \\ \lambda(t)=1, & t\in(0,T_{end}) \\ f(a,t)=f_{min}(a), & t\in(0,T_{end}),\forall a\in Q \end{cases}$$

Theorem 1 has been proved. ∎

It is worth noting that our algorithm achieves not only minimum T_{wait} but also minimum T_{end}. We can compute $f(a_i,t)$ and $T_s(a_i)$, $T_e(a_i)$, where $1\le i\le N_A$ based on these conditions. Due to space limit, we omit the details.

IV. Optimal Allocation in general networks

Theorem 1 solves the rate assignment problem in small networks where all links interfere with each other. We now develop the optimization problem of a general network.

minimize T_{wait}

subject to:

$$\begin{cases} T_{wait}=\frac{\sum_{a\in A}T_e(a)}{N_A} & \\ \int_0^{T_e(a)}f(a,t)dt=k(a), & \forall a\in A,t\in(0,T_{end}) \\ f(a,t)\ge f_{min}(a), & \forall a\in Q,t\in(0,T_{end}) \\ \sum_{e'\in E^C(e)}\frac{f(e',t)}{C(e')}\le 1, & \forall e\in E,t\in(0,T_{end}) \\ \sum_{e\in E^+(v)}f(e,t)-\sum_{e\in E^-(v)}f(e,t)= & \\ \begin{cases} 0, & \forall v\ne s(a),d(a),\forall a\in S,t\in(0,T_{end}) \\ \sum_{a\in A,s(a)=v}f(a,t)-\sum_{a\in S,d(a)=v}f(a,t), & \\ \qquad\forall v=s(a),d(a),\forall a\in S,t\in(0,T_{end}) \end{cases} & \end{cases}$$

As it is NP-hard to solve this formula, we develop a heuristic algorithm based on the solution of small networks. In a small network, a single session would use up all the bandwidth resources, because all links interfere with each other. In a general network, some links may still be available to carry traffic even though there is an active session. We can then schedule flows on these links without affecting the active sessions.

We first assign as much bandwidth as possible to the session with smallest $k(a)g(a)$, just as what we do in a small network. Then, we assign the remaining bandwidth to other sessions that do not conflict with the existing ones. We should select sessions based on the non-decreasing order of $g(a)k(a)$ to make use of the remaining bandwidth. When a session finishes, we stop all on-going sessions and assign rates again. Note that $k(a)$ of some remaining sessions may have changed, so we should re-rank the remaining sessions.

Details of the algorithm can be found in Algorithms 1 - 3. Algorithm 3 is the overall algorithm. We call this algorithm to compute T_{wait}. It assigns bandwidth to all sessions. First it computes $g(a)$ of all sessions, and then ranks all sessions in A in non-decreasing order of $g(a)k(a)$. After that, it transmits sessions one by one and assigns remaining bandwidth to other sessions orderly. When the algorithm terminates, T_{wait} is found. This algorithm invokes algorithm 1 and algorithm 2. Algorithm 1 initializes the bandwidth of all links. It assigns bandwidth of sessions in Q to the corresponding links. Algorithm 2 computes the bandwidth of one session, and adds the bandwidth of this session to all links on its path.

Algorithm 1 initialize

1: **for all** e such that $e\in E$ **do**
2: $\quad f(e,t)\leftarrow 0$
3: **end for**
4: **for all** a such that $a\in Q$ and $e\in a$ **do**
5: $\quad f(e,t)\leftarrow f(e,t)+f_{min}(a)$
6: **end for**

Algorithm 2 compute(a_i)

$f(a_i,t) \leftarrow \infty$
for all e such that $e \in a_i$ **do**
  $f(e,t) \leftarrow f(e,t) + x$
end for
for all e such that $e \in a_i$ **do**
  $\sum_{e' \in E^C(e)} \frac{f(e',t)}{C(e')} \leq 1$
  solve the above inequality and get a maximal x
  $f(a_i,t) \leftarrow \min\{f(a_i,t), x\}$
end for
for all e such that $e \in a_i$ **do**
  $f(e,t) \leftarrow f(e,t) + f(a_i,t)$
end for
return $f(a_i,t)$

Algorithm 3 Assign bandwidth

$T_{wait} \leftarrow 0$
$t \leftarrow 0$
for all a such that $a \in A$ **do**
  initialize
  $g(a) \leftarrow \frac{1}{\text{compute(a)}}$
end for
rank all sessions in A in non-decreasing order of $k(a)g(a)$, number them as $1, 2, \cdots, N_A$.
for $i = 1$ to N_A **do**
  initialize
  $f(a_i,t) \leftarrow$ compute(a_i)
  $T \leftarrow \frac{k(a_i)}{f(a_i,t)}$
  for $j = i+1$ to N_A **do**
    $f(a_j,t) \leftarrow$ compute(a_j)
    $k(a_j) \leftarrow k(a_j) - f(a_j,t) \cdot T$;
  end for
  $t \leftarrow t + T$
  $T_{wait} \leftarrow T_{wait} + t$
  rerank a_{i+1} to a_{N_A} in non-decreasing order of $k(a)g(a)$(number them from $i+1$ after sorted).
end for
$T_{wait} = \frac{T_{wait}}{N_A}$

V. Performance Evaluation

In order to evaluate the performance of our heuristic assignment algorithm, we conduct simulations. In our simulation experiments, 50 nodes are deployed in an area of $2500m * 2500m$. $D_T = 250m$ and $D_I = 550m$. The capacity of links are randomly distributed from $70 \sim 770$ units per second. There are 10 sessions in Q. Their minimum required flow rates range randomly from 0 to 100 units per second. The number of sessions in A ranges from 10 to 80. The amount of data to be transmitted of sessions in A ranges randomly from 0 to 100 units. We compare our heuristic assignment algorithm with the proportional fairness strategy. We refer to the ratio between waiting time of proportional assignment algorithm and that of our algorithm as improvement ratio. Improvement ratio=$\frac{T_{wait}^{proportioanl}}{T_{wait}^{our\ algorithm}}$, where $T_{wait}^{proportioanl}, T_{wait}^{our\ algorithm}$ represent waiting time of proportional assignment algorithm and that of our algorithm respectively. In order to reduce randomness, for specific N_A we compute improvement ratio twenty times and get average improvement ratio. Figure 2 presents the simulation results. Each point is the average of twenty runs based on randomly generated setting. The improvement ratio is about five. Our algorithm is very effective.

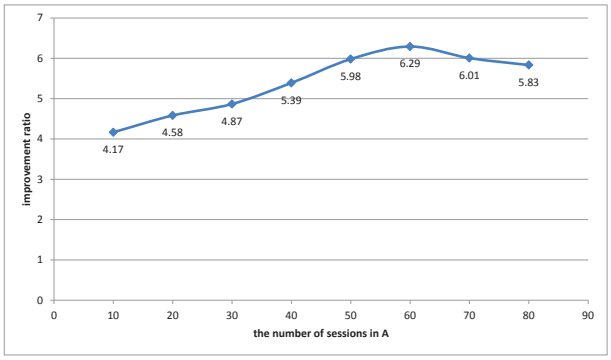

Fig. 2. $|Q| = 10$.

VI. Conclusion

In this paper, we have introduced a new strategy to assign bandwidth to improve network performance. We introduce an algorithm in small network and give a proof. We also introduce a heuristic algorithm and demonstrate its efficiency by simulation experiments. Our simulation results show that our method outperforms the existing methods.

Acknowledgment

This work is supported in part by the China National Science Fund for Distinguished Young Scholars (Grant No: 60725105), the China National Basic Research Program of China (Grant No: 2009CB320404), and the University of Hong Kong Seed Funding Programme for Basic Research (Grant No: 10400389).

References

[1] X. Su, S. Chan, and J. Manton, "Bandwidth allocation in wireless ad hoc networks: Challenges and prospects," *IEEE Commun. Mag.*, vol. 48(1), 2010.
[2] B. Li, "End-to-end fair bandwidth allocation in multihop wireless ad hoc networks," in *Proc. IEEE ICDCS*, 2005, pp. 471–480.
[3] X. Su and S. Chan, "Max-min fair rate allocation in multi-hop wireless ad hoc networks," in *Proc. MASS*, 2006, pp. 513–516.
[4] D. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 1992, ch. 6.
[5] J. L. Boudec, "Rate adaptation. congestion control and fairness: A tutorial," 2005. [Online]. Available: http://ica1www.epfl.ch/PS_files/LEB3132.pdf
[6] S. Sengupta, S. Rayanchu, and S. Banerjee, "Network coding-aware routing in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 18(4), 2010.
[7] H. Su and X. Zhang, "Modeling throughput gain of network coding in multi-channel multi-radio wireless ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 27(5), 2009.
[8] Y. T. Hou, S. S. Panwar, and H. H.-Y. Tzeng, "On generalized max-min rate allocation and distributed convergence algorithm for packet networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15(5), 2004.