Proactive Connection Migration in QUIC

Lizhuang Tan, Wei Su, Yanwen Liu
NGIID, School of Electronics and Information Engineering
Beijing Jiaotong University
Beijing, China
{lzhtan,wsu,19120081}@bjtu.edu.cn

Xiaochuan Gao
China Unicom
Beijing, China
gaoxc50@chinaunicom.cn

Na Li
Shandong Branch of National Computer network Emergency Response technical Team/Coordination Center (CNCERT/SD)
Ji'nan, China
lina@cert.org.cn

Wei Zhang
Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences)
Ji'nan, China
wzhang@qlu.edu.cn

ABSTRACT

Quick UDP Internet Connections (QUIC) provides a secure, reliable and low-latency communication foundation for HTTP. QUIC uses the connection ID to uniquely determine a connection from client to server. After user switches the network, the server recognizes the user request according to the connection ID and continues to provide services through the connection migration technology. This paper proposes a Proactive Connection Migration (PCM) mechanism for QUIC. PCM gives QUIC the ability to select the optimal network in a heterogeneous network environment. Firstly, PCM actively perceives the different networks available to users. Then, PCM integrates the network quality exploration of different paths into the user's multiple request actions. Finally, PCM takes response delay and jitter into account, and uses online learning to find the optimal network for current Internet service. Experimental results show that, compared with original QUIC, the average response delay of QUIC with PCM is reduced by 59.43% at most.

CCS CONCEPTS

• **Networks → Transport protocols**.

KEYWORDS

QUIC, connection migration, online learning, network selection

ACM Reference Format:
Lizhuang Tan, Wei Su, Yanwen Liu, Xiaochuan Gao, Na Li, and Wei Zhang. 2020. Proactive Connection Migration in QUIC. In *MobiQuitous'20: 17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, December 7–9, 2020, N/A, Cyberspace.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

MobiQuitous'20, December 7–9, 2020, N/A, Cyberspace

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

According to HTTPArchive[1], average web page loading time of mobile device is 87% longer than that of PC in 2019. Average time to first byte (TTFB) speed is 1.28s for PC and 2.5s for mobile device. User experience of interactive services such as online game, short video, and online shopping are closely related to network delay. Today, when mobile traffic has become mainstream, bloated and slow webpage loading has seriously affected user experience.

QUIC is a UDP-based transport protocol designed by Google[6], which aims to provide multiplexed streams over an encrypted transport. HTTP/3 has chosen to use QUIC instead of TCP as its transport layer protocol[7]. A feature of QUIC is connection migration (CM)[3]. QUIC uses connection ID to uniquely determine a transmission channel. Even if underlying network changes, as long as connection ID unchange, it can be considered as same connection. This feature ensures that when users switch between Wi-Fi, wired and mobile networks, QUIC can keep upper virtual channel unchanged and avoid various losses caused by reconnection.

If connection migration is slightly modified, QUIC can actively switches and explores available network in a HetNets environment. By comparing different network feedback information, QUIC can replaces user to explore and select the optimal network, which can improve QoE. We call this mechanism as Proactive Connection Migration (PCM), which runs between application layer and UDP layer to replaces user to explore and select the optimal network. Through PCM, third-party Internet services only need to deploy QUIC services on their servers without any other changes. And the contributions of this work are summarized below:

- **Proactive Connection Migration**: PCM, CM, and MPQUIC jointly promote QUIC to achieve the most efficient transmission in HetNets.
- **Path Selection Algorithm**: UCB-based PSA, which aims to minimize the cumulative response delay and jitter, helps PCM quickly find the optimal transmission path.
- **Performance improvement**: Compared with original QUIC, average response time of PCM is reduced by **59.43%** at most.
- **PCM prototype**: We open sourced PCM[2], which will help numerous service providers to improve user experience without changing their current complicated business codes.

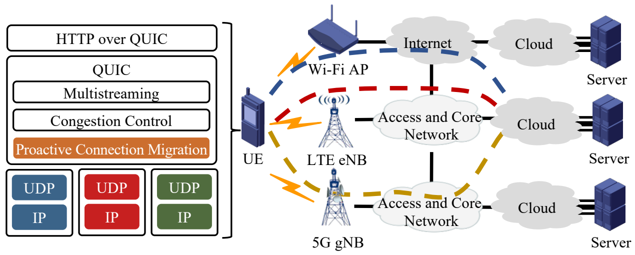


Figure 1: System model and QUIC with PCM.

2 SYSTEM DESIGN

2.1 System Model

As is shown in Fig. 1, assuming that there are three ways from UE to Server. The first is to directly access Internet through Wi-Fi of ISP A. The second and third are accessed through cellular networks of ISP B and ISP C. Before requesting service, UE is not aware of the connectivity and network quality between ISP's network and Internet Server, and cannot predict performance results[5].

2.2 Stationary/Non-stationary Response Time

We measured response time of of China Telecom's and China Mobile's LTE network in four common environments. They are indoor, walking, subway and high-speed rail environment, which can cover most mobile communication scenarios. Fig. 2 plots measurement results. We found some interesting phenomena:

(1) Response time of two networks in indoor environments is stable, but the delay is relatively high. Average delay of indoor environment is about 2.3 times that of outdoor environment.
(2) Average response delay of subway environment is better than that of outdoor environment.
(3) Failure rate and retransmission rate of high-speed rail environment are very high, which shows that ISPs are not very good at optimizing signal of this environment. The proportion of response time exceeding 100ms is much higher than that of other outdoor environment.
(4) When accessing same Internet service, the response time of different ISPs is often different. And in last three scenarios, response time usually stabilizes only in a short time.

We call the stable response time obtained in indoor environment as stationary response time. The response time of long-term drastic changes caused by movement, occlusion, etc. in an outdoor environment is called non-stationary response time.

PCM comprehensively consider stationary and non-stationary response time under a unified framework and algorithm. We take latest response time into consideration in path selection algorithm, and actively eliminate old measurements. So that PCM can better adapt to network change, and switch smoothly under stationary and non-stationary distribution scenarios.

2.3 Protocol Design

In QUIC, A connection can multiplex multiple streams. Each stream is independent. The loss of a single stream will not affect other streams. All streams in one request object are processed by the same session. All PCM data packets start with QUIC Common format, which includes a connection ID.

Each connection ID is bound to a QUIC connection, even if the client and/or server IP and port are changed, it will not change. Therefore, our idea is to keep the connection ID unchanged, modify the source address used for each user request, and send data from different NICs. Since all data packets use the same connection ID, server will think that client has migrated the connection and continue to provide services to new address. In our solution, there is no need to make any changes on server side. The current QUIC server can support PCM mechanism. We only need to change the packet sending logic of QUIC client and complete the measurement of the quality of different network.

The life cycle of a QUIC connection includes three stages: connection establishment, data transfer and connection termination.

2.3.1 Connection Establishment. QUIC Client first obtains all available IP addresses through net.InterfaceAddrs(). Client randomly selects a IP and initiates a connection. connection establishment puts version negotiation and encryption into the connection handshake. During connection handshake, the handshake must negotiate a variety of transmission parameters. PCM selects path according to the strategy described in Sec. 3. In addition, PCM allows the use of keys exchanged in the previous connection establishment without having to re-do HTTPS handshake.

2.3.2 Data Transfer. PCM selects the source IP address through some strategies, and records the stream complete time. Other processing logic is consistent with native QUIC.

Stream State Management maintains stream information, including stream establishment time, end time, and total transmitted bytes. It is expressed as $< StreamID, EstTime, EndTime, TotalBytes >$.

Because client initializes stream, *StreamID* is a variable unsigned integer odd number. *EstTime* records the time of stream creation, which is initialized by *Create Stream*. *EndTime* means the end time of the stream, which is the moment when a frame with FIN=1 is received. *TotalBytes* represents the total number of bytes transmitted on one stream, and its function is to normalize and compare streams of different lengths. *TotalBytes* is calculated based on the frame with FIN=1 of each stream. Its calculation equation is:

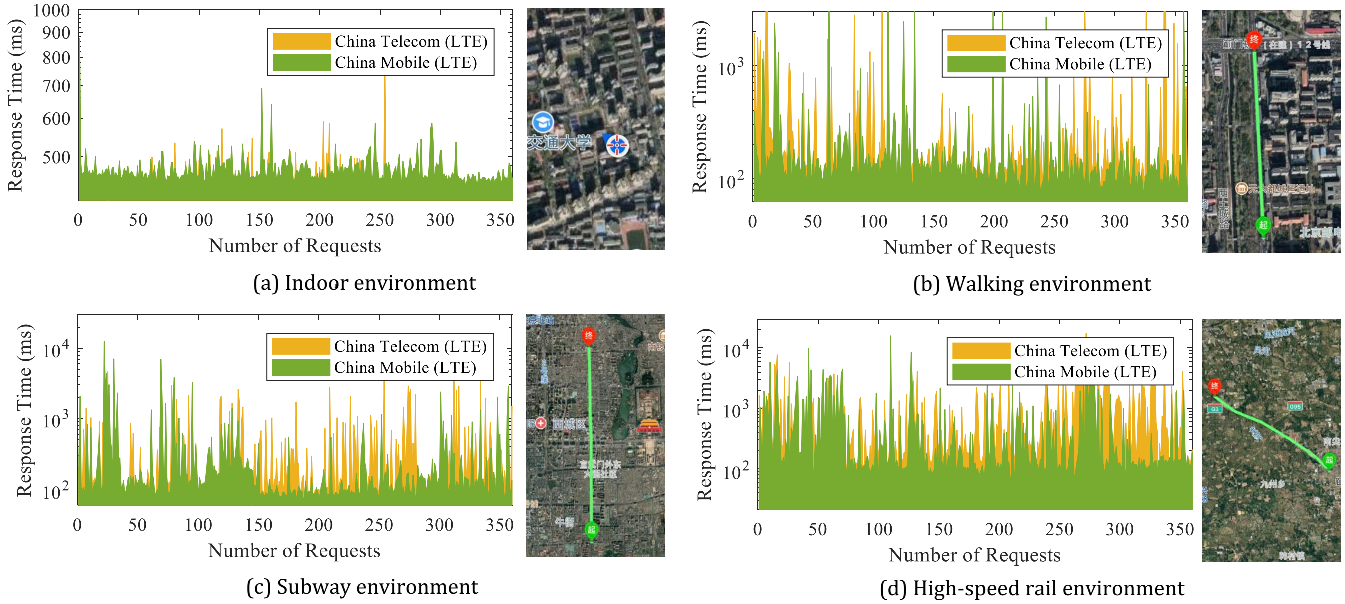
$$TotalBytes(StreamID) = Offset + DataLength. \tag{1}$$

we can use the response time per byte to evaluate efficiency. If all streams with fixed bytes, we can directly use response time.

$$ResponseTime = \frac{(EndTime - EstTime)}{TotalBytes}. \tag{2}$$

For streams with different numbers of bytes, by Eq.2, we can evaluate the completion time of different streams. In addition to the completion time, PCM module also needs to record the mapping relationship between stream and UDP, which can be expressed as $< StreamID, UDPSourceIP >$.

2.3.3 Connection Termination. The stream and connection closure of PCM is similar to QUIC, including three situations: Normal Termination, Abrupt Termination and Connection Termination.



(a) Indoor environment

(b) Walking environment

(c) Subway environment

(d) High-speed rail environment

Figure 2: Response time measurement results in different scenarios.

3 PATH SELECTION MODEL

PCM needs to find the best communication path for users as soon as possible through less exploration. This model is a path selection with discrete states, which is a typical Multi-Armed Bandit (MAB) problem[4]. The traditional MAB pursues the maximization of accumulated profit, which is a model that doesn't care about the process, only the result. However, in Internet services, users are not only seeking extreme low response time, but also long-term fluency. So, PCM care about the minimum response time and jitter corresponding to different paths.

We define the complete process of a request and response as a round. The path selection under each round is an action, the response time under each round is reward. For more detailed notations, please refer to Tab. 1.

3.1 Probability Distribution of Response Time

The response time is the time interval between the user sending a request and loading Largest Contentful Paint (LCP). We assume that the response time obeys the heavy-tailed distribution, whose kurtosis > 3. This is due to the presence of noise caused by queuing, which can lead to extremely bad situations.

In order to balance stationary and non-stationary response time, we propose a sliding window S, which means PCM does not use all response times, but uses the latest S response time values[8].

Assuming that X_k is a Lognormal random variable, $\ln(X_k) \sim N(\mu_k, \sigma_k^2)$ is a normal distributed random variable. Given a $x_k > 0$, the probability density function of X_k is:

$$f(x_k) = \frac{1}{\sqrt{2\pi} x_k \sigma_k} e^{-\frac{(\ln x_k - \mu)^2}{2\sigma_k^2}}. \quad (3)$$

Table 1: Notations in the model.

Notations	Description
K	Number of access paths supported by UE
T	Rounds
t	The t-th round
A_t	The actions selected by PCM when requested at t
R_t	The acquired response under action A_t
r_k	The average response on k-th path
f_k	The fluctuation(variance) on k-th path
N_k	The cumulative number of times the k-th path has been selected
X_k	response time of the k-th path
$\overline{X_k}$	Average response time of the k-th path
ρ	The regret after T rounds
μ^*	The optimal response
I_k	The indicator of the k-th path
S	The size of Sliding window

The maximum likelihood estimations of distribution parameters are:

$$\hat{\mu_k} = \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k} \quad (4)$$

$$\hat{\sigma_k^2} = \frac{[\ln(X_{k,i}) - \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k}]^2}{N_k - 1} \quad (5)$$

The expectation and variance of a Lognormal distributed random variable X_k are:

$$
\begin{aligned}
E(X_k) &= e^{\hat{\mu}_k + \frac{\hat{\sigma}_k^2}{2}} \\
&= e^{\frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k} + \frac{\frac{[\ln(X_{k,i}) - \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k}]^2}{N_k - 1}}{2}}
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
var(X_k) &= e^{2\hat{\mu}_k + \hat{\sigma}_k^2}(e^{\hat{\sigma}_k^2} - 1) \\
&= e^{2\frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k} + \frac{[\ln(X_{k,i}) - \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k}]^2}{N_k - 1}} \\
&\cdot (e^{\frac{[\ln(X_{k,i}) - \frac{\sum_{i=1}^{N_k} \ln(X_{k,i})}{N_k}]^2}{N_k - 1}} - 1)
\end{aligned} \tag{7}
$$

3.2 Response Function

The response on path k consists of two parts: the average value and variance of historical response delay, The former represents the delay performance and the latter represents the jitter performance. a and b are the weight parameters of these two parts, and $a + b = 1$. Different businesses can choose different a and b. For example, web search requires absolute delay, $a = 0.8, b = 0.2$ can be used. Therefore, the expression of r_k is

$$r_k = aE[X_k] + b \cdot var(X_k). \tag{8}$$

We give two path selection strategies, namely optimistic path selection (OPS) and pessimistic path selection (PPS).

In OPS, PCM always chooses the path with the smallest lower bound confidence. The path indicator I_k^{OPS} is

$$
\begin{aligned}
I_k^{OPS} &= r_k - C\sqrt{\frac{2\ln T}{N_k}} \\
&= aE[X_k] + b \cdot var(X_k) - C\sqrt{\frac{2\ln T}{N_k}}.
\end{aligned} \tag{9}
$$

In PPS, PCM chooses the path with the smallest upper confidence bound. The path indicator I_k^{PPS} is

$$
\begin{aligned}
I_k^{PPS} &= r_k + C\sqrt{\frac{2\ln T}{N_k}} \\
&= aE[X_k] + b \cdot var(X_k) + C\sqrt{\frac{2\ln T}{N_k}}.
\end{aligned} \tag{10}
$$

And T in Eq.9 and 10 should be replaced with S, when $T > S$.

$C\sqrt{\frac{2\ln T}{N_k}}$ is path bonus, which means: If the path is selected a few times and the confidence interval is wide, it will tend to be selected. C determines the scope of exploration. In the UCB algorithm, the selection of the C value is often an empirical value. The most suitable C value should not interfere too much with the expectation and variance of each path. So, we propose an adaptive C value selection strategy as shown in Eq.11.

$$C = \frac{\sum_{k=1}^{K} (aE[X_k] + b \cdot var(X_k))}{K} \tag{11}$$

Algorithm 1 Path Selection Algorithm (PSA)

Require: $a \in [0, 1]$, $b = 1 - a$, K, T, S
for $t = 1, 2, \ldots, K$ **do**
  $Time_1 = Time.Now()$
  $A_t = t$
  $ResponTime = Time.Now() - Time_1$
  for $k = 1, 2, \ldots, K$ **do**
    Update N_k
    Update $E[X_k]$ and $var(X_k)$ according to Eq.6 and Eq.7
  end for
end for
Calculate C according to Eq.11
for $t = K+1, K+2, K+3, \ldots, T$ **do**
  $I_{min} = +\infty$
  for $k = 1, 2, \ldots, K$ **do**
    Calculate I_k according to Eq.9 or Eq.10
    if $I_k \leq I_{min}$ **then**
      $PathNumber = k$
    end if
  end for
  $Time_1 = Time.Now()$
  $A_t = PathNumber$
  $ResponTime = Time.Now() - Time_1$
  for $k = 1, 2, \ldots, K$ **do**
    Update X_k and N_k
    Update $E[X_k]$ and $var(X_k)$ according to Eq.6 and Eq.7
  end for
end for

3.3 Accumulated Regret

We perform a weighted summation of the accumulated regret of delay and jitter to evaluate the effectiveness of the algorithm. The regret ρ after T rounds is

$$
\begin{aligned}
\rho &= \sum_{t=1}^{T} \tilde{r}_t - T\mu^* \\
&= \underbrace{a[\sum_{t=1}^{T} R_t - T \cdot \min(R_t)]}_{\text{The accumulated regret of delay}} + \underbrace{b\{var(R_t) - \min[var(X_k)]\}}_{\text{The accumulated regret of jitter}}
\end{aligned} \tag{12}
$$

3.4 Path Selection Algorithm

Algorithm 1 describes the pseudo code of path selection algorithm (PSA). The PSA consists of two stages. In first stage, PSA uses each available path in turn to generate the first round of I_k. In second stage, PSA selects the path with the smallest I_k for transmission.

4 EXPERIMENT RESULTS

4.1 Performance of PCM

We deployed the experimental topology shown in Fig. 3. Client accesses Internet through four ways. We turn on personal hotspots on mobile phones equipped with SIM cards of different ISPs. Clients

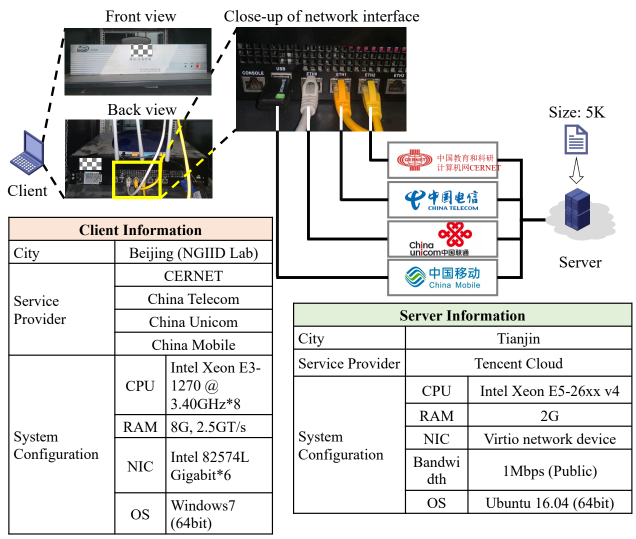


Figure 3: Experimental environment and topology.

respectively access personal hotspots of different mobile phones through wireless routers, thus realizing access to different ISP networks. As evidenced by Tab. 2, we measured network performance through PING as benchmark for PCM. We evaluated the impact of four path selection strategies. They are polling, ε-Greedy, optimistic PSA and pessimistic PSA.

Polling strategy represents average performance of these four paths. ε-Greedy strategy always tend to choose the path with the best historical performance, and randomly explore other paths with a certain probability ε. This strategy is a very stable solution. In this experiment, $\varepsilon = 0.05$. Optimistic PSA and pessimistic PSA choose path according to Eq.9 or Eq.10 respectively. $S = 100$ and $a = 0.8$. We evaluated the above strategies from response time per round, cumulative regret, and optimal path selection probability.

4.1.1 *Response Time Per Round.*

Fig. 4 demonstrates the response time per round. The response time of the polling strategy is close to periodic. ε-Greedy strategy has alleviated the drastic fluctuation of polling strategy. The two PSA strategies further reduce the fluctuation of network delay by selecting the optimal path for transmission. Particularly, the pessimistic PSA has been significantly lower than the other three strategies. The reason is that the performance of path-1 and path-2 are close, and the probability of optimistic PSA strategy switching between the two paths is greater than that of pessimistic PSA strategy.

Compared with polling, the mean and variance of 1000-round response time of the optimistic PSA strategy are reduced by 41.01% and -9.59%. Compared with the ε-Greedy strategy, it is reduced by 11.99% and 8.71%. Pessimistic PSA outperforms the other three. Compared with the polling, its mean and variance are reduced by 59.43% and 27.52% respectively. Compared with the ε-Greedy, the two results are 39.47% and 39.62%.

4.1.2 *Accumulated Regret.*

As shown in Fig. 5, the accumulated regret of polling strategy is obviously inferior to the other three strategies. The two PSA strategies are significantly better than the other two strategies, which shows that the PSA strategy is closer to the optimal exploration strategy.

4.1.3 *Probability of Selecting Best Path.*

Fig. 6 illustrates the probability of different strategies to select the optimal path. In most rounds, the latter three strategies choose path-2 as the transmission path, which shows that path-2 is the optimal path under the current experimental condition. The polling strategy is undoubtedly only a 25% hit probability. The ε-Greedy strategy has a hit probability of 91.3% in the end, because the greedy coefficient ε will cause this strategy to avoid the optimal path with a certain probability

Table 2: Statistics of path performance.

Value / Statistic / Path	Ave RTT	Std Dev	TTL	Loss
Path-1 (CERNET)	7.120ms	**0.446ms**	51	0%
Path-2 (China Telecom)	**6.707ms**	0.636ms	53	0%
Path-3 (China Unicom)	41.004ms	6.458ms	**50**	0%
Path-4 (China Mobile)	37.568ms	16.756ms	**50**	0%

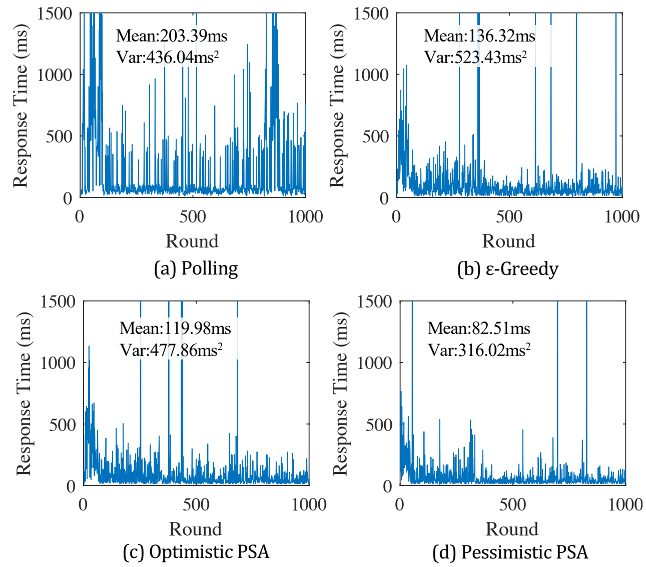


Figure 4: The response time per round of the four strategies.

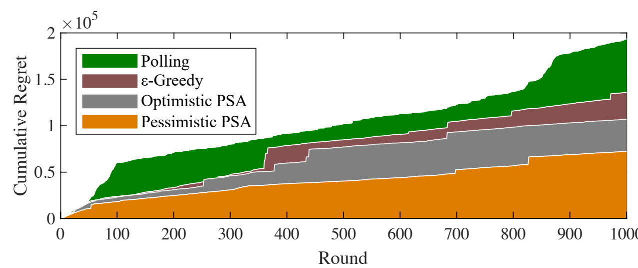
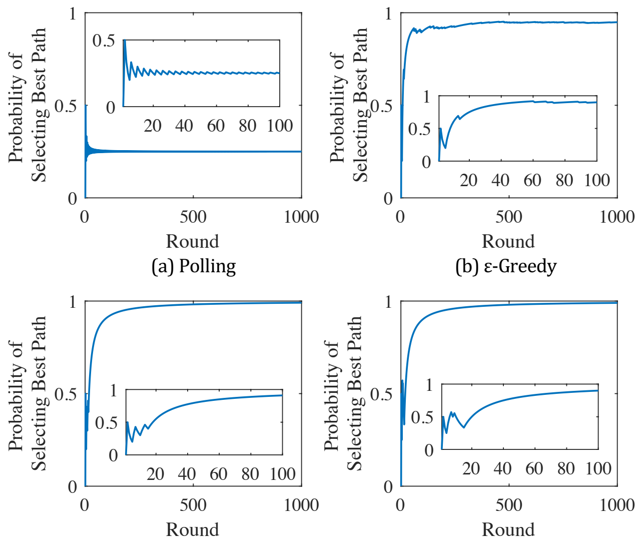


Figure 5: Cumulative regret of four strategies.



(a) Polling (b) ε-Greedy (c) Optimistic PSA (d) Pessimistic PSA

Figure 6: Probability of selecting best path.

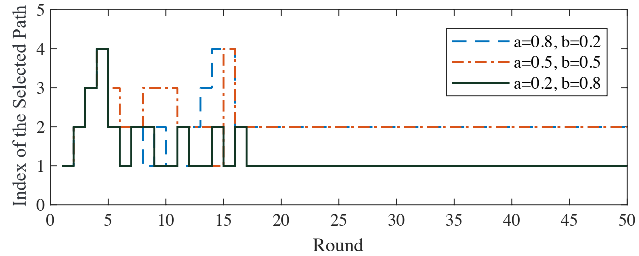


Figure 7: Path selection for the first 50 rounds.

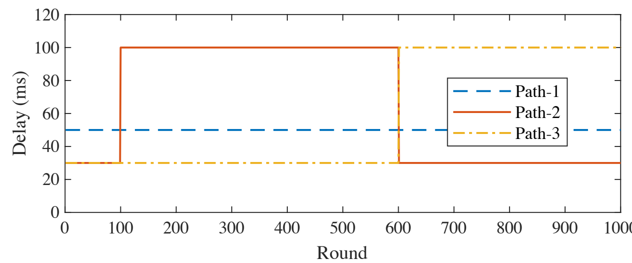


Figure 8: Delay variation of three paths.

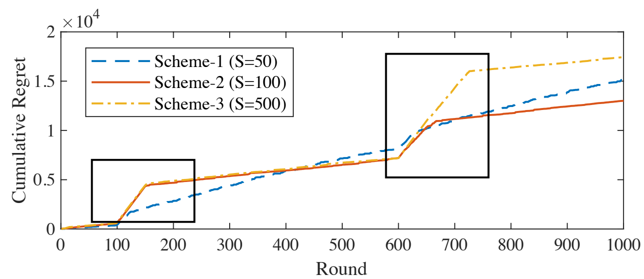


Figure 9: Cumulative regret of three shcemes.

ε. The hit probability of the two PSA strategies approached 100% as quickly as possible, which shows that PSA has an efficient and accurate path exploration effect.

4.2 The Influence of Parameters on PCM

4.2.1 Weight Parameter a and b. Fig. 7 compares the influence of different combinations of a and b under pessimistic PSA model, they are $a = 0.8$, $a = 0.5$ and $a = 0.2$. After 15 rounds, the former two finally selected path-2 as the best path, and the third selected path-1. This result shows that PCM can meet requirements of different applications. We can choose the emphasis on path response time expectation and variance by setting a and b.

4.2.2 Sliding Window. We opened three virtual NICs on VMware Linux server to simulate three different access paths. Network delay, packet loss and bandwidth are set through VMware Fusion.

Fig. 8 shows the network delay changes. Fig. 9 presents the cumulative regret changes of three schemes. Unexpectedly, when S=100, the accumulated regret is the smallest. The reason for scheme-3 is that PCM is slow to respond to response time change. The reason for scheme-1 is that PCM not only considers the path reward, but also considers path bonus. Therefore, scheme-1 has to frequently switch to a path with poor performance. So the smaller S is, the more PCM pays attention to the latest response time. The larger S is, the longer the influence of historical response time is.

5 CONCLUSION

This paper proposes Proactive Connection Migration (PCM) mechanism for QUIC, which uses the completion time at the transport layer to realize the network selection of HetNets.

6 ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China under Grant No.2018YFB1800305 and No.2019YFF0303101, the Natural Science Foundation of Shandong Provincial under Grant No. ZR2019LZH013.

REFERENCES

[1] 2019. HTTPArchive. https://httparchive.org/
[2] 2020. PCM. https://github.com/lzhtan/PCM
[3] Yong Cui, Tianxiang Li, Cong Liu, Xingwei Wang, and Mirja Kühlewind. 2017. Innovating transport with QUIC: Design approaches and research challenges. *IEEE Internet Computing* 21, 2 (2017), 72–76. https://doi.org/10.1109/MIC.2017.44
[4] Zhiyong Du, Bin Jiang, Kun Xu, Shengyun Wei, et al. 2019. Second-order multi-armed bandit learning for online optimization in communication and networks. In *TURC'19*. ACM, Chengdu, China, 1–6. https://doi.org/10.1145/3321408.3323078
[5] Yan Huang, Yongce Chen, Y Thomas Hou, and Wenjing Lou. 2020. Achieving Fair LTE/Wi-Fi Coexistence with Real-Time Scheduling. *IEEE Transactions on Cognitive Communications and Networking* 6, 1 (2020), 366–380.
[6] Adam Langley, Janardhan Iyengar, Jeff Bailey, Jeremy Dorfman, and Ian Swett. 2017. The QUIC Transport Protocol: Design and Internet-Scale Deployment. In *SIGCOMM'17*. ACM, Los Angeles, USA, 183–196. https://doi.org/10.1145/3098822.3098842
[7] Jan Rüth, Konrad Wolsing, Klaus Wehrle, and Oliver Hohlfeld. 2019. Perceiving QUIC: Do Users Notice or Even Care?. In *CoNEXT'19*. ACM, Orlando, Florida, 144–150. https://doi.org/10.1145/3359989.3365416
[8] Ruowu Wu, Xiang Chen, Hui Han, Haojun Zhao, and Yun Lin. 2018. Abnormal Information Identification and Elimination in Cognitive Networks. *International Journal of Performability Engineering* 14, 10 (2018), 2271–2279. https://doi.org/10.23940/ijpe.18.10.p3.22712279