Message Passing Delay in Network Congestion Management

Hengky Susanto and ByungGuk Kim

Department of Computer Science
University of Massachusetts at Lowell
{hsusanto, kim}@cs.uml.edu

Abstract — Network Utility Maximization (NUM) framework has been extensively studied. Generally, existing solutions for NUM require message exchange between network and users to regulate the flow of network traffic, and information is frequently assumed to be available instantaneously and the traffic flow adjustment is accomplished immediately. However, realistically, there is delay in message exchange because time is required for the messages to reach the designated destinations and for the traffic flow adjustment in network to take effect. Consequently, without proper synchronization, transmission rate and network pricing may oscillate, resulting in the algorithm’s failure to converge, even when there is a solution that converges to an optimal solution. Here, we propose a synchronization methodology to prevent the algorithm from oscillating.

Index Terms — networks optimization, congestion control, QoS.

1. INTRODUCTION

The study of bandwidth allocation and network congestion is often formulated into the Network Utility Maximization (NUM) framework [1,2,3,4,5,12,15]. The standard approach in solving NUM problems typically involves feedback mechanism in which the network adjusts price to control traffic and users adapt transmission rates according to the price change [3,4,5]. These existing literatures on NUM frequently assume that information on price update is available instantaneously to users and network traffic flow is adjusted immediately. In reality however, there is time delay to deliver price updates from the network to users. Additionally, network also requires some time to observe for traffic load adjustment to take effect. Thus, the time interval between price updates must be synchronized; otherwise the algorithm does not converge and will oscillate. This paper firstly addresses and investigates the asynchronous condition in message passing that leads to oscillation. Following this, we propose a synchronization protocol to prevent oscillation.

Much of the existing studies on feedback scheme have been devoted to estimate or determine the appropriate time interval for price updates. In [9], the authors observe that 3 microseconds interval time between updates is sufficient for their feedback based congestion control algorithm. However, a constant time interval may be ineffective when the message relay time is longer than 3 microseconds, or inefficient when the delay is significantly less than 3 microseconds. Alternatively, authors of [10,11] propose a pricing feedback scheme that requires users or network to sample the network state and its round trip time (RTT) where the broadcast update interval is determined according to the information observed from collected samples. Notably, the sampling is conducted during non-congested periods and therefore the proposed RTT used in determining the update interval is not suitable for a congested network because such network usually has longer RTT. Furthermore, the proposed algorithms in [9,10,11,17] assume there is sufficient time for every party in the network to exchange messages within the determined time interval. In addition, message exchange between parties is also assumed to be properly synchronized. Likewise, authors of [8,12] incorporates notification delay into their NUM model with an assumption that message passing is synchronized. On the other hand, Explicit Congestion Notification (ECN), an extension of IP/TCP, provides end-to-end network congestion notification mechanism by marking packets, enabling sources to lower their transmission rate [6,7]. However, unlike solutions proposed to solve NUM, synchronization in ECN is between flows or sources, not between network and sources. In summary, all of the above proposals do not consider synchronization.

In this paper, we investigate the impact of asynchronous and delay in message exchange, specifically oscillation. Then, we propose a mathematical model for the required maximum delay to deliver price notifications and a message exchange protocol that resolves oscillation. We begin our proposal with problem formulation in section II. Following this, we present our major contributions: the mathematical model for required maximum delay and a synchronization protocol. The simulation results are presented and discussed in section IV, followed by concluding remarks.

II. SYSTEM SETUP

Consider a network with a set of links $L$, and a set of link capacities $C$ over the links. Given a utility function $U_s(x_s)$ of user $s$ with the allocated bandwidth of $x_s$, the NUM formulation becomes

$$\text{maximize } \sum_{s \in S} U_s(x_s) \quad (P)$$

$$\text{s.t. } Ax \leq C$$

$$\text{over } x \geq \bar{x}$$

One of the existing solutions for NUM is the Contract-Net protocol. The Contract-Net protocol is a two-level negotiation protocol that runs between a buyer and a seller. The buyer proposes a contract that includes a price and a deadline, and the seller accepts or rejects the contract. The buyer then repeats the process until a contract is accepted. This protocol is known to be inefficient when the delay is long.
where $S$ and $A$ denote sets of users and routing paths, respectively, and $\bar{0}$ is a vector of zeros. A route $r$ consists of a series of links $l$ such that $A_{rl} = 1$ if $l \in r$ and $A_{rl} = 0$, otherwise. The user utility function is defined as follows.

$$U_s(x_s) = \frac{1}{1 + e^{-x_s}} \tag{1}$$

The NUM formulation is solved by the Lagrangian method. Typically, a dual problem to the primal problem of (P) is constructed as follows.

$$L(x, \lambda) = \sum_{s \in S} U_s(x_s) - \lambda^T (C - Ax),$$

$$\quad = \sum_{s \in S} U_s(x_s) - \sum_{s \in S} \lambda_s x_s + \sum_{l \in L} \lambda_l c_l,$$

where the Lagrangian multipliers $\lambda_s$ are interpreted as the link costs and

$$\lambda_s = \sum_{l \in r_s} \lambda_l.$$

The dual problem of (P) is then defined as

$$\min D(\lambda) \tag{D}$$

subject to $\lambda \geq \bar{0},$

where the dual function

$$D(\lambda) = \max_{0 \leq x \leq x_{max}} L(x, \lambda).$$

The transmission rate $x_s(\lambda_s)$ of user $s$ at link cost $\lambda_s$ can be computed in a distributed manner by

$$x_s(\lambda_s) = \arg \max_{0 \leq x \leq x_{max}} (U_s(x_s)), \tag{2}$$

A subgradient projection method is used in [3], where the network on each link $l$ updates $\lambda_s$ on that link, resulting in an iterative solution given by

$$\lambda^{(t+1)} = [\lambda^{(t)} - \sigma^t (C - Ax(\lambda^{(t)}))]^+, \tag{3}$$

where $x(\lambda^{(t)})$ is the solution of (3) and $C - Ax(\lambda^{(t)})$ is a subgradient of $D(\lambda)$ at link price $\lambda = \lambda^{(t)}$ and $x(\lambda^{(t)})$ denotes the rate allocation at $\lambda^{(t)}$, for $\lambda^{(t)} \geq \lambda_{\min}$, where $\lambda_{\min}$ can be interpreted as the network’s operation cost [13]. Also, $x(\lambda^{(t)})$ denotes the rate allocation at $\lambda^{(t)}$ and $\sigma^t$ denotes the step size to control the tradeoff between a convergence guarantee and the convergence speed, such that

$$\sigma_t \to 0, \text{ as } t \to \infty \text{ and } \sum_{t=1}^{\infty} \sigma_t = \infty.$$ 

III. MESSAGE PASSING

A. Price and Bandwidth Allocation Oscillation

In this section, we demonstrate how asynchronous communication results in oscillation in bandwidth allocation and network price. Our simulation illustrates this phenomenon by employing a network of four nodes with three links shared by three users (or flows), as depicted in figure 1. Every link in this network has a capacity limit of 10.

![Fig. 1. Parking Lot Configuration.](image)

<table>
<thead>
<tr>
<th>$x^\text{min}$</th>
<th>User 0</th>
<th>User 1</th>
<th>User 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow 0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Flow 1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Flow 2</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

| Distance from D | 1 | 2 | 3 |

Table 1. Simulation setups.

In this scenario, the three users have identical setup, as described in table 1. Variable $x^\text{min}$ in table 1 denotes user minimum bandwidth requirement. Each user is associated with time delay, i.e. the time required to deliver price notification packet to reach its destination and for user to adjust their transmission rate. The delay also includes propagation and queue delays. The simulation is implemented in the following steps. Network congestion occurs at link CD and router D broadcasts the price notification to user 0, 1, and 2. In this simulation, users use the utility function described in (1) to decide their transmission rate. The network sends price update every 1 unit time by solving eq. (3) while each user responds to the price update by adjusting the transmission rate by solving (2).

![Fig. 2.a (Network price) and 2.b (Bandwidth allocation).](image)

Figure 2.a describes two outcomes from our simulation: first, the figure demonstrates that the price oscillates. Secondly, the price fluctuation becomes more severe over time, as illustrated by the progressively enlarging waves. The price oscillation causes bandwidth allocation of every user in the network to also oscillate, as shown in figure 2.b. This should not be the outcome because with synchronized message exchange all three users with identical setup should receive the same bandwidth allocation. Further, figure 2.b
illustrates that the discrepancies between the three rate allocations continue to grow. The reason is network updates its price too early and that there is insufficient time for price notification to reach user 1 and 2 before network sends another price update. This also causes the network to make false observation because there has not been sufficient time for the adjustment in transmission rate to take effect. In other words, the network updates the price based on the traffic condition that is observed every one unit time interval while the accurate assessment of the traffic is after a lapse of 3 units time or longer (when after the entire users have updated their transmission rate). This simulation confirms that lack of synchronization between network and users leads to oscillation in price and bandwidth allocation, resulting in failure to converge. We address the price oscillation problem in the following section. We estimate the appropriate delay required to delivering price notification by modeling traffic delay from the origin of price notification to intended users.

B. Notification Delay

In order to minimize the delay of delivering price notifications, our model treats every price notification packet as a high priority packet using non-preemptive priority queue scheme. In doing this, we place price notifications in the front of the router processing queue and is served as soon as the resource becomes available. However, in our model, the undergoing service packet is allowed to complete service without interruption even if packets of higher priority arrive in the meantime. Using little theorem [14], the delay price notification \( d_{s(l)}^{\text{info}} \) of user \( s \) at each link \( l \) can be estimated as follows.

\[
d_{s(l)}^{\text{info}} = \frac{T_{s'(l)}}{1 - \rho_{s}^{\text{info}}}, \tag{4}
\]

where \( T_{s'(l)} \) denotes the average remaining service time of user \( s' \) being served in link \( l \) when \( s \) arrives at \( l \), and the processing ratio

\[
\rho_{s}^{\text{info}} = \frac{x_{s}^{\text{info}}}{x_{s}}.
\]

Here, \( x_{s}^{\text{info}} \) is the time required to process price notification packet and \( x_{s} \) is amount of bandwidth allocated to user \( s \). So, the time required \( d_{s}^{\text{info}} \) for the price notification packet to reach user \( s \) can be estimated by

\[
d_{s}^{\text{info}} = \sum_{l \in r(s)} \left( d_{s(l)}^{\text{info}} + d_{s(l)}^{\text{propa}} \right),
\]

where \( d_{s(l)}^{\text{propa}} \) denotes the negligible propagation delay at link \( l \) and \( r(s) \) is path \( r \) associated with user \( s \), for \( l \in r(s) \). Thus, with this estimation, the longest delay \( d_{l}^{\text{max}} \) from \( l \) to every user that uses link \( l \) is

\[
d_{l}^{\text{max}} = \max \left( d_{s}^{\text{info}} | s \in S(l) \right), \tag{5}
\]

where \( S(l) \) is a set of users that uses link \( l \).

With eq. (5), network therefore can estimate how much bandwidth should be allocated to \( s \) to achieve the desired price notification delivery speed. Assume that network decides the desired speed \( d_{l}^{\text{max}} \), the average delay \( d_{l(z)}^{\text{avg}} \) at each \( l \) along the path of \( r_{z} \), where \( z \) is the user with the longest delay, can be estimated as follows,

\[
d_{l(z)}^{\text{avg}} = \frac{d_{l}^{\text{max}}}{\| r_{z} \|}, \quad \text{for } l(z) \in r_{z},
\]

since network flow is naturally determined according to the slowest speed processing rate in its path, the processing rate along the path \( r_{z} \) is homogenous once bandwidth is allocated to user \( z \).

Since the size of price notification packet is known to network, \( x_{s}^{\text{info}} \) can be measured as

\[
x_{s}^{\text{info}} = \frac{\text{price notification packet size}}{x_{s}}.
\]

Thus, by substituting \( d_{l(z)}^{\text{info}} \) and manipulating eq. (4), the allocated bandwidth \( x_{s} \) can be estimated as follows.

\[
x_{s} \geq \sqrt{d_{l(z)}^{\text{max}} \frac{\text{price notification packet size}}{T_{s'(l)} - T_{s(z)} | r_{z} |}}.
\]

Conceding that user has minimum demand \( x_{s}^{\text{min}} \), problem (3) therefore can be reformulated in the following manner.

\[
\max \sum_{s \in S} U_{s}(x_{s}') \tag{6.a}
\]

\[
\sum_{s \in S} x_{s}' \leq C_{l}, \quad \forall l \in L, \tag{6.b}
\]

\[
\text{over } x_{s}' \geq x_{s}^{\text{min}} + x_{s} \tag{6.c}
\]

The advantage of problem (6) is that the additional traffic caused by price notification packets can be absorbed into the bandwidth allocation of respective users, as described in (6.c). This avoids the problem of overhead cost from notification messages.

![Fig. 3. Price updates in every RTT.](image-url)
C. Message Synchronization

In this section, we propose a synchronized message exchange scheme between users and network. When congestion is detected at link $l$, network (router at link $l$) broadcasts the price notification to users whose flow traverse through it. Then, users respond with an ACK message upon receiving the price notification. The network observes whether traffic condition improves after ACK messages from the entire users on $l$ are received. The synchronization relationship between users and the network is depicted in figure 4.

![Fig. 4. Message Passing Protocol.](image)

In practice, ICMP packet can be adopted for price notification packet scheme by taking advantage of the unused reserved type in ICMP header, shown in figure 5.a. Similarly, this concept can be considered for ACK message. However, there is high overhead cost with the addition of ACK messages in the network because users may have to send messages to every link in which their flow traverses. To reduce overhead, ACK message can be realized through RTT by utilizing information from eq. (5). This allows user to only update their transmission rate upon receiving the price notification, and network next updates the price when RTT expires. In order to compute RTT, router also needs to know other routers’ processing power that lay on the path to destination. This can be accomplished by using ICMP’s traceroute function [16] and take advantage of the unused reserved type in ICMP header to collect the necessary information from each router on the path $r(s)$ and modified ICMP header, as depicted in figure 5.b.

![Fig. 5.a. ICMP header for price notification.](image)

![Fig. 5.b. ICMP header of the processing rate messages.](image)

The interval length between price update can be bounded with $RTT_1$ by utilizing (4) and (5), such that

$$RTT_1 \geq 2d_l^{\text{max}} + d_l^{\text{serv}},$$

where $d_l^{\text{serv}}$ is the slowest processing rate at the destination server in $l$ to process the price notification and to adjust the transmission rate in $l$. So the maximum interval length $RTT_{\text{max}}$ can be estimated as follows.

$$RTT_{\text{max}} = \max\{RTT_1^s\} + \varepsilon, \quad \forall s \in l, \quad (7)$$

where $\varepsilon$ is a positive constant error term and $RTT_1^s$ denotes $RTT_1$ over a period time of $t \to \infty$.

Proposition 1: There exists a solution for dual problem $(D)$ that converges to an optimal solution, then the solution with $RTT_{\text{max}}$ also converges to an optimal solution.

Proof. Assume there exists a solution for dual problem $(D)$ that converges to an optimal solution under the condition that both the price notification and user’s transmission rate update is available instantaneously. By condition (7), the time required for network to observe the transmission update made by user is bounded by

$$RTT_s \leq RTT_{\text{max}}, \quad \forall s \in l,$$

where $RTT_s$ denotes the RTT of each $s$ observed by network. Then all information becomes available at every $RTT_{\text{max}}$ interval which is akin to network processing instantaneous information needed to compute a new price. Thus, if the solution for problem $(D)$ converges, then the solution also converges under iteration with $RTT_{\text{max}}$ interval.

Proposition 1 implies that oscillation can be prevented if the pricing update interval is performed in every $RTT_{\text{max}}$. This provides sufficient time for the price notification to reach the intended users and for the network to observe the changes in traffic load. Moreover, $RTT_{\text{max}}$ may also be used to estimate the duration required for the algorithm to converge. That is $t^* RTT_{\text{max}}$, where $t^*$ is the number of iterations required until the algorithm converges.

IV. SIMULATION

To illustrate the impact of the proposed protocol for message exchange synchronization, we conduct a simulation with a setup identical with the simulation in the earlier section: network with four nodes shared by three users with different distance from where the price notification originates, as depicted in figure 1 and table 1. In this section, we demonstrate that the new proposed protocol for message passing resolves the oscillation problem presented in the previous section.

![Fig. 6.a and 6.b. The network pricing and bandwidth allocation.](image)
In this simulation, the interval length between two iteration is $RTT_{\text{max}} = 7$ from solving eq. (7); this is the RTT between the furthest user from the link that broadcast the price notification, that is link “D” in figure 1. Additionally, users use the utility function described in (1). As depicted in figure 6a, the algorithm converges as the rate allocation of every user converges. In comparison to figure 2b, where each user receives different bandwidth allocation even though the setup of each user is identical, figure 6a demonstrates that the synchronization protocol allows each user to receive identical bandwidth allocation as expected. Figure 7 displays the first 50 iteration of figure 6a to describe that even though the initial adjustment occurs at different time but they eventually converge to a solution. Similarly, as shown in figure 6b, network price also achieves convergence. This simulation demonstrates that price and bandwidth allocation oscillation can be resolved with synchronization through understanding of the delay required to deliver the price notification and the waiting time needed to observe the changes in traffic load caused by users adjusting their transmission rate.

V. CONCLUSION

In this paper, we discuss the practical challenges and implications in implementing solutions that require message exchange, such as the asynchronous condition that leads to oscillation and failure to converge. Such challenges that result from implementing the solution for NUM problems must also be considered in NUM frameworks. To solve this problem, we first model the message passing delay to determine the appropriate interval for price updates. Then, we propose a synchronization protocol that avoids oscillation and leads to algorithm convergences. However, reliability and the impact of $d_{i}^{\text{max}}$ to network performance have not been discussed in this literature. Additionally, the price notification packet may be lost due to network congestion, which result in users not receiving the price updates. Thus, these issues must be addressed in our future work.

Aside from this, this methodology has the additional benefit of being applicable to pricing models in cloud computing, where the cloud charges different prices according to the resources availability and the processing load. With this approach, user has better assessment of the processing cost. This approach also provides the foundation for cloud network to determine the appropriate interval to update its price, such that oscillation in pricing and processing allocation can be avoided.

REFERENCE


