Calendar Allocation Based on Client Traffic in the Flexible Ethernet Standard

Yun Liao*, Seyyed Ali Hashemi*, Hesham ElBakoury†, John Cioffi*, Andrea Goldsmith*
*Department of Electrical Engineering, Stanford University, USA
†Futurewei Technologies, USA
yunliao@stanford.edu, ahashemi@stanford.edu, helbakoury@gmail.com, cioffi@stanford.edu, andrea@wsl.stanford.edu

***Abstract*—An adaptive bandwidth allocation mechanism for the calendar associated with the Flexible Ethernet (FlexE) standard is proposed. The proposed method bases the FlexE calendar design on the clients' real transmit data rates. In particular, the proposed method treats clients with very low bandwidth utilization as minor clients and allows them to transmit in an opportunistic manner. Experiments on real Ethernet packet traces indicate that by using the proposed calendar scheme to allocate bandwidth to clients, the total required FlexE bandwidth can be reduced by up to 60% while meeting packet drop requirements.**

I. Introduction

The Ethernet standard is consistently evolving to improve device connectivity over local area networks (LANs). In comparison with wireless connections, wired LANs based on the Ethernet standard provide better stability and reliability, augmented security, and easier scalability to larger networks. In order to communicate an increasing amount of information using the Ethernet standard, there is a growing demand for new and efficient interfaces to the links in Ethernet-based LANs [1]. There have been several efforts to provide adaptive physical layer (PHY) interfaces within the Ethernet standard that can flexibly change parameters with respect to the modulation schemes and the data rates [2], [3]. However, these methods are not compatible with the existing routers that have multiple fixed-rate Ethernet PHY modules. In addition, switching between different links with different capacities should be performed to exploit full network capability.

Flexible Ethernet (FlexE) was recently developed and standardized to break the limitation of only supporting media access control (MAC) rates that match existing PHY interface capacities [4]. FlexE not only efficiently aggregates different PHY modules to support higher MAC data rates, but also provides a mechanism to allow multiple MAC clients with different data flows to be able to transport data through a single or multiple PHY links. The key advantage of FlexE is that it provides flexibility without the need for standardization of new PHY interface capacities. As a result, it has been used in a number of applications such as optical transport networks [5] and network slicing for the fifth generation of cellular communications (5G) [6].

To dissociate the MAC client data rate and the Ethernet PHY capacity, FlexE uses a shim that is placed between the MAC and PHY layers to coordinate and distribute the client data flows into PHY instances. Inside the shim, the PHY bandwidth is divided into blocks, and a calendar is used to specify the assignment of MAC clients to the blocks. There are two main issues associated with FlexE: first, the calendar slots are allocated to the clients according to the bandwidth of the physical links that carry them. However, this can be too conservative in practice because the majority of the clients do not transmit at rates close to the link capacity. Second, the mechanism for handling MAC clients with data rates that are much lower than the rate of a calendar slot is not efficient. In fact, if a MAC client transmits data at a very low rate, one straightforward approach is to assign a calendar slot to that MAC client and use dummy data to fill the slot, which leads to significant bandwidth loss. Another method is to use a MAC frame buffer to collect the low-rate data, and transmit when the buffer is full [4]. However, this approach introduces long delays in data transmission through the network.

The proposed method bases the FlexE calendar allocation scheme on the clients' real bandwidth utilization information. This flexible allocation lowers the FlexE link bandwidth requirement and accommodates fluctuations in the data flow. In particular, the proposed method treats the clients with very low data rates or low bandwidth utilization as *minor clients* and allows them to transmit in an opportunistic way instead of getting designated calendar slots. As a result, FlexE bandwidth utilization is significantly improved. For the *major clients* with high data rates, two bandwidth estimation methods are introduced based on their historical data rates. The proposed schemes are evaluated on real Ethernet packet traces and the results show that the proposed flexible calendar allocation schemes effectively accommodate the fluctuations in clients' data rate. In particular, the proposed scheme reduces the total bandwidth requirement by up to 60% compared to the current FlexE standard. Furthermore, when a buffer with limited size is used at the FlexE shim where the current FlexE standard fails to provide the desired packet drop rate, the proposed scheme can automatically determine the amount of increase in the bandwidth requirement and keep the packet drop rate within a small range of values. To the best of the authors' knowledge, this is the first paper to investigate dynamic FlexE calendar allocation based on the clients' dynamic data rates. This work shows the potential of FlexE to serve significantly more clients than the current standard.

The rest of this paper is structured as follows. Section II provides a brief background on the current FlexE standard.

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

In Section III, the proposed FlexE calendar allocation scheme is described and the bandwidth estimation methods are introduced. Experiments and results are presented in Section IV. The paper is concluded in Section V.

II. Flex Ethernet

FlexE is an implementation agreement by Optical Internetworking Forum (OIF) that provides a mechanism to support MAC client rates that do not necessarily correspond to available PHY rates [4]. This mechanism allows for flexible data rate allocation through three different schemes:

- *Bonding*: Allows a MAC client to transmit data at a rate that is higher than the available PHY rate by bonding several PHY instances. For example, two PHY instances that can support a data rate of 100 gigabits per second (100G) can be bonded to support a MAC client with a data rate of 200G.
- *Channelization*: Allows multiple MAC clients to transmit data through a single or a number of bonded PHY instances. For example, two MAC clients each having a data rate of 50G can transmit data through a single PHY instance that can support a data rate of 100G.
- *Sub-rating*: Allows a MAC client with a lower data rate than the data rate of a single or a number of bonded PHY instances to transmit data through them. For example, a MAC client with a data rate of 75G can transmit data through a PHY instance that can support a data rate of 100G.

A combination of the aforementioned three schemes is also foreseen in the FlexE standard.

Fig. 1 illustrates the architecture of FlexE with examples of bonding, channelization, and sub-rating. There are three main components in the FlexE architecture: MAC clients, FlexE shim, and FlexE group. The MAC clients aggregate data in blocks of 64 bits and encode them into 66 bits (64b/66b blocks). These 64b/66b blocks are then transmitted towards the FlexE shim as a bit stream at the data rates of 10G, 40G, or integer multiples of 25G. The FlexE shim at the transmitting side is responsible for allocating the 64b/66b blocks that are received from MAC clients into the FlexE group. At the receiving side, the FlexE shim is responsible for assigning the 64b/66b blocks from the FlexE group to their intended MAC clients. Therefore, the FlexE shim at the transmitting side is called the FlexE mux and at the receiving side it is called the FlexE demux. The FlexE group consists of a number of Ethernet PHY instances with each PHY being capable of supporting a data flow of either 100G, or 200G, or 400G. It should be noted that the 200G and the 400G PHYs are constructed using two and four bonded 100G PHYs respectively.

The FlexE mux assigns the 64b/66b blocks that are received from the MAC clients to a logical serial stream of 64b/66b blocks according to a calendar. For each of the 100G PHY instances in a FlexE group, a sub-calendar is allocated that has 20 slots with each slot having a bandwidth of 5G. Therefore, for a FlexE group with M 100G PHY instances,

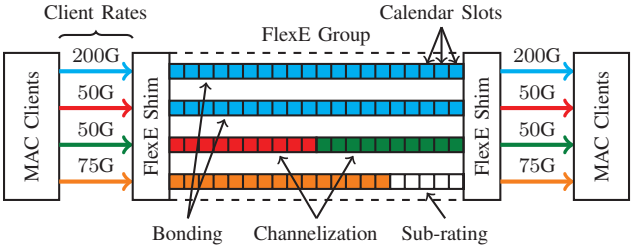


Fig. 1: FlexE architecture.

the calendar has a length of $20M$. The sub-calendars are sequentially distributed to the PHY instances in a FlexE group. To synchronize the reception of data at the FlexE demux, an overhead block is inserted into all of the sub-calendars in the calendar every 1023 transmissions of the calendar. This corresponds to having an overhead block for every 1023×20 data blocks for a given PHY instance.

III. FlexE Calendar Allocation based on Client Data Rates

In the current standard, the FlexE shim allocates calendar slots to each client according to the PHY rate of the physical link that carries the client's data. Despite its simplicity, this scheme results in a rigid calendar structure that cannot efficiently accommodate the fluctuations in the data flows. Furthermore, when multiple low-rate clients coexist with high-rate ones, the traffic of the low-rate clients is buffered and transmitted once in several calendars. This requires buffers large enough to store the packets received from low-rate clients, which consequently incurs a long delay. This section introduces a flexible FlexE calendar-allocation scheme that improves the FlexE utilization efficiency, and reduces the delay experienced at low-rate clients.

A. Calendar Allocation Scheme

To improve the efficiency of the high-speed FlexE link, the new method evaluates the transmit data rates of the FlexE clients, and uses a threshold to label the clients as either major clients or minor clients accordingly. A client is labeled "minor" if its real transmit data rate is lower than the assigned threshold and it is labeled "major" otherwise. The threshold is significantly lower than the PHY capacity in order to distinguish between the minor and major clients.

In the proposed scheme, the FlexE shim allocates calendar slots *only* to the major clients in accordance with the estimates of their required bandwidth. The minor clients do not get designated calendar slots. Instead, they transmit in an opportunistic manner. In other words, the minor clients transmit data either when the major clients do not completely use their designated calendar slots, or when the total number of designated calendar slots to the major clients is less than the number of available calendar slots. When there are enough calendar slots available for the minor clients, they transmit in a first-come-first-served manner. The proposed method does not take into consideration the heterogeneous delay requirement

between the clients or the packets for simplicity. However, the proposed method can readily accommodate such delay requirement by either treating the clients with stricter delay requirements as major clients, or treating them as minor clients and prioritizing them in the queue where minor clients transmit their data.

The proposed calendar allocation scheme follows the current FlexE standard that each calendar contains 20 slots, which are allocated to the clients. However, this allocation relaxes the FlexE PHY bandwidth constraints that are limited to integer multiples of 100G. Instead, the new method estimates the total bandwidth to transmit all clients' data and uses this estimate as a metric to evaluate the proposed calendar allocation scheme's performance. In fact, this bandwidth requirement can be directly translated to the number of clients that can transmit data on a high speed link, considering the link has a fixed bandwidth similar to the FlexE case. Additionally, adjustments to the proposed calendar allocation scheme occur periodically every T seconds.

The FlexE shim will take N major clients and estimate the required bandwidth for the i-th major client during the time period s with duration T, as $\tilde{B}_i(s)$. Further, the fixed bandwidth of each calendar slot will be r. The number of calendar slots allocated to the i-th major client during the time period s can thus be calculated to be at least $\left\lceil \frac{\tilde{B}_i(s)}{r} \right\rceil$, where $\lceil \cdot \rceil$ denotes the ceiling function. To ensure that the total number of calendar slots allocated to the major clients does not exceed 20, for every time period s, r needs to satisfy

$$\sum_{i=1}^{N} \left\lceil \frac{\tilde{B}_i(s)}{r} \right\rceil \leq 20. \tag{1}$$

In addition, for every time period s,

$$\frac{\sum_{i=1}^{N} \tilde{B}_i(s)}{r} \leq \sum_{i=1}^{N} \left\lceil \frac{\tilde{B}_i(s)}{r} \right\rceil < \frac{\sum_{i=1}^{N} \tilde{B}_i(s)}{r} + N, \tag{2}$$

where the fact that $x \leq \lceil x \rceil < x + 1$ for any x is used.

Let $B = \sup_{\forall s} \{ \sum_{i=1}^{N} \tilde{B}_i(s) \}$. Then, the minimum bandwidth requirement per calendar $r_{\min}$ satisfies

$$\frac{B}{20} \leq r_{\min} < \frac{B}{20 - N}. \tag{3}$$

Consequently, the total required bandwidth to carry the data for all major clients is in the range of

$$B \leq 20 r_{\min} < \frac{20B}{20 - N}. \tag{4}$$

A binary search in the above interval is used to find the total required bandwidth $20 r_{\min}$.

Each major client has at least one calendar slot in the shim, and there are 20 slots in one calendar in total. As a result, the derivation in (4) implicitly assumes that the number of major clients N is no larger than 20.

B. Required Bandwidth Estimates for Major Clients

This subsection provides several methods to estimate each of the major client's required bandwidth. Each major client carries data over its own Ethernet link before entering the FlexE shim. Therefore, it is possible to estimate each client's required bandwidth separately before entering the FlexE shim.

The current FlexE standard directly uses the clients' PHY link bandwidth to guarantee that each major client has sufficient bandwidth in any time period. While simple, this strategy is very inefficient because usually the major clients do not fully use the link bandwidth all the time. Based on this observation, the proposed calendar allocation depends on each client's measured data rate and allows dynamic adjustments to the calendar allocation as the major clients' data-flow pattern changes.

1) Bandwidth estimates based on the last peak data rate: The first proposed strategy uses the peak data rate of each client i during time period s to estimate that client's required bandwidth in time period $s + 1$. The peak data rate of client i during the s-th time period is $\hat{R}_i(s)$. The strategy uses the following expression to estimate $\tilde{B}_i(s + 1)$:

$$\tilde{B}_i(s + 1) = f(\hat{R}_i(s)), \tag{5}$$

where $f(\cdot)$ can be any function. One such choice is a simple linear function $f(x) = \alpha x$, with parameter $\alpha \geq 1$ to estimate $\tilde{B}_i(s + 1)$. The strategy allows changes in the calendar slot-allocation for every time period of duration T. Since this strategy is of low complexity, it can be used in applications where real-time bandwidth estimation is desired.

2) Bandwidth estimates based on model-driven data rate forecasting: Instead of using the peak data rate from one period before to estimate the required bandwidth, a variant of this strategy could review a longer history of the client's data rate to forecast the data rate during future periods and then use the prediction to determine the required bandwidth. Several time-series analysis methods could be adopted to forecast the major clients' data rates. This work adopts the autoregressive integrated moving average (ARIMA) model [7] for the forecasting task, which is known to be the most general class of models for forecasting a discrete time-series.

An ARIMA(p, d, q) model has three parameters where the lag order p shows the number of lag observations included in the model, the degree of differencing d is the number of times that the raw observations are differenced, and the order of moving average q specifies the size of the moving average window.

Definition 1 (ARIMA(p, d, q) model): Consider $\{X_t | t \in \mathbb{Z}\}$ as a discrete time-series and L as a time lag operator with $L(X_t) = X_{t-1}$. The ARIMA(p, d, q) model is a noisy discrete-time linear equation of the form

$$\left(1 - \sum_{k=1}^{p} \gamma_k L^k\right) (1 - L)^d X_t = \left(1 + \sum_{k=1}^{q} \beta_k L^k\right) \epsilon_t, \tag{6}$$

where γ_k and β_k are model coefficients, and ϵ_t denotes the white noise at time stamp t.

From the above ARIMA(p, d, q) model, the one-step forecast at time stamp t is given as

$$\hat{X}_t = \frac{\mu + \sum_{k=1}^{p} \phi_k L^k (1-L)^d X_t - \sum_{k=1}^{q} \theta_k L^k e_t}{(1-L)^d}, \quad (7)$$

where μ, ϕ_k, and θ_k are coefficients that can be fitted from the data, and $e_t = \hat{X}_t - X_t$ is the prediction error at time stamp t. Note that by adding the prediction error terms, the model automatically incorporates all the historical data up to time stamp t in the prediction.

The ARIMA model provides a prediction of the data rate at a certain time stamp t. However, a forecast of the bandwidth is required during a relatively long time period T. Therefore, prediction at a single time stamp may not result in the desired performance. To further improve the accuracy of the estimates, the time stamp t is sampled from the time period s, i.e., the time interval between two samples is T/W, where $W > 1$ is an integer. Then, the ARIMA model makes predictions for the next W time steps and selects the highest predicted data rate among the W predictions as the estimated peak data rate $\hat{R}_i(s)$ in time period s. Similar to the previous strategy, a scaling parameter α is selected and $\alpha\hat{R}_i(s)$ is used as the required bandwidth for the i-th major client.

Note that since the prediction algorithms are of low complexity [1], they can use the same processor as the PHY layer algorithms and no off-chip controller is needed.

C. Buffer

The proposed calendar allocation schemes allow the aggregation of FlexE clients whose total link capacity exceeds the FlexE link bandwidth. Based on the observation that the vast majority of Ethernet links are not running near their capacity, the proposed adaptive calendar allocation schemes can reduce the required FlexE bandwidth. However, the bandwidth saving may cause occasional short-term bandwidth over-subscription, where the instantaneous bandwidth of the clients becomes larger than the allocated bandwidth. To tackle this issue, a buffer is used to absorb the instant spikes in the aggregated input data flow and to avoid data loss. Since a buffer is already foreseen to be part of the FlexE standard that can be used for this purpose, the proposed scheme does not introduce the need for an extra buffer.

The buffer size is crucial to the design of the proposed calendar allocation scheme. A large buffer can effectively suppress the instant spikes in the ingress and thus allows for further reduction in the bandwidth requirement. By contrast, a small buffer can be easily filled with the occasional ingress spikes, thus requiring a large bandwidth to avoid potential packet loss caused by buffer overflow. The buffer size is characterized by the time it takes to fill the buffer with data. For example, a 1ms buffer for a 100G link corresponds to the buffer size of $100\text{G} \times 1\text{ms} = 100\text{Mb}$.

[1]The complexity of fitting an ARIMA model is $\mathcal{O}(N)$, where N is the length of the training sequence, and the complexity of prediction is $\mathcal{O}(1)$.

TABLE I: Statistics of packet traces

Trace ID	Capture date	Start time	# of packets	Ave rate (Mbps)	Standard deviation (Mbps)
1	2018/05/09	04:29	55.30M	529.35	188.99
2	2018/05/09	04:44	50.11M	499.15	162.95
3	2018/05/09	05:44	50.30M	504.63	180.59
4	2019/04/09	04:14	69.40M	852.77	321.55
5	2019/04/09	05:00	89.80M	1118.01	512.06
6	2019/04/10	14:00	271.97M	2796.28	552.85
7	2019/04/17	14:00	260.89M	2555.38	390.31
8	2019/04/24	14:00	279.92M	2761.80	438.75
9	2019/05/01	14:00	165.64M	2117.70	568.47
10	2019/05/08	14:00	254.92M	2510.26	390.20

IV. Experiments and Results

This section describes the experimental settings and analyzes the performance of the proposed FlexE calendar slot-allocation schemes in terms of the required bandwidth at the FlexE shim and the packet drop rate experienced by the clients.

A. Dataset

Since there are no public datasets for FlexE traffic flows, the results here use 15-minute-long packet traces from the WIDE MAWI archive [8]. The traces are captured from a 10G link using a commodity server. To simulate a FlexE link with multiple clients, 10 packet traces captured in 2018 and 2019 are aligned and treated as 10 different clients. The detailed information of the selected 10 traces are shown in Table I. The results here use only the start time and the number of packets that are provided in Table I. Since the packet traces are captured on 10G links and in accordance with the current FlexE standard, a 100G FlexE link will be required to carry all these 10 clients.

B. Experimental Settings

For both of the proposed flexible FlexE calendar allocation schemes, the FlexE shim changes the calendar allocation every second, i.e., $T = 1$. For the rate forecasting with the ARIMA model, the prediction interval t is set to 0.1s. This setting then allows the ARIMA model to predict $T/t = 10$ time steps ahead and to use the 10 predictions to determine the bandwidth to be designated to each major client.

For the rate forecasting with the ARIMA model, since different models ARIMA(p, d, q) give significantly different forecasts, the results conduct a grid search on parameters (p, d, q) for each major client, test the prediction's reliability on an arbitrarily selected 20-second period, and select the best set. The search range of the parameters is $p \in \{0, 1, 2, 4, 6, 8\}$, $d \in \{0, 1, 2\}$, and $q \in \{0, 1, 2\}$. The measuring metric for reliability is

$$\sigma = \mathbb{E}\left[\left[1 + 4.5\text{sgn}(\hat{X}_t - X_t)\right]\left(\hat{X}_t - X_t\right)^2\right]^{1/2}, \quad (8)$$

which is the weighted mean squared error and penalizes more underestimates than overestimates.

The tests of packet drop rates under all schemes are conducted on an arbitrarily selected 20-second period. The test period does not overlap with the optimization period for the

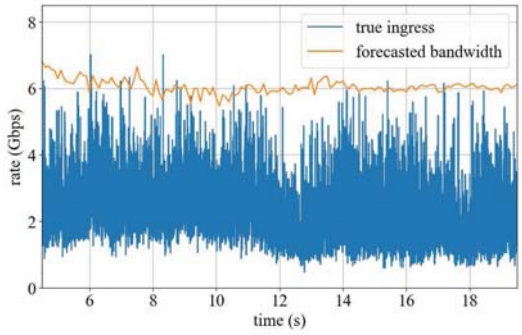


Fig. 2: Required bandwidth forecast by the ARIMA model.

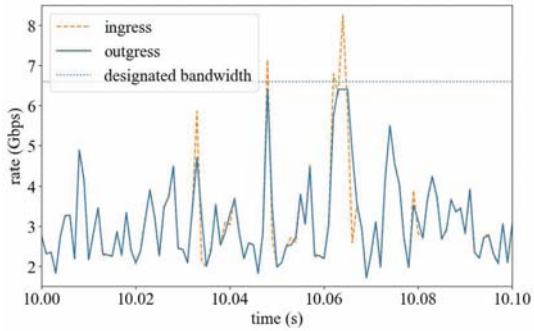


Fig. 3: Detailed illustration of ingress and outgress of one major client.

ARIMA models. The selected test period contains 33.1 million ingress packets.

C. Results and Analysis

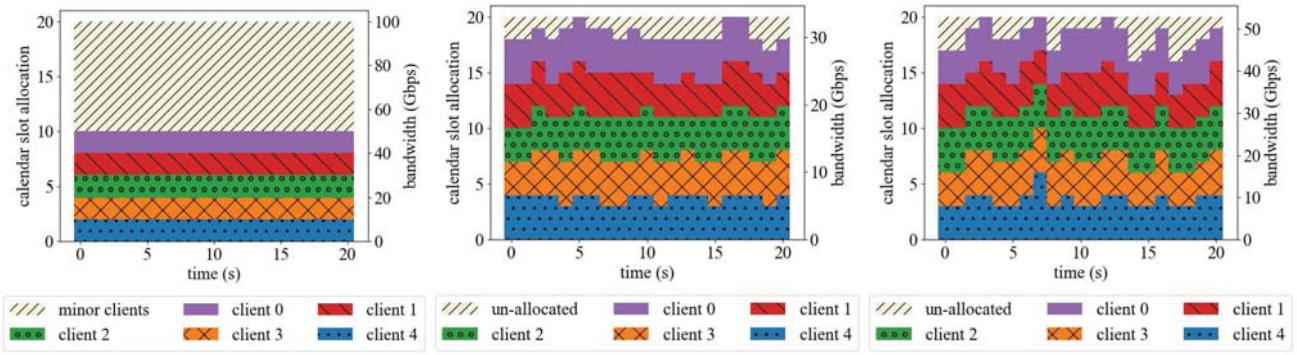
Fig. 2 shows the data rate forecast of the second major client given by the ARIMA$(8, 1, 1)$ model. It can be seen that the real incoming data rate may occasionally exceed the forecast bandwidth as short spikes, but it does not remain above the forecast for long periods of time. This indicates that the forecast given by the ARIMA model can be treated as a safe estimate of the rate of the incoming data flow.

Fig. 3 shows the detailed ingress and outgress data rate of one major client (the one corresponds to trace ID 1). This experiment uses the calendar allocation scheme with the ARIMA forecast model, and the buffer size is 1ms. During the time window in the plot, the client gets three designated calendar slots, which corresponds to 6.6G bandwidth. The figure shows that the instant ingress data rate sometimes exceeds the allocated bandwidth, but the data can still be successfully transmitted after a short delay in the buffer. To make the plot clean, the ingress and outgress rates shown are for every 1ms. Therefore, the ingress curve remaining under the designated bandwidth does not necessarily mean that the instant ingress rate also remains under the bandwidth. This explains why there are a few mismatches on the ingress and outgress curves (for example, around 10.03s) even when they both remain under the designated bandwidth. This figure shows that using a proper packet buffer at the FlexE shim can potentially suppress the bandwidth allocated to clients to a level that is even smaller than their peak data rates without any packet loss.

Fig. 4 illustrates the calendar slot-allocation schemes based on the last peak data rate. The left y-axis shows the number of calendar slots allocated to each client, and the right y-axis indicates the corresponding bandwidth. The bandwidths shown in Figs. 4b and 4c are tuned through variation of the scaling factor α to ensure zero packet drop in the test set. The flexible schemes in Figs. 4b and 4c show the irregular calendar allocation schemes. These schemes sometimes leave several calendar slots un-allocated, which can be used to transmit minor clients' packets. Also, with proper buffer size, the flexible calendar allocation schemes require significantly less total bandwidth to carry all clients. For example, if the FlexE shim is equipped with a 1ms buffer, the required bandwidth can be reduced by more than one half compared to the standard scheme as in Fig. 4a. Furthermore, comparison of Fig. 4b and Fig. 4c shows not only larger total required bandwidth in Fig. 4c, but also more fluctuation in the calendar slot-allocation over time. This variation is expected because a smaller buffer has less capability to absorb the bandwidth over-subscription, or spikes in incoming flows. A FlexE link with larger bandwidth is required to guarantee that the data pass through smoothly, and the calendar needs to accommodate the fluctuations in the data flow more carefully so that the clients can make efficient use of the bandwidth.

Fig. 5 evaluates the packet drop rate of the proposed adaptive calendar allocation scheme under different buffer times. The solid curves represent the calendar allocation schemes based on the clients' last peak data rate measured during the previous $T = 1$s period, and the dashed curves are the results of the calendar allocation schemes given by the ARIMA model forecasts. It can be observed that, for any given buffer time and the same total bandwidth, the schemes given by the ARIMA forecast model achieve lower packet drop rate compared to the scheme based on the last peak data rate. Equivalently, to achieve the same level of packet drop rates, the schemes given by ARIMA models require less bandwidth. This result indicates that the schemes given by the ARIMA forecast model outperform the ones based on the last peak data rate. Besides, for the cases with 1ms buffer or 0.1ms buffer, at a desired packet drop rate of 10^{-6}, the required bandwidth of both proposed schemes is significantly smaller than 100G as specified in the current FlexE standard.

Fig. 6 compares the required bandwidth from another perspective that evaluates the bandwidth needed to achieve zero packet drop rate in the test set. For 1ms buffer and 0.1ms buffer, the conventional scheme can successfully transmit the packets without any drops, but when the buffer time is reduced to 0.05ms, the conventional scheme starts to experience a high packet loss. This suggests that, for the conventional scheme to work smoothly, the FlexE shim must be accompanied by a sufficiently large buffer. Among all evaluated schemes, the scheme based on the ARIMA model is better in terms of the total required bandwidth. However, forecasting with ARIMA models can induce non-negligible computational overhead,



(a) (b) (c)

Fig. 4: Calendar allocation schemes: (a) calendar allocation according to current FlexE standard; (b) flexible calendar allocation based on last peak data rate with 1ms buffer; (c) flexible calendar allocation based on last peak data rate with 0.1ms buffer.

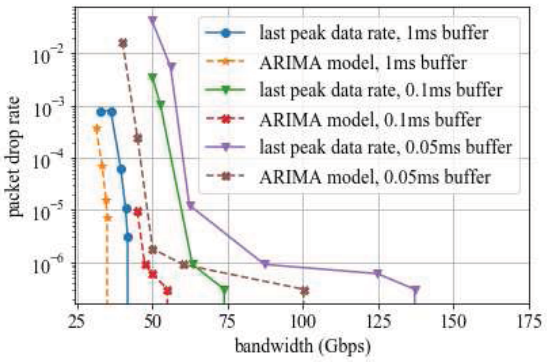


Fig. 5: Packet drop rates vs bandwidth with different calendar allocation schemes.

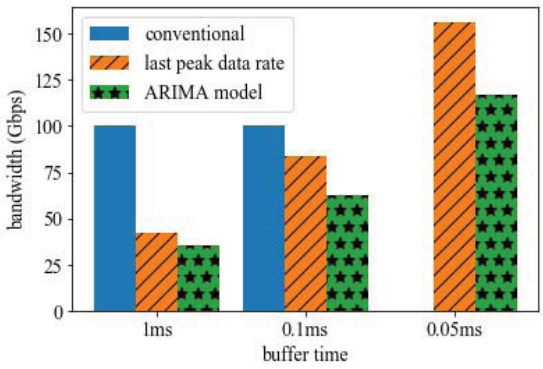


Fig. 6: Required bandwidth to achieve zero packet drop rate in test set.

because the model needs to be refit for every forecast point. With this in mind, when the buffer size is not too limited, directly using the peak data rate from the previous time period can be a good alternative. This saves computational overhead for the cost of a small increase of the required bandwidth.

V. Conclusion

This paper proposes two adaptive calendar-allocation schemes for FlexE shims, namely a scheme based on the last peak data rate and a model-driven data rate forecasting scheme. These are alternatives to the current rigid scheme in the FlexE standard. Experiments were conducted on real Ethernet packet traces to evaluate their performance in terms of the required bandwidth and packet drop rates. The results show that the proposed schemes can significantly reduce the required bandwidth by up to 60% compared to the current standard while meeting packet drop requirements. Moreover, the proposed schemes could also be easily extended to the scenarios where the packet buffer is too limited for the current FlexE calendar scheme to work. Incorporating heterogeneous network topology and delay requirements into the calendar allocation schemes as well as using emerging machine learning algorithms to enhance the data flow prediction and improve the packet drop rate are topics for future research.

Acknowledgment

This research has been supported by Huawei. In addition, the work of S. A. Hashemi is supported by a Postdoctoral Fellowship from the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] M. Filer, J. Gaudette, M. Ghobadi, R. Mahajan, T. Issenhuth, B. Klinkers, and J. Cox, "Elastic optical networking in the Microsoft cloud," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, no. 7, pp. A45–A54, Jul. 2016.

[2] B. Clouet, J. Pedro, N. Costa, M. Kuschnerov, A. Schex, J. Slovak, D. Rafique, and A. Napoli, "Networking aspects for next-generation elastic optical interfaces," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 8, no. 7, pp. A116–A125, Jul. 2016.

[3] T. Hofmeister, V. Vusirikala, and B. Koley, "How can flexibility on the line side best be exploited on the client side?" in *Optical Fiber Communications Conference and Exhibition*, Mar. 2016, pp. 1–3.

[4] Optical Internetworking Forum, "Flex Ethernet 2.0 implementation agreement," Jun. 2018. [Online]. Available: https://www.oiforum.com/wp-content/uploads/2019/01/OIF-FLEXE-02.0-1.pdf

[5] A. Eira, A. Pereira, J. Pires, and J. Pedro, "On the efficiency of flexible Ethernet client architectures in optical transport networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 1, pp. A133–A143, Jan. 2018.

[6] K. Katsalis, L. Gatzikis, and K. Samdanis, "Towards slicing for transport networks: The case of Flex-Ethernet in 5G," in *IEEE Conference on Standards for Communications and Networking*, Oct. 2018, pp. 1–7.

[7] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018.

[8] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the wide project," *USENIX 2000 FREENIX Track*, Jun. 2000.