ICIC Express Letters
Volume **10**, Number **10**, October **2016**

ICIC International ©2016 ISSN 1881-803X
pp. **2429–2434**

DELAY-BASED BACKPRESSURE ALGORITHM FOR IMPROVING CONTINUITY OF VIDEO STREAMING IN MULTI-HOP WIRELESS NETWORKS

Shu Fan and Honglin Zhao

Communication Research Center
Harbin Institute of Technology
No. 92, West Dazhi Street, Nangang Dist., Harbin 150080, P. R. China
fanshufs@sina.com; hlzhao@hit.edu.cn

Received March 2016; accepted June 2016

Abstract. *Video continuity is a significant performance metric influencing the viewer-perceived video quality of on-demand video streaming that increases dramatically in wireless networks. One of the main causes of video discontinuity is due to buffer underflow caused by network delay. Therefore, it is meaningful to reduce per-flow end-to-end delay and provide delay guarantees in video streaming transmission. In this study, by applying the Lyapunov optimization framework, a backpressure-style cross-layer algorithm that uses explicit delay information from the head-of-line packet of each queue is proposed. The policy ensures deterministic worst case delay guarantees and is designed for multi-hop models. A novel virtual queue that can be considered as a discontinuity penalty is also designed to reduce the buffer underflow. The simulation results evaluated with Matlab show that compared with the existing works, the algorithm presented in this paper obtains better video continuity and lower end-to-end delay.*

Keywords: Lyapunov optimization, Wireless networks, Backpressure, Video continuity, End-to-end delay

1. **Introduction.** With a significant increase in the use of smart multimedia devices equipped in wireless multi-hop networks, on-demand video streaming has become a major component of the dramatically increasing data traffic [1]. To keep up with the increasing demand of bandwidth, a backpressure-style policy [2] becomes a promising scheme among various different existing policies because of its throughput-optimal characteristic. By applying the Lyapunov optimization technique, we can design cross-layer algorithms providing throughput utility optimal operation guarantees [3,4]. In [5], a typical backpressure-style joint routing and scheduling algorithm for data transmission in multi-hop wireless networks is proposed. Recently, cross-layer schemes based on the Lyapunov optimization framework have been combined with multimedia services [6].

The viewer-perceived video quality is distinctively based on video continuity [6] which is the length of time that a video is played without interruptions. Video discontinuity is caused by buffer underflow that is usually due to low network throughput and high end-to-end delay. The traditional way of reducing the buffer underflow is to pre-buffer a sufficiently large video at the decoder. However, high and unbounded end-to-end delay of video packets will also result in an unwanted perceived initial buffering delay to viewers. Motivated by these considerations, in this paper, we focus on reducing end-to-end delay of video packets and providing worst case delay guarantees to limit video discontinuity.

Related works on delay reduction can be found in [7-11]. In [7], the delay bounds in wireless ad hoc networks are studied using backpressure scheduling with either one-hop or multihop traffic flows. In [8], the authors propose a cross-layer algorithm providing average end-to-end delay guarantees. These prior works can only provide bounds on the overall average delay via Little's theorem, except for individual sessions. There are several works aiming to reduce end-to-end delay for individual sessions. In [9], the authors develop a delay-aware cross-layer algorithm using a novel link-rate allocation strategy and a regulated scheduling policy. A hop-count based queuing structure is used in [10] to provide a worst case hop count to the destination. However, these works fail to provide explicit end-to-end delay guarantees. Deterministic worst case delay guarantees are derived from the algorithm in [11] that uses explicit delay information from the head-of-line packet at each queue in one-hop networks. However, the algorithm in [11] is only for one-hop network model.

To our best knowledge, our algorithm DBAIC (Delay-based Backpressure Algorithm for Improving Continuity) is the first work to increase continuity of on-demand video streaming using backpressure routing/scheduling policies in multi-hop wireless networks. The key contributions of this paper can be summarized as follows.

- By applying the Lyapunov optimization framework [5], we have developed a joint routing and scheduling algorithm which provides deterministic worst case delay guarantees for multi-hop wireless networks.
- A novel virtual queue at the decoder node is designed and introduced into the Lyapunov optimization framework to reduce the buffer underflow and increase video continuity in multi-hop wireless networks
- The simulation results show that compared with existing works, the proposed algorithm obtains better video continuity with a lower end-to-end delay.

The structure of the rest of the paper is as follows. Section 2 introduces the system model and design of queues. In Section 3, the algorithm is designed using Lyapunov optimization. Simulation results are presented in Section 4. Conclusions are provided in Section 5.

2. Network Model and Design of Queues.

Let the network be modeled by a directed connectivity graph $G(N, L)$, where N denotes the set of nodes and $(i, j) \in L$ represents a unidirectional wireless link between node i and node j. Define M as the set of unicast video sessions m between source-destination pairs in the network. N_s denotes the set of source nodes s_m, and N_d represents the set of destination nodes d_m of session m. Packets from the source node traverse multiple wireless hops before arriving at the destination node in the networks. FIFO (First-In-First-Out) is applied as the service discipline. Video packets arriving after the playout deadline are assumed to be lost. We also assume that video reordering has been done before the packets enter the decoder buffer. Pre-buffering and re-buffering policies are applied at the decoders. The system is assumed to run in a time-slotted manner. $\alpha_{nj}(t) \in \{0, 1\}$ is used for indicating whether link (n, j) is used for transmitting packets in time slot t. $\alpha_{nj}(t) = 1$ implies that the link (n, j) is scheduled. In this model, scheduling is subject to the following constraints:

$$\sum\nolimits_{i\in O(n)} \alpha_{ni}(t) + \sum\nolimits_{j\in I(n)} \alpha_{jn}(t) \le 1 \tag{1}$$

$$\alpha_{ni}(t) + \sum\nolimits_{k\in N} \sum\nolimits_{l} \alpha_{kl}(t) \le 1 \tag{2}$$

where node l is in the transmission range of n. $O(n)$ denotes the set of nodes i with $(n, i) \in L$. $I(n)$ denotes the set of nodes j with $(j, n) \in L$.

There is a queue for each session at each node, and the queue backlog for session m at the network layer of node n in slot t is denoted by $Q_n^{(m)}(t)$. In each slot t, the queue backlog is updated as follows:

$$\begin{aligned} Q_n^{(m)}(t+1) = \max &\left[Q_n^{(m)}(t) - \sum\nolimits_{i\in O(n)} \mu_{ni}^{(m)}(t) - D_n^{(m)}(t), 0 \right] \\ &+ \sum\nolimits_{j\in I(n)} \mu_{jn}^{(m)}(t) + 1_{\{n=s_m\}}.r_m(t) \end{aligned} \tag{3}$$

where $r_m(t) = 1$ represents the number of packets generated in session m and injected into the network layer in slot t. $\mu_{ni}^{(m)}(t)$ represents the number of packets of video session m to be forwarded from node n to node i in time slot t. $D_n^{(m)}(t)$ represents the number of packets of session m dropped at node n in slot t and $D_n^{(m)}(t) \in \{0, 1\}$. $1_{\{n=s_m\}}$ denotes an indicator function that is equal to 1 if $n = s_m$ and to 0 otherwise. $Q_n^{(m)}(t) = 0$ if $n = d_m$.

The virtual queue X_m that is at d_m of session m is updated in each time slot as follows:

$$X_m(t+1) = \max\left[X_m(t) - \sum\nolimits_{i\in I(d_m)} \mu_{id_m}^{(m)}(t), 0\right] + p_m(t) \tag{4}$$

where $p_m(t) \in \{0, 1\}$ denotes the number of video playout packets removed from the buffer in each slot. If each virtual queue X_m is guaranteed to be stable, according to the necessity and sufficiency for queue stability [5], the average rate of video packets received by the decoder is higher than the average rate of video packets being played out, and the buffer underflow can be reduced to maintain video continuity.

$H_n^{(m)}(t)$ denotes the waiting time of the head-of-line packet in the queue of session m at node n in slot t, and define $H_n^{(m)}(t) = 0$ if there is no packet in the queue in slot t. $H_n^{(m)}$ satisfies the following update rule in each slot:

$$\begin{aligned} H_n^{(m)}(t+1) = \max&\left[H_n^{(m)}(t) - \left(\sum\nolimits_{i\in O(n)} \mu_{ni}^{(m)}(t) + D_n^{(m)}(t)\right).T_n^{(m)}(t), 0\right] \\ &+ 1 - 1_{\left\{Q_n^{(m)}(t+1)=0\right\}} \end{aligned} \tag{5}$$

where $T_n^{(m)}(t)$ denotes the inter-arrival time between the head-of-line packet and the subsequent packet in the queue of session m at node n in slot t. $1_{\left\{Q_n^{(m)}(t+1)=0\right\}}$ denotes an indicator function that is equal to 1 if $Q_n^{(m)}(t+1) = 0$ and equal to 0 otherwise.

3. **Dynamic Algorithm via Lyapunov Optimization.**

The Lyapunov optimization technique is applied. Let $\Theta(t) = [Q(t), X(t), H(t)]$ be the network state vector in time slot t. Define the Lyapunov function as follows:

$$L(\Theta(t)) = \frac{1}{2}\left\{\sum\nolimits_{m\in M}\sum\nolimits_{n\neq d_m}\left[\left(Q_n^{(m)}(t)\right)^2 + \left(H_n^{(m)}(t)\right)^2\right] + \sum\nolimits_{m\in M}(X_m(t))^2\right\} \tag{6}$$

Thus, the following inequality can be derived:

$$\begin{aligned} &E\{\Delta(\Theta(t))\} \\ &\leq B + \sum\nolimits_{m\in M} X_m(t)\left[p_m(t) - \sum\nolimits_{i\in I(d_m)} \mu_{id_m}^{(m)}(t)\right] \\ &\quad + \sum\nolimits_{m\in M}\sum\nolimits_{n\neq d_m} H_n^{(m)}(t)\left[1 - 1_{\left\{Q_n^{(m)}(t+1)=0\right\}}\right. \\ &\quad \left. - \left(\sum\nolimits_{i\in O(n)} \mu_{ni}^{(m)}(t) + D_n^{(m)}(t)\right).T_n^{(m)}(t)\right] \\ &\quad + \sum\nolimits_{m\in M}\sum\nolimits_{n\neq d_m} Q_n^{(m)}(t)\left[\sum\nolimits_{j\in I(n)} \mu_{jn}^{(m)}(t) + 1_{\{n=s_m\}}.r_m(t)\right. \\ &\quad \left. - \sum\nolimits_{i\in O(n)} \mu_{ni}^{(m)}(t) - D_n^{(m)}(t)\right] \\ &= B + \Psi_1(t) - \Psi_2(t) - \Psi_3(t) - \Psi_4(t) \end{aligned}$$

where $\Psi_1(t)$, $\Psi_2(t)$, $\Psi_3(t)$ and $\Psi_4(t)$ can be evaluated as follows:

$$\begin{aligned}
\Psi_1(t) &= \sum\nolimits_{m\in M} Q_n^{(m)}(t).r_m(t).1_{\{n=s_m\}} + \sum\nolimits_{m\in M}\sum\nolimits_{n\neq d_m} H_n^{(m)}(t)\left(1-1_{\left\{Q_n^{(m)}(t+1)=0\right\}}\right) \\
&\quad + \sum\nolimits_{m\in M} X_m(t).p_m(t) \\
\Psi_2(t) &= \sum\nolimits_{m\in M}\sum\nolimits_{n\neq d_m}\sum\nolimits_{i\in O(n), i\neq d_m} \mu_{ni}^{(m)}(t)\left[Q_n^{(m)}(t) - Q_i^{(m)}(t) + H_n^{(m)}(t)T_n^{(m)}(t)\right] \\
\Psi_3(t) &= \sum\nolimits_{m\in M}\sum\nolimits_{l\in I(d_m)} \mu_{ld_m}^{(m)}(t)\left[X_m(t) + Q_l^{(m)}(t) + H_l^{(m)}(t)T_l^{(m)}(t)\right] \\
\Psi_4(t) &= \sum\nolimits_{m\in M}\sum\nolimits_{n\neq d_m} D_n^{(m)}(t)\left[Q_n^{(m)}(t) + H_n^{(m)}(t)T_n^{(m)}(t)\right]
\end{aligned}$$

B is a constant and satisfies the following:

$$\begin{aligned}
B \geq{}& \frac{1}{2}\sum\nolimits_{m\in M}\left[(p_m(t))^2 + \left(\sum\nolimits_{i\in I(d_m)} \mu_{id_m}^{(m)}(t)\right)^2\right] \\
&+ \frac{1}{2}\sum\nolimits_{m\in M}\sum\nolimits_{n\neq d_m}\left[\left(\sum\nolimits_{j\in I(n)} \mu_{jn}^{(m)}(t) + 1_{\{n=s_m\}}.r_m(t)\right)^2\right. \\
&\left. + \left(\sum\nolimits_{i\in O(n)} \mu_{ni}^{(m)}(t) + D_n^{(m)}(t)\right)^2\right] \\
&+ \frac{1}{2}\sum\nolimits_{m\in M}\sum\nolimits_{n\neq d_m}\left\{\left(1-1_{\left\{Q_n^{(m)}(t+1)=0\right\}}\right)^2\right. \\
&\left. + \left[T_n^{(m)}(t).\left(\sum\nolimits_{i\in O(n)} \mu_{ni}^{(m)}(t) + D_n^{(m)}(t)\right)\right]^2\right\}
\end{aligned}$$

Considering $p_m(t) \in \{0,1\}$, $0 \leq \mu_{ij}^{(m)}(t) \leq C_{\max}$, $r_m(t) = 1$, $T_n^{(m)}(t) \leq H_n^{(m)}(t) \leq$ *Delaybound* (shown in the packet dropping policy) and $D_m(t) \in \{0,1\}$, we can conclude that the constant B must exist.

The algorithm DBAIC includes the following four components.

- **Joint routing and scheduling at relay nodes:** At node $n \neq d_m$, routing and scheduling decisions $\mu_{ni}^{(m)}(t)$ for each session $m \in M$ can be made by solving the following:

$$\begin{aligned}
&\max \Psi_2(t) \\
&\text{s.t. } 0 \leq \mu_{ni}^{(m)}(t) \leq C_{\max}, \quad (1),\ (2)
\end{aligned}$$

For each link (n,i), the session m^* for link (n,i) can be chosen as follows: $m^* = \arg\max_{m\in M}\left\{Q_n^{(m)}(t) - Q_i^{(m)}(t) + H_n^{(m)}(t)T_n^{(m)}(t)\right\}$. The weight of link (n,i) is defined as follows: $w_{ni} = Q_n^{(m^*)}(t) - Q_i^{(m^*)}(t) + H_n^{(m^*)}(t)T_n^{(m^*)}(t)$. Therefore, the joint routing and scheduling problem can be reduced to the following:

$$\begin{aligned}
&\max \sum\nolimits_{n\neq d_m}\sum\nolimits_{i\in O(n), i\neq d_m} \mu_{ni}^{(m^*)}(t).w_{ni} \\
&\text{s.t. } 0 \leq \mu_{ni}^{(m^*)}(t) \leq C_{\max}, \quad (1),\ (2)
\end{aligned} \tag{7}$$

Transmission rates $\mu_{ni}^{(m^*)}(t)$ are chosen on the basis of (7), which is a difficult problem to solve as it requires global knowledge and a centralized algorithm.

- **Joint routing and scheduling at destination nodes:** Choose $\mu_{ld_m}^{(m)}(t)$ to solve the following problem:

$$\begin{aligned} &\max \Psi_3(t) \\ &\text{s.t. } 0 \le \mu_{ld_m}^{(m)}(t) \le C_{\max}, \quad (1),\ (2) \end{aligned}$$

For each session m, the node l^* for session m can be chosen as follows: $l^* = \arg\max_{l \in I(d_m)} \left\{ X_m(t) + Q_l^{(m)}(t) + H_l^{(m)}(t) T_l^{(m)}(t) \right\}$. The weight of session m is defined as follows: $w_m = X_m(t) + Q_{l^*}^{(m)}(t) + H_{l^*}^{(m)}(t) T_{l^*}^{(m)}(t)$. Therefore, the joint routing and scheduling problem can be reduced to the following:

$$\begin{aligned} &\max \sum\nolimits_{m \in M} \mu_{l^* d_m}^{(m)}(t).w_m \\ &\text{s.t. } 0 \le \mu_{l^* d_m}^{(m)}(t) \le C_{\max}, \quad (1),\ (2) \end{aligned} \tag{8}$$

Problem (8) is difficult to solve as it requires global knowledge and a centralized algorithm.

- **Packet dropping:** For each queue that has the packets that were not transmitted successfully in the transmission scheduling, drop the head-of-line packet of the queue if $H_n^{(m)}(t) > \mathit{Delaybound}$, where *Delaybound* is a constant.
- **Update of queues:** $Q(t)$, $X(t)$ and $H(t)$ are updated using (3), (4) and (5), respectively, in each time slot.

The backlog of a queue is no more than the number of waiting slots of the head-of-line packet in the queue, and thus $Q_n^{(m)}(t) \le H_n^{(m)}(t)$. $Q_n^{(m)}(t)$ is bounded by *Delaybound*, for $H_n^{(m)}(t)$ is bounded by *Delaybound* according to the packet dropping policy. Therefore, network is stable.

4. **Simulation.** For the simulations, we consider a network with 20 nodes randomly distributed in a square area of 1600 m^2. The transmission or interference range of a node is 15 m. There are four unicast video sessions which randomly choose sources and destinations in the networks. Simulations are run in Matlab R2014a. The greedy maximal scheduling (GMS) method [12] is used for schedulable link set generation. The length of a video file is 1000 packets. The link transmission capacity is 3 packets per time slot. *Delaybound* is 50 slots and the playout delay deadline is 100 slots. In the simulations, playout continuity is the ratio of the length of time that a video is played to the length of total time. Since the cross-layer scheme called backpressure proposed in [5] is the typical backpressure algorithm with throughput-optimal characteristic, it can be regarded as the baseline scheme of video transmission in multi-hop wireless networks. In the simulations, the performance of DBAIC is compared with backpressure.

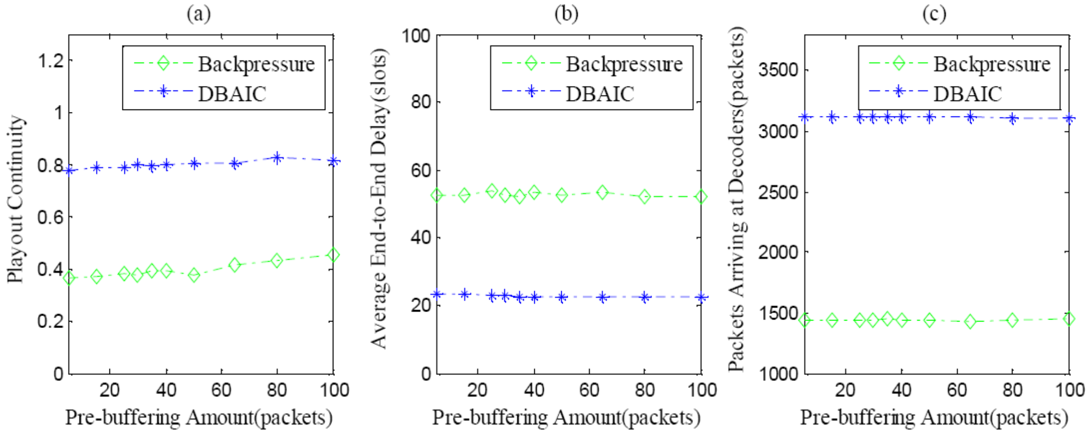

Figure 1. Performances achieved by DBAIC and backpressure

From Figure 1(a), we can see that the value of the playout continuity that DBAIC can achieve is about 0.8 and is much higher than the playout continuity achieved by backpressure. Figure 1(b) shows that the average end-to-end delay in DBAIC is approximately 20 slots and is about 30 slots lower than the average end-to-end delay in backpressure. Compared with backpressure, DBAIC increases the amount of video packets arriving at decoders by about 1500 packets, which is shown in Figure 1(c). The simulation results show that DBAIC performs much better than backpressure.

5. **Conclusions.** By using the Lyapunov optimization framework, in this study, we developed a cross-layer control algorithm, which is applicable to backpressure-based wireless networks. Multi-hop models were considered. Compared with the existing works, the algorithm achieved a better performance by using explicit delay information of the head-of-line packet in each queue in the routing and scheduling. Theoretical analyses demonstrated that the network stability could be guaranteed. For future study, we plan to combine this policy with adaptive playout rate schemes to reduce the amount of pre-buffering while maintaining video continuity.

Acknowledgment. This work was supported by the National Basic Research Program of China under Grant No. 2013CB329003.

REFERENCES

[1] S. Ehsan and B. Hamdaoui, A survey on energy-efficient routing techniques with Qos assurances for wireless multimedia sensor networks, *IEEE Communications Surveys & Tutorials*, vol.14, no.2, pp.265-278, 2012.

[2] L. Tassiulas and A. Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, *IEEE Trans. Automatic Control*, vol.37, no.12, pp.1936-1948, 1992.

[3] A. Cammarano, F. L. Presti, G. Maselli, L. Pescosolido and C. Petrioli, Throughput optimal cross-layer design for cognitive radio ad hoc networks, *IEEE Trans. Parallel and Distributed Systems*, vol.26, no.9, pp.2599-2609, 2014.

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

[5] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, Morgan & Claypool Publishers, 2010.

[6] E. Tan and C. T. Chou, A frame rate optimization framework for improving continuity in video streaming, *IEEE Trans. Multimedia*, vol.14, no.3, pp.910-922, 2012.

[7] L. B. Le, K. Jagannathan and E. Modiano, Delay analysis of maximum weight scheduling in wireless ad hoc networks, *The 43rd Annual Conference on Information Sciences and Systems*, pp.389-394, 2009.

[8] D. Y. Xue and E. Ekici, Delay-guaranteed cross-layer scheduling in multihop wireless networks, *IEEE/ACM Trans. Networking*, vol.21, no.6, pp.1696-1707, 2013.

[9] H. Z. Xiong, R. Li, A. Eryilmaz and E. Ekici, Delay-aware cross-layer design for network utility maximization in multi-hop networks, *IEEE Journal on Selected Areas in Communications*, vol.29, no.5, pp.951-959, 2011.

[10] L. Ying, S. Shakkottai, A. Reddy and S. Liu, On combining shortest-path and back-pressure routing over multihop wireless netwoks, *IEEE/ACM Trans. Networking*, vol.19, no.3, pp.841-854, 2011.

[11] M. J. Neely, Delay-based network utility maximization, *IEEE/ACM Trans. Networking*, vol.21, no.1, pp.41-54, 2013.

[12] Z. Z. Jiao, Z. Yao, B. X. Zhang and C. Li, NBP: An efficient network-coding based backpressure algorithm, *IEEE International Conference on Communications*, Budapest, Hungary, pp.1625-1629, 2013.