Inter-Burst Segregation Protocol Guaranteeing Loss-Free Packet-Switched Networks

Sachin Sharma, Didier Colle, Wouter Tavernier, Mario Pickavet, and Piet Demeester

***Abstract*— In this letter, we propose a novel protocol, inter-burst segregation protocol (IBSP), which guarantees zero packet loss in packet-switched networks. The protocol is implemented and evaluated in network simulator-3 and in the data plane development kit (DPDK) implemented by INTEL. The results confirm that a packet-switched network can achieve zero packet-loss using IBSP, although nearly all the bandwidth is consumed in the network. In addition, the jitter is significantly low and bounded using IBSP.**

***Index Terms*— TDM, packet-switching, leaky bucket.**

I. Introduction

TDM (Time-Division Multiplexing) networks achieve high Quality of Service (QoS) in terms of packet-loss, jitter, and bandwidth. This is because resources (i.e., bandwidth) are statically divided into different time slots in periodic intervals (i.e., timeframes) and each user is allocated a unique time slot (in each timeframe) to transmit data traffic. The problem is that if a user does not have traffic to transmit, its allocated time slot gets wasted. Packet-switched networks solve this problem by dynamically sharing resources among many users. However, due to accumulated congestion, these networks cannot guarantee QoS in terms of packet-loss, jitter, and bandwidth.

Many standards, such as IEEE 802.17 for optical fiber [1] and IEEE 802.1Qbb (approved in 2011) for Ethernet [2], have been proposed to decrease packet loss in packet-switched networks. In these standards, a control packet is sent back to the sender to notify about congestion and hence, to control its traffic rate (or burstiness). The issue here is that if the control packet is lost or there is a large distance between the congested and sender node, the former may suffer buffer overflow and packet-loss can happen [3]. Furthermore, Diff-Serv (differentiated services) is proposed to deliver QoS to high-priority traffic in packet-switched networks. However, Diffserv can only provide QoS when there is a small fraction of high priority traffic in the network [4]. Moreover, it does not guarantee zero packet-loss.

In this article, we propose the Inter-Burst Segregation Protocol (IBSP) to overcome above problems. The main contributions of our article are:

1) A network scenario, which illustrates a possibility of packet-loss even in extreme low load conditions in a packet-switched network. (Section II)

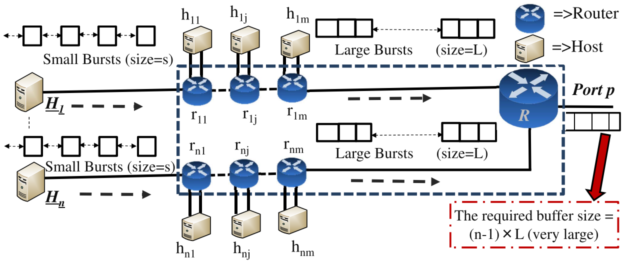

Fig. 1. Worst case network scenario of lossy packet-switched networks.

2) A novel protocol (i.e., IBSP), which implements a concept of TDM in a packet-switched network to guarantee zero packet-loss. (Section III)

3) Experimentation results (e.g., using DPDK) on the FIRE testbed [5]. (Section IV)

II. Packet-Loss in Packet-Switched Networks

In this section, we describe a worst case scenario (Fig. 1) in which, although the offered traffic load is less than (or equal to) the traffic load that can be handled by the network, congestion occurs and packet-loss happens.

For illustrating this scenario, we assume that the link capacity (in bits/s) between nodes (routers shown in Fig. 1) in the network is at least equal to the average of incoming rates of data transfers from different users (e.g, hosts in Fig. 1). In addition, the maximum rate of a data transfer (from a user) depends on the bandwidth assured in its service level agreement (SLA). In order to ensure the maximum rate, traffic shaping such as leaky bucket [6] is applied at the ingress (i.e., at all hosts in Fig. 1). Leaky bucket shapes traffic such that burstiness is bounded and on top of that it limits traffic by dropping (or marking) excess traffic beyond the SLA. In case of a well-dimensioned non-overbooked network, using leaky buckets at the ingress, the average load will not exceed the load that can be handled by the network.

In Fig. 1, there are two different hosts: H_i (where $1 \leq i \leq n$) and h_{ij} (where $1 \leq j < m$). Host H_i transmits bursts (shaped by leaky bucket), which are forwarded through a linear chain of routers (r_{ik}, where $1 \leq k \leq m$) and then forwarded through router R from port p. Instead, host h_{ij} transmits bursts (shaped by leaky bucket) to its neighboring host (i.e., h_{ij+1}) such that these create interference at r_{ij} to bursts originated from H_i, resulting in delay in transmission of bursts of H_i at r_{ij}. As bursts originated from all H_i collide at router R, we derive the required buffer size of router R at port p. Let t_H and t_h be the average time intervals at which hosts H_i and hosts h_{ij} transmit bursts respectively. For the worst case scenario, t_H is assumed to be less than t_h.

In the network scenario (Fig. 1), bursts originating from H_i are of size S (in bits) and when a burst from H_i passes through a router (i.e., r_{ij}), the burst of H_i may have to wait until bursts

Manuscript received June 28, 2016; accepted July 18, 2016. Date of publication July 27, 2016; date of current version October 7, 2016. This research was partially funded by ICON project - MECaNO (grant no. 130646). The associate editor coordinating the review of this paper and approving it for publication was M. Naeini. *(Corresponding author: Sachin Sharma.)*

S. Sharma was with Ghent University-iMinds, Gent 9052, Belgium. He is now with NEC Laboratories Europe, Heidelberg 69115, Germany (e-mail: sachin.sharma@neclab.eu).

D. Colle, W. Tavernier, M. Pickavet, and P. Demeester are with Ghent University-iMinds, Gent 9052, Belgium.

Digital Object Identifier 10.1109/LCOMM.2016.2594291

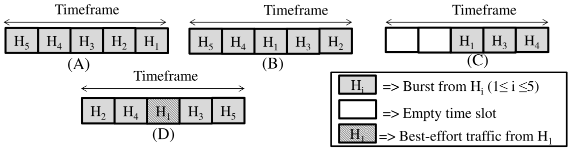

Fig. 2. Timeframes at port p of router R (see Fig. 1) in IBSP.

of another host (i.e., h_{ij}) are transmitted. In addition, as t_H is assumed to be less than t_h, it may happen that the current burst of H_i catches up with previous bursts of H_i, which may have been queued somewhere in the network, making a large burst (size = L, in bits). The value of L can be represented by Eq. 1, i.e., as a function of the size of bursts originated from H_i (i.e, S), the average number of interfering flows per hop (i.e., f), the average delay (i.e., d) occurred per interfering flow, the number of hops traveled (i.e., m), and the average time interval at which bursts are transmitted (i.e., t_H and t_h).

$$L = F(S, f, d, m, t_H, t_h) \tag{1}$$

Let T be a time interval at which a burst of size L can be transmitted through port p (i.e., $T = L/bitrate$, where $bitrate$ is the bandwidth of port p). Assume that bursts of size L from each host H_i reach at router R at the same time. Hence, the total size of incoming bursts at R at time T is given by $n \times L$. As the size of an outgoing burst at time T is given by L, the buffer size requirement of port p is given by:

$$B_p = n \times L - L = (n - 1) \times L \tag{2}$$

Eq. 2 illustrates that if L is very large (i.e., many bursts of host H_i catch up with previous bursts), router R requires a very large buffer in order to guarantee zero packet loss. The presence of large buffers may result into unnecessary latency and poor performance (bufferbloat problem [7]). However, in the absence of large buffers, packet-loss will happen.

III. Inter-Burst Segregation Protocol (IBSP)

A. IBSP Approach

The idea of IBSP is taken from TDM in which a time slot is allocated for each user (i.e., high priority user) in each timeframe and the duration of the time slot for a user depends on the bandwidth assured in its SLA. Unlike TDM, IBSP has an advantage that resources reserved for a user in a timeframe may be used by other flows (e.g., best-effort), when the user does not have packets to transmit in its allocated time slot.

By transmitting traffic (or bursts) in timeframes, IBSP does not allow the current burst of a host[1] to be chased up with its previous bursts and hence, prevents making a large burst (e.g., a burst of size L in Fig. 1).

Fig. 2 shows bursts transmitted in timeframes at port p of router R (shown in Fig. 1) in IBSP. As a timeslot is reserved for each H_i host in each timeframe at port p, we see bursts (high priority) from all H_i hosts in the timeframe in Fig. 2A and 2B. However, unlike TDM, a unique time slot is not allocated in each timeframe for each host, i.e., a time slot can be occupied by a burst (the size assured in SLA) of any host on the first come first serve basis. In other words, a burst of a host may be distributed in any place of a timeframe, but the size of the burst should not exceed the size assured in SLA. Fig. 2C shows the case when some of hosts (i.e., H_2 and H_5) do not have traffic to transmit in their time slots and therefore, no traffic is transmitted in the corresponding time slots at that timeframe. Fig. 2D shows the case when host H_1 does not have high-priority traffic to transmit in its timeslot and it transmits best-effort traffic in that time slot.

In IBSP, if a timeslot of a timeframe is empty (i.e., neither high priority nor best-effort traffic is transmitted), each node (i.e., router) along the path does not transmit any other traffic (i.e., traffic from other users) in the corresponding time slot (shown in Fig. 2C). Therefore, bursts of a host remain separated from their previous bursts, preventing making of a large burst. Hence, packet-loss does not happen.

B. IBSP in Detail

In order to gurrantee zero packet-loss, each router first ensures that all bursts belonging to a timeframe (shown in Fig. 2) are transmitted in a single timeframe. It ensures this by first filling all these bursts into a buffer (in an outgoing port) and then transmitting these bursts in a single timeframe (i.e., emptying the buffer). To perform this filling and emptying of a buffer, we propose three buffers (each having the capacity to accommodate traffic of the timeframe duration i.e. the size is equal to $T_f \times bitrate$, where T_f is the timeframe duration) at each port in each router (the reason of using three buffers is explained in the next subsection). In addition, each router applies ping-pong buffering [8] for filling and emptying buffers. In ping-pong buffering, while one buffer is being filled, another buffer is emptied.

In IBSP, if the duration of timeframes is different in different nodes (routers and hosts), it may rise to slow input or fast input issues (discussed in the next subsection). In these issues, the current burst start chasing previous bursts. Therefore, packet-loss can happen. To solve this, the duration is assumed to be equal in all nodes in a network and is given by Eq. 3. In Eq. 3, there are $F(p)$ flows to be transmitted from port p and each flow, i.e., f_i has allocated an amount $s(f_i)$ (in bits) of data to be transmitted in each timeframe. Amount $s(f_i)$ in Eq. 3 depends on the SLA with the user of flow f_i and the $padding$ time (in Eq. 3) is described in the next subsection.

$$T_f = \max_{\forall p \ in \ a \ network} \left(\frac{\sum_{i=1}^{F(p)} s(f_i)}{bitrate}\right) + padding \tag{3}$$

The followings are the three main activities of IBSP.

1) **Timer-start activity**: Each node (router and host) starts a periodic timer of the T_f duration.
2) **Timer-expiry activity**: Each node (router and host) notifies the expiration of the T_f to its neighboring nodes by transmitting a control packet (end-of-frame, EOF).
3) **Inter-burst separation activity**: Each router ensures that bursts belonging to different timeframes on an incoming port are transmitted in separate timeframes on an outgoing port. Note that hosts do not need to perform this activity. Instead, these need to apply leaky bucket to ensure that offered traffic load is lower than (or almost equal to) the load that can be handled by the network.

[1] A host can actually transmit high-priority traffic (or best-effort traffic) from many users. However, for simplicity, we interpret a host as it is transmitting high priority traffic (or best-effort traffic) from one user.

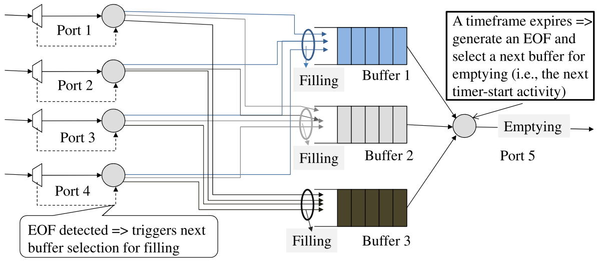
Port 1
Port 2
Port 3
Port 4
Filling
Buffer 1
Filling
Buffer 2
Filling
Buffer 3
A timeframe expires => generate an EOF and select a next buffer for emptying (i.e., the next timer-start activity)
Emptying
Port 5
EOF detected => triggers next buffer selection for filling

Fig. 3. Filling and Emptying buffers.

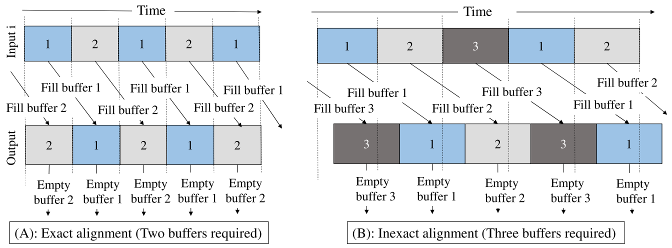

Fig. 4. IBSP Solution. Buffers are 1, 2, and 3. All rectangles are timeframes.

The timer-start activity can be implemented by running a periodic timer that calculates the T_f duration.

The timer-expiry activity can be performed by transmitting an EOF when the duration of a timeframe expires. The EOFs are used by neighboring nodes to know that all the packets received after an EOF on an incoming port belong to a different timeframe. Therefore, in the inter-burst separation activity, packets of different timeframes on an incoming port can be transmitted in separate timeframes on an outgoing port.

The inter-burst separation activity starts at the beginning of the timer-start activity and completes at the end of the timer-expiry activity. During this activity, each router performs two tasks: (1) fill a buffer and (2) empty a buffer. For filling a buffer of an outgoing port (see Port 5 in Fig. 3), IBSP depends on EOFs received from incoming ports (see Port $1, 2, 3, 4$). By default, in beginning, IBSP selects a buffer, which is not currently used for emptying. Then, each time when an EOF is received on a port, the next buffer (chosen in a round-robin fashion) is selected for filling packets from that port.

For emptying a buffer, IBSP depends on the timer-start activities. At the triggering of the first timer-start activity, IBSP selects one of the buffers (in each port) for emptying. It then selects the next buffer (chosen in a round-robin order) at the occurrence of the every next timer-start activity.

C. Justification of Using Three Buffers

IBSP proposes three buffers (instead of two) in each outgoing port. However, if input is exactly aligned with output (i.e., EOFs from incoming ports are received exactly at the same time when timer-start activities are triggered by IBSP), IBSP only requires two buffers. Fig. 4A illustrates such a case. It shows that two buffers are sufficient for implementing IBSP, as filling and emptying tasks (described in the previous subsection) can be performed on separate buffers all the time.

The problem is that it is not possible to make inputs aligned exactly with the output (i.e., due to the clock difference between two nodes, propagation delay, etc.). Because of this issue, it may happen that a buffer (in case of two buffers) is filled (i.e, EOF is received from an input) and the output is still transmitting from the other buffer (i.e., the current timeframe is not yet expired). So, at this time, no buffer can be used for filling. To solve the issue, we propose the third buffer in each port. Therefore, when an EOF is received while emptying a buffer, the input selector can be advanced to the next buffer, which will be available for filling (See Fig. 4B).

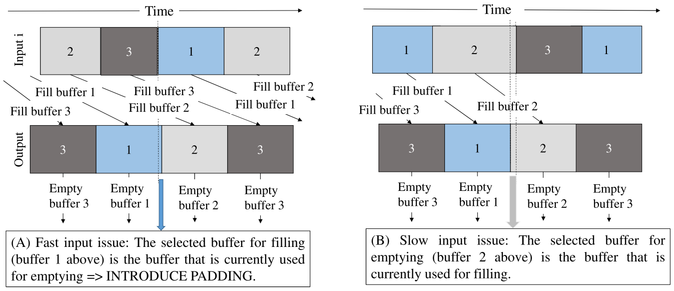

Fig. 5. Issues that may occur due to clock difference between nodes.

We propose only three buffers in each port to guarantee zero packet-loss. However, if more than three buffers (i.e., $n + 3$, where $n > 0$) are used, zero packet-loss can also be guaranteed using IBSP. But, in this case, n number of buffers will always be empty at all the time. Therefore, there is no advantage to have more than three buffers.

D. Solutions to the Issues of Our Approach

There are two additional issues that IBSP needs to address. These issues may occur due to clock difference between two nodes. This difference can be very marginal (e.g., in ppm).

1) The Next Selected Buffer for Filling Is the Buffer That Is Currently Used for Emptying: This issue may occur due to the fast clock of a neighboring node. In Fig. 5A, when the input selects buffer 1 for filling, the output is still emptying this buffer. To solve this issue, we propose *padding* (i.e., no data to be allocated) at the end of each timeframe. Therefore, when the issue occurs, the node should immediately expire the current timeframe and should start the next timer-start activity. As we propose that a node should not transmit any data during the padding time, the expiration of the timeframe before the actual timer expires will not cause any packet-loss. We propose that *padding* should be equal to the size needed to accommodate the clock difference due to the fast input (i.e., can be in ppm).

2) The Next Selected Buffer for Emptying Is the Buffer That Is Currently Used for Filling: This issue may occur due to the slow clock of a neighboring node. In Fig. 5B, when the output selects buffer 2 for emptying, the input is still filling this buffer. To solve this issue, we propose that the input should immediately switch to the next consecutive buffer for filling (in Fig. 5B, the next consecutive buffer is 3), when the issue occurs. In this case, when sufficient padding is present at the end of the received timeframe, no data will be received anymore before the receipt of the upcoming EOF, which will advance the buffer for this input once more. Thus, virtually a timeframe at the output will not contain any data for this input port (i.e., slow input port).

E. Delay and Jitter Using IBSP

IBSP has a disadvantage that it always adds some delay in packet forwarding, independent from the load (which in typical light load conditions may turn out longer). As each router first

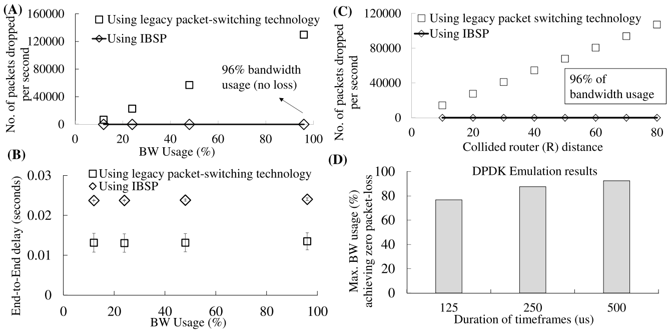

Fig. 6. (A) Packet drops vs. BW. (B) Delay and jitter vs. BW. (C) Packet drops vs. collided router distance. (D) DPDK emulation results.

fills a buffer (capacity of 1 timeframe) and then empties it, the delay per hop using IBSP can be of 1 timeframe duration. Furthermore, as due to a slow input the buffer for filling is forcefully moved to the next buffer, the maximum delay per hop could be 1 timeframe longer than the normal case. Similarly, the jitter (end-to-end) can be within 1 timeframe and is 1 timeframe longer than the normal case when the slow input catches up with fast output (does not happen very often). Nevertheless, as the size of timeframes can be as short as $125us$, the jitter and delay will be low. In addition, delay and jitter are bounded using IBSP.

IV. Experimental Study

We first performed NS-3 simulations and then performed emulations on DPDK. The worst-case scenario (Fig. 1) is tested. In experiments, there are five H_i and each host h_{ij} is connected through 4 ports with r_{ij}, emulating 4 different flows that generate interfering bursts to bursts of H_i.

A. Simulations

In the simulations, two different experiments are performed: (1) by varying the bandwidth usage (BW) in router R at port p (Fig. 6A and 6B) and (2) by varying the number of routers in the linear chain in Fig. 1 (Fig. 6C). The $ontime$ of bursts originated from H_i and h_{ij} is kept as $24us$, and the $offtime$ of bursts coming from H_i is kept as $101us$. Moreover, the $offtime$ of bursts coming from h_{ij} is varied according to the bandwidth usage of port p shown in Fig. 6. As there can be five small bursts in each router (including r_{ij} and R), the timeframes in IBSP should be at least $120us$ (i.e., $5 \times 24us$). We keep $5us$ as the padding time. Therefore, the duration of timeframes in our experiment is $125us$. We also perform simulations of legacy packet-switched networks where only a single buffer is used. To make a fair comparison between legacy packet-switched networks and networks using IBSP, the size of the buffer in legacy packet-switched networks is kept three times the single buffer size in IBSP. The propagation delay and bandwidth of each link is $100us$ and $1Gb/s$ respectively. In addition, the packet size is kept as 624 bytes.

Fig. 6A and 6C illustrate that there is packet-loss in legacy networks even at a low BW (i.e., 12% in Fig. 6A) and at a small distance (the number of hops) of collided router (i.e., R.) (Fig. 6C) respectively. It then increases with the increase in the BW and the collided router distance. However, there is no packet-loss using IBSP in both scenarios. Moreover, the end-to-end delay in IBSP is longer than the legacy network (see Fig. 6B). Furthermore, the jitter (error bars) is low in IBSP. In Fig. 6B, we see a little fluctuation in the average delay even-though the BW is increased. This is because we simulated the worst case scenario in which due to interference caused by h_{ij} at each r_{ij} for the bursts coming from H_i, the average end-to-end delay is approximately same for all received packets in all bandwidth usage (BW).

B. DPDK Emulations

IBSP requires a platform which can process packets (incoming and outgoing) at line rate. This is possible by implementing IBSP in: (1) hardware or (2) high performance software. We chose the second option and implemented IBSP in DPDK.

The difference between the emulation and simulation scenario lies in the number of routers in the linear chain and the size of bursts generated by hosts. There are 20 routers in emulations and the burst size is chosen according to the timeframe size and the bandwidth usage shown in Fig. 6D. In our implementation, due to timing inaccuracy in software, the padding time may include the time needed to process EOFs or the other packets. Therefore, we need to put additional padding in each timeframe to guarantee zero packet-loss. Fig. 6D confirms that using IBSP, the maximum of 76% of bandwidth can be used without having any packet-loss for short timeframes ($125us$). However, when the timeframe size is increased, the bandwidth can be used more than 90%, without having any packet-loss. In addition, the results also confirm (not shown in Fig. 6D) that the jitter and delay increases, when the size of timeframes increases.

V. Conclusions

In this article, we have proposed a novel protocol (i.e., IBSP) which guarantees zero packet-loss (with low jitter) in packet-switched networks in two conditions: (1) when the offered traffic load is less than (or almost equal to) the load that can be handled by the network and (2) padding in each timeframe is sufficient to accommodate the clock difference between two nodes. The results confirmed that legacy packet-switched networks cannot guarantee zero packet-loss, although the bandwidth usage is very low. However, using IBSP, zero packet-loss can be guaranteed, even though nearly all the bandwidth is consumed in the network.

References

[1] *Resilient Packet Rings (RPR)*, IEEE Standard 802.17, 2011, pp. 1–712.

[2] *IEEE Standard for Local and Metropolitan Area Networks—Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks—Amendment 17: Priority-Based Flow Control*, IEEE Standard 802.1Qbb-2011, 2011, pp. 1–40.

[3] G. R. de los Santos, M. Urueña, A. Muñoz, and J. A. Hernández, "Buffer design under bursty traffic with applications in FCoE storage area networks," *IEEE Commun. Lett.*, vol. 17, no. 2, pp. 413–416, Feb. 2013.

[4] T. Braun, M. Diaz, J. E. Gabeiras, and T. Staub, *End-to-End Quality of Service Over Heterogeneous Networks*. Springer, 2008. [Online]. Available: http://www.springer.com/us/book/9783540791195

[5] M. Berman *et al.*, "Future Internets escape the simulator," *Commun. ACM*, vol. 58, no. 6, pp. 78–89, 2015.

[6] T. G. Orphanoudakis, C. N. Charopoulos, and H. C. Leligou, "Leaky-bucket shaper design based on time interval grouping," *IEEE Commun. Lett.*, vol. 9, no. 6, pp. 573–575, Jun. 2005.

[7] J. Gettys and K. Nichols, "Bufferbloat: Dark buffers in the Internet," *Commun. ACM*, vol. 55, no. 1, pp. 57–65, Jan. 2012.

[8] Y.-M. Joo and N. McKeown, "Doubling memory bandwidth for network buffers," in *Proc. IEEE INFOCOM*, Mar./Apr. 1998, pp. 808–815.