Linear Network Coding Reduces Buffering in High-Speed Ethernet Parallel Transmission Systems

Xiaomin Chen*, Anna Engelmann*, Admela Jukan* and Muriel Médard**
Technische Universität Carolo-Wilhelmina zu Braunschweig, Germany*
Massachusetts Institute of Technology, USA**

***Abstract*—Parallel transmission is a known technique of transmitting flows over multiple paths from a source towards the same destination. In high-speed Ethernet standards, for instance, large bandwidth flows are inverse-multiplexed into multiple lower-speed flows and transmitted in parallel. However, when flows traverse different paths, attention needs to be paid to the resulting differential delay, which requires computationally expensive path optimizations and large buffering at the receiver. In this paper, we show analytically that linear network coding can significantly reduce the buffering in high-speed Ethernet systems at a price of en/decoding overhead, while relaxing the requirements on path optimality. We implement the proposed decoding buffer model according to the IEEE 802.3ba standard, and show that linear network coding reduces the buffer size up to 40% as compared to systems without coding. With linear network coding, input interfaces of the destination node can deploy relatively smaller buffers, which is critical for wider practical deployment of high-speed Ethernet systems at 100 Gbps and beyond.**

***Index Terms*—Multipath Routing, Linear Network Coding, Differential Delay, Buffering, High-speed Ethernet**

I. Introduction

Parallel (multpath) transmission is a known technique of transmitting unicast large flows over multiple paths from a source towards the same destination. In high-speed Ethernet, parallel transmission over multiple paths is a standardized solution in IEEE 802.3ba, enabling immediate capacity for large bandwidth flows (40/100 Gbps and more) by inverse-multiplexing into multiple lower-speed flows (e.g., 10 Gbps). From a practical perspective, parallel transmission makes networks highly backwards compatible: instead of upgrading to high-speed interfaces, the existing low-speed interfaces can be fully utilized to support large connections.

When multiple flows traverse different paths, attention needs to be paid to the resulting differential delay, which requires computationally expensive path optimizations and large buffering at the receiver. Frames arriving from different parallel paths need to be correctly aligned and multiplexed into a single flow at the receiver, which requires buffering. To this end, all related past work has focused on optimizing the splitting ratio among paths chosen and minimizing differential delay via optimizations, e.g., [1].

Unlike any previous work, this paper proposes to apply network coding in end-systems, and at the price of small coding overhead, to consequently lower the requirements on optimality of multipath routing, while reducing the buffer required for differential delay compensation, thus making the studied systems highly practically relevant. We analytically derive an upper bound of the buffer size required for decoding, and show that it is still smaller than the buffer size required by re-ordering in conventional systems without coding. We show with simulation results that linear network coding significantly reduces the buffer, up to 40%. The reduction of the overall buffer required is crucial for practical deployment of high-speed network systems, where buffering at multi-Gigabit capacity presents a major cost component.

II. System Model

The reference model is shown in Fig. 1. The source node follows the specification in IEEE 802.3ba, which scrambles Ethernet frames and groups the data into 66b data blocks. The data blocks are distributed to h Ethernet virtual lanes in a round robin fashion. A linear encoder is introduced which packetizes the data blocks, encodes and distributes the packets to N paths in parallel, $N \geq h$. The linear coding process is performed over a field F_{2^q}, where 2^q is the field size. Hence, traffic as a binary sequence is decomposed into *symbol* sequence with each symbol of the same length q. A *packet* [1] consists of N symbols and packets encoded with the same set of coding coefficients are referred to as in the same *generation*. At the destination, received packets are stored in a decoding buffer and processed by a decoder, which runs Gauss elimination over the received packets. In the decoding buffer, a virtual output queue (VOQ) is generated for each generation. The decoder is a batch processor [2] which checks all VOQs for complete generations with the decoding interval δt.

We adopt the network model from [3], where network is a directed and acyclic graph $G(V, E)$, with V and E as vertex set and edge set, respectively. Incoming links and outgoing links of a node $v \in V$ are denoted as $head(e_i) = v$ and $tail(e_j) = v$, respectively. The capacity of a network link is modeled as one packet (with generation ID in case of network coding) per time unit, and is chosen based on the link capacity of the physical link. For instance, if the physical network is with 10 Gbps per channel, and assuming finite field F_{2^8}, then the time unit is $0.8\ ns$ when a packet contains only a symbol. In case of a network with different link capacities, the greatest common divisor of link capacities will be chosen as the unit capacity. For instance, a link of 40 Gbps is modeled as four parallel links with unit capacity per time unit. Traffic originated from source node on lane i is as traffic generated by the process $X(s, i)$ with an identical and constant entropy of q bits per unit time. $Y^t(p_i)$ is the encoded packet on path p_i at time t. To model linear network coding in this system, a set

[1]We adopt a generic terminology of linear network coding and refer to a group of symbols as a packet. Each packet has the same number of symbols, which is a different concept from IP packets.

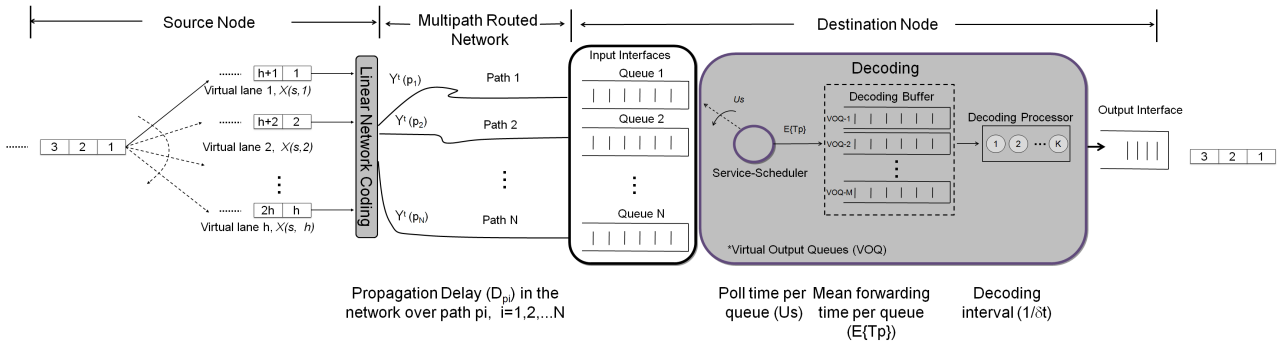

Fig. 1: Reference multipath routed system with parallel transmission and linear network coding

of virtual incoming links is introduced at the source node. In practical high-speed Ethernet systems, these virtual incoming links corresponds to multiple *lanes*. Packets are distributed to the virtual incoming links in a round robin fashion and encoded at the source node as defined in Eq. (1), i.e., the encoded packets on an outgoing link of s at time $t+1$ is denoted as $Y^{t+1}(e_j)$, $s = tail(e_j)$ is as follows:

$$\forall e_j : tail(e_j) = s : \quad Y^{t+1}(e_j) = \sum_{i:\mathcal{I}=\{1,2,\ldots,h\}} a_{i,e_j} \cdot X^t(s,i) \tag{1}$$

where $X^t(s,i)$ is the information sent out on lane i at time t by the process $X(s,i)$. The coding process at an intermediate node v at time $t+1$ is represented as follows:

$$\forall e_j : tail(e_j) = v : Y^{t+1}(e_j) = \sum_{e_i:head(e_i)=v} f_{e_i,e_j} \cdot Y^t(e_i) \tag{2}$$

The received information at time t on the output lane i is:

$$Z^t(d,i) = \sum_{e_j:head(e_j)=d} \sum_{u=0}^{\delta t+1} b_{i,e_j} \cdot Y^{t-u}(e_j) \tag{3}$$

Where a_{i,e_j}, f_{e_i,e_j} and b_{i,e_j} are randomly chosen from the finite field F_{2^q} and collected in the matrices $\mathcal{A}$, $\mathcal{F}$ and $\mathcal{B}$, respectively. A triple $(\mathcal{A},\mathcal{F},\mathcal{B})$ is referred to as a *linear network code* [4][2]. The received information is buffered at the destination and decoded by running Gaussian Elimination. When a generation is successfully decoded, it is immediately released from the decoding buffer. When the number of paths are larger than the generation size, i.e., $N \geq h$, we call the system has r *redundancy*, and $r = N - h$.

A. Assumptions and Preliminaries

The multipath routed systems of interest are based on the circuit switching networking with parallel transmission, such as high-speed Ethernet specified in IEEE 802.3ba. In such networks, transmission rate per circuit path can be defined. As discussed earlier, the link capacity is normalized to be a packet per time unit. We hereby assume that all input queues at the destination node are $D/M/1$ queues with identical arrival rate (λ and $\lambda = 1$) and job processing follows exponential distribution. Given that the output data rate of the destination is $h \cdot \lambda$, we assume that service scheduler runs h time faster than the arrival rate of the input queue. Therefore, there are almost no packets buffered at the input queues and the size of the input buffer does not affect the decoding. In addition, we define that the service scheduler starts from the queue with corresponding shortest path at time τ_0 when the first packets arrive at the destination. The notations used are as follows:

- μ_s: the mean service frequency of the service scheduler
- U_s: poll time per queue of the service scheduler
- $E\{T_p\}$: mean forwarding time of a packet
- τ_0: arrival time of the first packet at the destination
- T_{b_i}: time from τ_0 till the first packet on queue i is served
- D_{p_i}: propagation delay of path p_i
- M_{p_i}: buffer at the input interface required by p_i
- M_B: buffer required by re-ordering
- M_D: buffer required by decoding
- δt: decoding interval, i.e., the decoding process happens every δt time units
- $\tilde{p}$ and p': the longest path and the shortest path in the multipath routed system, respectively.

B. Upper Bound of the Decoding Buffer

The buffer in each input interface of the destination node is modeled as a First-In-First-Out (FIFO) queue. Queue i, denoted as Q_i, buffers packets from path p_i. When the system is steady, i.e., all input queues are non-empty, the cycle time of the service scheduler is:

$$T_{cycle} = 1/\mu_s = N \cdot (U_s + E\{T_p\}) \tag{4}$$

1) Buffer model before steady state: Different paths have different delays and thus exhibit different arrival time of the first packet at the destination. Before the steady state, service scheduler fetches $\sum_{i=1}^{N} x_i < N$ packets in each cycle, where x_i indicates if a packet is fetched from Q_i in this cycle. Let us denote the shortest among the multiple paths as p_1, i.e., $p' = p_1$ and at τ_0, the first packet arrives at Q_1 and $x_1 = 1$. For $i \geq 2$, the value of x_i at time t is determined by the differential delay between p_i and p_1:

$$x_i = \begin{cases} 1, & \text{if } t \geq D_{p_i} - Dp' \\ 0, & \text{else.} \end{cases}$$

Assume the system has run m cycles at time t, i.e., $t =$

[2] $\mathcal{A} = \{a_{i,e_j}\}$ is a $h \times N$ matrix contains all coefficients used at the source node; $i \in \mathcal{I}$ and $tail(e_j) = s$ $\mathcal{B} = \{b_{e_j,i}\}$ is a $h \times N$ matrix contains all coefficients used at the destination node; $i \in \mathcal{I}$ and $head(e_j) = d$. $\mathcal{F} = \{f_{e_i,e_j}\}$ is a $N \times N$ matrix contains the network information. $f_{e_i,e_j} = 1$ if link e_j has input information from e_i, otherwise, $f_{e_i,e_j} = 0$.

$\sum_{n=1}^{n=m} T_{cycle}^{n}$. The $m+1$ cycle, denoted as T_{cycle}^{m+1} is:

$$T_{cycle}^{m+1} = N \cdot U_s + \mathcal{X}^{m+1} \cdot E\{T_p\} \tag{5}$$

where $\mathcal{X}^{m+1}$ is the number of packets forwarded to the decoding buffer in the cycle T_{cycle}^{m+1}; $\mathcal{X}^{m+1} = \sum_{i=1}^{N} x_i$ with

$$x_i = \begin{cases} 1, & \text{if } \sum_{n=1}^{n=m} T_{cycle}^{n} \geq D_{p_i} - Dp' \\ 0, & \text{else.} \end{cases} \tag{6}$$

The number of packets that have to be stored in the decoding buffer till it starts decoding is then:

$$M_0 = \sum_{n=1}^{K} \mathcal{X}^n \tag{7}$$

where $K > \left\lceil \frac{D_{\tilde{p}} - D_{p'}}{T_{cycle}} \right\rceil$ and $\mathcal{X}^n$ is determined by Eq.(5) and Eq.(6) in each cycle.

Lemma 1. *The system has to run at least K cycles, $K > \left\lceil \frac{D_{\tilde{p}} - D_{p'}}{T_{cycle}} \right\rceil$, before $Q_N \neq \emptyset$, where Q_N corresponds to the longest path in the multipath routed transmission system.*

Proof: In the worst case scenario, the service scheduler has to poll and process $N-1$ queues before it reaches the queue of the longest path, i.e., $\tilde{p} = p_N$. As defined earlier, the time from τ_0 till the first packet from Q_N is forwarded is $T_{b_N} = \sum_{n=1}^{K} T_{cycle}^{n} + N\{U_s + E\{T_p\}\}$. According to Eq.(6), $Q_N \neq \emptyset$ if and only if $\sum_{n=1}^{n=K} T_{cycle}^{n} \geq D_{\tilde{p}} - Dp'$. And $\forall n, n \leq K$, $T_{cycle}^{n} < T_{cycle}$. Hence, $K \cdot T_{cycle} > \sum_{n=1}^{n=K} T_{cycle}^{n} \geq D_{\tilde{p}} - Dp'$. Therefore, $K > \left\lceil \frac{D_{\tilde{p}} - D_{p'}}{T_{cycle}} \right\rceil$. ∎

Lemma 2. *The buffer size required to start decoding is upper bounded, i.e., $M_0 = N \cdot (D_{\tilde{p}} - D_{p'})$, where $\tilde{p}$ and p' are the longest and the shortest path, respectively.*

Proof: According to Eq.(6), $Q_i \neq \emptyset$ if and only if at time $t_i \geq D_{p_i} - D_{p'}$. In the worst case, the decoding has to wait for the packets from the longest path, i.e., $\tilde{p}$ to complete the first generation, and $Q_N \neq \emptyset$ if and only if at time $t_N \geq D_{p_i} - D_{p'}$ Hence, all packets on path p_i arrive during $t_N - t_i$ have to be stored in the decoding buffer. Denote arrival rate of Q_i as λ_i, the number of packets forwarded from Q_i to the decoding buffer is $\lambda_i \cdot (D_{\tilde{p}} - D_{p_i})$. In our model, $\lambda_i = 1$ packet per time unit, therefore, during K cycles, $\sum_{i=1}^{N} (D_{\tilde{p}} - D_{p_i})$ packets are stored in the decoding buffer. $\forall p_i$, $(D_{\tilde{p}} - D_{p_i}) \leq (D_{\tilde{p}} - D_{p'})$, the upper bound is derived. ∎

2) Buffer model during the steady state: In every decoding cycle, the number of packets arrived at the decoding buffer is:

$$M_{\delta t} = \left\lfloor \frac{\delta t}{U_s + E\{T_p\}} \right\rfloor \tag{8}$$

$M_{\delta t}$ is composed of packets that can complete $\gamma_{\delta t}$ generations in the decoding buffer. After each decoding cycle, $\gamma_{\delta t}$ generations are completed and released from the decoding buffer. Hence, the decoding buffer is:

$$M_D = N \cdot (D_{\tilde{p}} - D_{p'}) + M_{\delta t} - N \cdot \gamma_{\delta t} \tag{9}$$

The worst case scenario exists in case of linear network coding when only one generation is decoded after each decoding cycle, i.e., $\gamma_{\delta t} = 1$. In the best-case scenario, packets received at each decoding interval can complete $\gamma_{\delta t} = M_{\delta t} \cdot \frac{h}{N}$ generations, where h is the generation size, $h \leq N$.

3) Lower bounds of the buffer required in absence of linear network coding: When all the components related to linear network coding (marked in shadow) are removed, i.e., the linear network coding block at the source node and decoding block at the destination node, the system corresponds to the conventional multipath routing. We now derive the buffer bounds for this case, and compare to the results obtained above. For a fair comparison, the packets here are the same as the packets used in a linear network coding system, i.e., packets with fixed size. Upon arrival, each packet is routed to one of the N queues accordingly, i.e., packets transmitted on path p_i is routed to queue i of interface i. A packet leaves a queue only if all packets that are sent earlier have been served. This ensures that all packets leave the system in the correct order, as they were sent out from the source node.

We consider the worst case scenario, where the path delay is in descending order of packet order. Denote the delay of path p_i as D_{p_i}, then the worst case scenario is $D_{p_1} \geq D_{p_2} \geq \cdots \geq D_{p_N}$. At τ_0, the first packet routed on the shortest path p_N arrives at the system. Packets have to be buffered on each input interface till the first packet on the longest path, i.e., p_1 is served. T_{b_1} is defined as:

$$T_{b_1} = \left\lceil \frac{D_{p_1} - D_{p_N}}{NU_s} \right\rceil \cdot NU_s + U_s + E\{T_p\}. \tag{10}$$

All other input queues cannot be served when the first packet on the first queue has been served. The time that the first packet on all other queues has to wait till it is served is:

$$T_{b_i} = T_{b_{i-1}} + U_s + E\{T_p\}, i = 2, 3, ..., N. \tag{11}$$

From Eq. (10) and Eq. (11), we can derive the time till the first packet on queue N is served as:

$$T_{b_N} = \left\lceil \frac{D_{p_1} - D_{p_N}}{NU_s} \right\rceil \cdot NU_s + N(U_s + E\{T_p\}). \tag{12}$$

In practice, the longest path can be connected to any input interface. Hence, the buffer required by the path p_N, i.e., the shortest path, is the minimum buffer size on each input interface to ensure the packets processed in the right order. Let us denote the size of queue N as M_{p_N}:

$$M_{p_N} = \left\lceil \frac{D_{p_1} - D_{p_N}}{NU_s} \right\rceil \cdot NU_s + N(U_s + E\{T_p\}). \tag{13}$$

Total buffer size required by the reordering M_B is the sum of all the buffers on all input interfaces, i.e., $M_B = N \cdot M_{p_N}$. To generalize the buffer model, let us denote the longest path as $\tilde{p}$ and the shortest path as p'. We can derive a lower bound of the total buffer required by a multipath routed system, i.e.,

$$M_B = N \cdot \left\{ \left\lceil \frac{D_{\tilde{p}} - D_{p'}}{NU_s} \right\rceil \cdot NU_s + T_{cycle} \right\}. \tag{14}$$

III. Evaluation

A. Analytical results

Fig. 2 illustrates the buffer bounds required by re-ordering and by linear network coding. The maximum differential delay $D_{\tilde{p}} - D_{p'}$ is set to 125 $timeunits(tu)$ and the parallel transmission is set to use 10 paths; T_{cycle} is 1 tu. Fig. 2 shows the impact of the decoding interval δ_t on the buffer size. The size of decoding buffer, M_D (Eq. (9)) is normalized by the re-ordering buffer M_B (Eq. (14)).

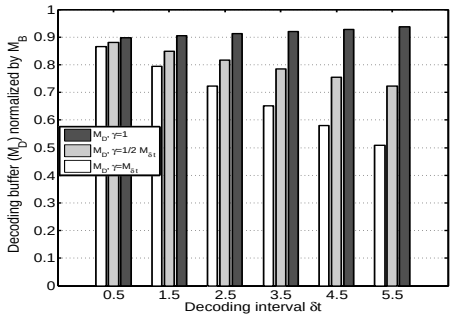

Fig. 2: Decoding buffer M_D normalized by M_B vs. decoding interval δ_t ($T_{cycle} = 1\ tu$, $D_{\tilde{p}} - D_{p'} = 125\ tu$, $N = 10$)

Per Eq. (9), the size of decoding buffer depends on the number of generations (γ_{δ_t}) decoded in each decoding interval. In the worst case, the decoder can only decode one generation in each decoding interval, i.e., $\gamma_{\delta_t} = 1$. As shown in Fig. 2, linear network coding reduces about 10% buffer size when $\delta_t = 0.5$ with $\gamma_{\delta_t} = 1$. However, the decoder can generally decode and release multiple generations that are complete in the decoding buffer, further reducing the size of M_D. When the decoder run faster than the service scheduler, i.e., $\delta_t < T_{cycle}$, the possibility of getting more than one complete generation from the decoding buffer is low. As shown the case of $\delta_t = 0.5\ T_{cycle}$, larger γ_{δ_t} only slightly reduces the buffer. When the decoding interval is larger than T_{cycle}, more packets are stored in the decoding buffer, increasing the possibility of decoding multiple complete generations. When the decoding interval is large, on the other side, e.g., $\delta_t = 5.5\ tu$, a large number of generations are decoded and released, resulting in a small decoding buffer. The minimum size of decoding buffer is as defined in Eq.(7). Fig. 2 shows that linear network coding can reduc the buffer size without an extremely fast decoder.

B. Case study: High-Speed Ethernet

We now implement the proposed buffer model for a case study of high-speed Ethernet system, using an event-driven simulator written in Java. The parameters used are summarized in Tab. I. In the simulation, the differential delay between a channel p_i and the shortest path is defined $i \cdot (125/4)\ \mu s\ i = 1, 2, .., 4$. For instance, assume the shortest path is denoted as p_5, then the differential delay between p_1 and p_5 is $125/4\ \mu s$. Generally, the generation size is equal to the number of paths in our model. When the generation size is smaller than the number of paths, we refer to as parallel transmission with redundancy. For instance, if the generation size is 4 and we use five paths, this redundancy of one more path can be used against link failures as a spare path, and also for fast decoding.

We define the *throughput* as the percentage of successfully decoded generations in all generations that are originated from the same sender node. Fig. 3 shows the throughput as a function of the size of decoding buffer. We can see that 100% throughput can be achieved using 60% of M_B (as calculated in Eq. (14)). 90% generations can be successfully decoded when the decoding buffer is only half of M_B. When the generation size is smaller than the number of paths, the required buffer is always reduced. For a larger path redundancy, for instance, for $h = 4$ and $h = 3$, 50% and 60% of buffer reduction can be achieved.

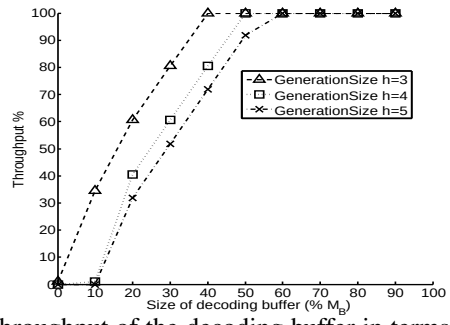

Fig. 3: Throughput of the decoding buffer in terms of percentage of successfully decoded generations

Parameter	Value
Field size	2^8
Packet size	6 symbols with 8 bits/symbol
Generation ID	2 bytes
Number of generations	150,000
Number of parallel channels	5
Maximal differential delay	125 μs
TransmissionRate/channel	10 Gbps, i.e., time unit=$6.4ns$
Buffer without coding	$M \approx 1MB$ ($\approx$ 195,313 packets)
Size of each input queue	10 packets

TABLE I: Summary of Parameters

Discussion on coding overhead vs. differential delay. Successful decoding requires a complete generation to obtain the full rank of the coding matrix. The differential delay issue in parallel transmission necessitates buffering of all packets for one generation to compete. The generation ID attached to each packet is then the coding overhead. Thus, in the worst case, the number of generations that need be stored is $(D_{\tilde{p}} - D_{p'})/TimeUnit$, each requiring a generation ID. As two bytes are sufficient to distinguish 65,536 generations, this overhead is a small price to pay in comparison to the buffer size savings, especially with large packet sizes.

IV. Conclusion

In this paper, we showed analytically that linear network coding can significantly reduce the buffer required in parallel transmission over multipath routed network. The case study of high-speed Ethernet standard IEEE 802.3ba showed a reduction of 40% of the buffer size with linear network coding. With linear network coding, input interfaces of the destination node can deploy very small buffers, which is critical to practical implementation of high-speed Ethernet.

References

[1] X. Chen and A. Jukan, "Optimized parallel transmission in OTN/WDM networks to support high-speed Ethernet with multiple lane distribution," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 4, no. 3, pp. 248–258, Mar. 2012.

[2] H. Gold and P. Tran-Gia, "Performance analysis of a batch service queue arising out of manufacturing system modelling," *Queueing Systems*, vol. 14, no. 3-4, pp. 413–426, 1993.

[3] R. Koetter and M. Medard, "An Algebraic Approach to Network Coding," *IEEE/ACM Transactions on Networking*, 2003.

[4] T. Ho, R. Koetter, M. Medard, D. Karger, and M. Effros, "The Benefits of Coding Over Routing in a Randomized Setting," in *IEEE International Symposium on Information Theory*, 2003.