QoE-Aware Traffic Aggregation Using Preference Logic for Edge Intelligence

Pingping Tang, *Member, IEEE*, Yuning Dong, *Member, IEEE*, Yin Chen, *Member, IEEE*, Shiwen Mao, *Fellow, IEEE*, and Saman Halgamuge, *Fellow, IEEE*

***Abstract*—Traffic flows with different requirements of quality of service (QoS requirements) are aggregated into different QoS classes to provide differentiated services (Diffserv) and better quality of experience (QoE) for users. The existing aggregation approaches/QoS mapping methods are based on quantitative QoS requirements and static QoS classes. However, they are typically qualitative and time-varying at the edge of the beyond fifth generation (B5G) networks. Therefore, the artificial intelligence technology of preference logic is applied in this paper to achieve an intelligent method for edge computing, called the preference logic based aggregation model (PLM), which effectively groups flows with qualitative requirements into dynamic classes. First, PLM uses preferences to describe QoS requirements of flows, and thus can deal with both quantitative and qualitative cases. Next, the potential conflicts in these preferences are eliminated. According to the preferences, traffic flows are finally mapped into dynamic QoS classes by logic reasoning. The experimental results show that PLM presents better performance in terms of QoE satisfaction compared with the existing aggregation methods. Utilizing preference logic to group flows, PLM implements a novel way of edge intelligence to deal with dynamic classes and improves the Diffserv for massive B5G traffic with quantitative and qualitative requirements.**

***Index Terms*—Aggregation, differentiated services (Diffserv), edge intelligence, network traffic, preference logic, quality of experience (QoE), quality of service (QoS).**

Manuscript received July 16, 2020; revised November 5, 2020 and January 11, 2021; accepted April 1, 2021. Date of publication April 26, 2021; date of current version September 10, 2021. This work was supported in part by the NSFC under Grant 61271233 and Grant 62071005, in part by the AnHui Universities Natural Science Researches (AHUNSR) under Grant KJ2019A0491 and Grant 2008085MF181, and in part by the NSF under Grant ECCS-1923717. This article was presented in part at the IEEE ICCCS, Singapore, February 2019. The associate editor coordinating the review of this article and approving it for publication was L.-C. Wang. *(Corresponding author: Yuning Dong.)*

Pingping Tang is with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, and also with the College of Physics and Electronic Information, Anhui Normal University, Wuhu 241000, China (e-mail: tpping@ahnu.edu.cn).

Yuning Dong is with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: dongyn@njupt.edu.cn).

Yin Chen is with the Graduate School of Media and Governance, Keio University, Kanagawa 252-0882, Japan (e-mail: yinchen@sfc.keio.ac.jp).

Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org).

Saman Halgamuge is with the School of Electrical, Mechanical and Infrastructure Engineering, University of Melbourne, Melbourne, VIC 3010, Australia (e-mail: saman@unimelb.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2021.3071745.

Digital Object Identifier 10.1109/TWC.2021.3071745

I. Introduction

WITH the rapid development of the beyond fifth generation (B5G) network, Internet of Everything (IoE) has been widely accepted as one vision of future Internet [2], where massive number of devices (e.g., vehicles, mobile phone, and wearable devices) are connected, which result in complex heterogeneous networks (HNs) at the edge as shown in Fig. 1. Edge intelligence (EI), powered by artificial intelligence (AI) techniques, is considered to be crucial and represents a key enabling factor for future IoE. According to the 6G white paper [9], *EI is to use advanced communication technologies and AI to support ubiquitous data collection, aggregation, fusion, processing, distribution and services at the edge*. EI includes many issues, such as traffic management [3], data aggregation [4], resource allocation [5], differentiated service (Diffserv) [6], scheduling strategy [7], security issues [8], etc. Traffic aggregation, the prerequisite and foundation for further Diffserv, scheduling and management, is an important research topic in EI, which has aroused great concern in the field of communications [10]. As shown in Fig. 1, traffic flows are grouped into different aggregates (i.e., *classes*) according to their different requirements for quality of service (QoS). However, in heterogeneous networks of IoE, it is challenging for EI to implement data aggregation, which is due to the following issues.

- Traffic from heterogeneous devices may have quite different requirements [11], e.g., some flows generated by smartphone apps in B5G systems may have requirement on extremely low delay (ELD) while some others generated by sensors may have requirement only on break (see F_3 in Table I). It is difficult for an aggregation model to cover all aggregation criteria, e.g., break [12], loss (i.e., packet loss rate) [10], user's awareness [13], context [4], etc. It is impossible for a flow to offer all values. That is, the aggregation criteria are not fixed for those heterogeneous traffic.
- With the rapid development of B5G network, some qualitative human-centered [14] and socially-aware metrics [15] (e.g., user level: silver) are already involved in the calculation, so we need effective means to deal with qualitative information.
- Networks in IoE have different specifications for classes [16]. For example, 3GPP (3rd Generation Partnership Project) defines 4 classes: conversation,

1536-1276 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

TABLE I
HYBRID QOS REQUIREMENTS OF HETEROGENEOUS TRAFFIC

Flow	Bandwidth (kbps)	Delay (ms)	ELD (ms)	Loss	Jitter (ms)	breaks	Price ¥/Gb	Service Provider	User Level	Security Level
F_1	134	46	3	0.0015	57	—	—	—	—	—
F_2	—	32	3	—	—	—	—	—	—	—
F_3	—	—	—	—	—	5	—	—	—	—
F_4	—	—	—	—	—	—	15	CTCC	Gold	—
F_5	102	110	5	0.01	90	1	10	CUCC	Silver	High
F_6	120	70	4	0.001	81	1	Don't care	CMCC	Copper	Don't care

streaming, interaction and background. TIPHON (Telecommunication and Internet Protocol Harmonization Over Networks) specifies 3 classes: wideband, narrowband and BE. When a flow travels from a 3GPP domain to a TIPHON domain network, the target classes are totally different.

The existing aggregation models are not directly applicable to edge intelligence. In these models, the *aggregation criteria* are fixed [10], the *requirements* are quantitative [17] and the *classes* are static [16]. We need to explore a generic aggregation model to deal with such variations. Therefore, based on the AI technology of logic reasoning, we proposed a novel preference logic based aggregation model (PLM) to map massive heterogeneous traffic with different quantitative and qualitative requirements (i.e., hybrid requirements) into time-varying classes. The major contributions are summarized below.

- Based on the preference cognition, a novel description of QoS requirements is proposed to deal with the hybrid requirements (see Section III-B), which breaks through the limitations of the existing aggregation models that can handle only quantitative requirements.
- Based on the logic reasoning, an innovative edge computing PLM is presented to aggregate traffic flows into dynamic QoS classes. With the powerful reasoning capability, PLM shows a wonderful performance to deal with time-varying aggregation criteria and classes.
- For the scenarios of this paper, we make improvements on many aspects of the preference logic, e.g., preference evaluation and preference conflict detection, which would promote the application of preference logic in other research fields.

The remainder of the paper is organized as follows. In Section II, we present some related work and analyze their limitations. PLM is defined and theoretically verified in Section III. The datasets and performance metrics are discussed in Section IV. Section V evaluates the performance. Section VI concludes this paper.

II. RELATED WORK

There are many aggregation methods proposed to provide Diffserv for network traffic. Wu *et al.* [10] exploited K-Means to aggregate flows across HNs, where flows with similar QoS requirements (e.g., transmission rate) are clustered together into one macro flow. Considering that different flows may have different preferences on QoS parameters, Hijazi *et al.* [17] presented the class weight based K-nearest neighbor method (CWK-NN), where a weight-learning algorithm is explored to assign weights to different QoS parameters according to their importance in aggregation. For example, in the study of [16], the QoS requirement of flow F_1 is $\{$134kbps, 46ms, 0.0015, 57ms$\}$ as shown in Table I, and their weights are $\{0.5, 0.25, 0.15, 0.1\}$. With the development of traffic aggregation, researchers began to realize that user's awareness is an important factor that should be considered in aggregation models [14]. For example, it is somewhat useless to distinguish flows which have the delay requirements of 50ms and 60ms, if users are not sensitive to the difference between them. That is, users usually have their own sensitivities and preferences [18]. Network resources would be wasted if more resources are allocated to the users who cannot perceive much gain due to their cognitive limitations. Therefore, Chen *et al.* [12] proposed an efficient aggregation model based on reinforcement learning (RL) to minimize the breaks in presence (BIP) and thus improve the quality of experience (QoE) of different users. Targeting user's QoE and energy consumption, He *et al.* [19] developed a dynamic Q-learning based aggregation model, where flows with different QoE values are mapped into different **networks**, e.g., UMTS (Universal Mobile Telecommunications System) and Wi-Fi (wireless fidelity). With the in-depth research, human-centered [14] and socially-aware [15] networking techniques are in demand, where quantitative criteria would be widely involved in. For example, in the specification of RFC2594, the network flows are divided into four levels: platinum, gold, silver and copper [22]. Kasgari *et al.* [14] took into account qualitative criteria (e.g., user level and **gender**) in their study on delay requirement model. Based on the delay requirements, users are assigned with different network resources.

The differences between the existing works and the proposed PLM are summarized in Table II. Three are three issues that need to be pointed out here: i) Work [14] is semi-hybrid since qualitative criteria such as gender are just used to achieve the delay requirements; the final mapping is still based on the quantitative delay requirements and not the qualitative criteria. ii) The source of mapping is mainly the user or traffic. Note that the user would generate different types of traffic (e.g., video and email) which may have different QoS requirements. Therefore, we think traffic

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

TABLE II
COMPARISONS BETWEEN DIFFERENT WORKS

Methods	Mapping (Source → Target)	Criteria	Type	Value	Purpose
K-Means [10]	Traffic → QoS Class (High, Low, etc.)	Transmission rate, Arrival Time, etc.	Fixed	Quantitative	Reduce delay
K-NN [17]	Traffic → Business (Video, Audio, etc.)	Server Port, Flow Size, etc.	Fixed	Quantitative	Maximize precision
RL [12]	User → Base Station	User's Location, User's Orientation	Fixed	Quantitative	Minimize breaks
PCA [20]	Traffic → Node	Temperature, Voltage	Fixed	Quantitative	Reduce power
Q-learning [19]	Business → Network (UMTS, WiFi, etc.)	Throughput, Delay, Loss and Jitter	Fixed	Quantitative	Optimize energy consumption
Matching [21]	User → Network	Delay, Loss, etc.	Fixed	Quantitative	Maximize throughput
GMM [14]	User → Queue	Latency Requirements	Fixed	Semi-Hybrid	Improve QoE
PLM	Traffic → Queue	(Unlimited)	Variable	Hybrid	Improve QoE

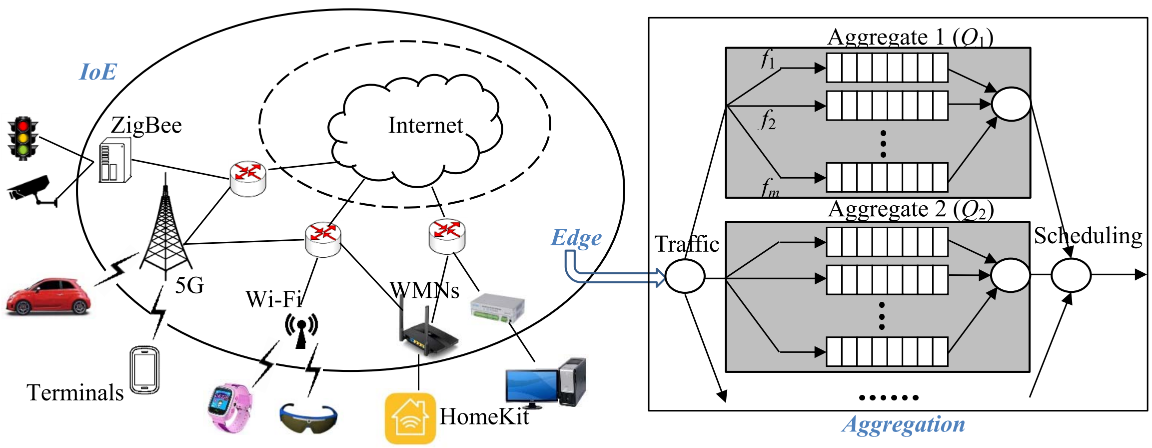

Fig. 1. An illustration of aggregation in edge intelligence.

makes more sense than users for Diffserv. iii) The mapping can be set to User→Queue, Traffic→Queue, Traffic→Node, User→ BS/Networks, etc. Note that the main focus in traffic transmission under Diffserv framework is resource allocation at the router queues. Thus, we discuss the mapping Traffic→ Queue in this paper. From Table II, the existing methods show the following limitations:

i) The aggregation criteria are fixed. For example, when a flow gives the requirements: {134kkps, 46ms, 57ms}, the aggregation algorithm [23] would calculate the distance between the flow and the class by $||(f_i - Q_i)||$. In this algorithm, the aggregation criteria are fixed to {bandwidth, delay, jitter}. Each flow should provide the precise requirements for bandwidth, delay, and jitter. When any of these values is missing, the aggregation algorithm cannot work properly. However, the requirements of traffic in HNs are quite different, so we need to explore a generic aggregation model which can deal with variable criteria for heterogeneous traffic.

ii) The QoS requirements are quantitative. For example, Alkharasani *et al.* [24] proposed a scheduling algorithm: $Wq = 1 - e^{-1/(Bw+De+Pl)}$ where Wq denotes the coefficient of resource allocation; Bw, De, and Pl are bandwidth, delay and loss, respectively. Their weights are $[1, 1, 1]$, which means the three parameters have absolutely equal importance. However, in reality, flows usually have different preferences. Some of them prefer low delay (e.g., telemedicine) and others prefer high bandwidth (e.g., video on demand) [25]. Such "preferences" are inappropriate to be quantified with accurate numbers. Besides, as stated in Section I, some qualitative metrics are involved in aggregation, so we need effective means to deal with qualitative information.

iii) The QoS classes are static. For example, Wang and Hsieh [26] implemented an elastic mapping method in the long-term evolution (LTE) network, where the QoS classes are fixed to conversational voice, live streaming, real time gaming, etc. If a class is changed, which means the cluster center is varied, the aggregation scheme needs to be completely re-trained. However, IoE systems contain a lot of HNs such as Wi-Fi and ZigBee, which have different specifications for their classes [9]. Aggregation models in IoE should be able to deal with variable classes.

In summary, based on fixed aggregation criteria, quantitative requirements, and static target classes, the current aggregation methods cannot be applied to edge intelligence, where the aggregation criteria are variable, the requirements of flows are typically complex, and target classes are dynamic. Therefore,

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

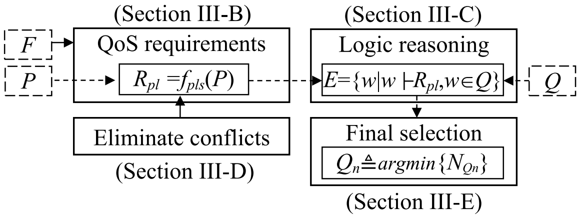

Fig. 2. Block diagram for the organization of Section III.

the preference logic is explored in this paper to achieve an innovative aggregation model PLM. Preference logic was initially proposed by Von Wright in 1963 [27]. It is used to solve qualitative problems in economics under highly variable environments [28]. Therefore, based on the AI technology of preference logic, PLM works well when facing with hybrid requirements under variable aggregation criteria and dynamic classes. This novel intelligent edge computing PLM provides a general aggregation model for better Diffserv/QoE support in B5G network.

III. The Proposed Aggregation Model PLM

A. Problem Statement

The proposed aggregation model PLM is described as

$$M_{plm} = f_{pl}(F, Q, R_{pl}) \tag{1a}$$

$$\text{s.t.} \quad R_{pl} = f_{pls}(P) \tag{1b}$$

where $F = \bigcup f_m$ denotes a group of flows, and f_m represents an individual flow $(m = 1, 2, \ldots, \aleph_f)$; $Q = \bigcup Q_n (n \in z_d^+)$ represents the set of dynamic QoS classes, and z_d^+ is the set of natural number greater than 0; $P = \bigcup p_j \ (j \in z_d^+)$ is the set of variable QoS parameters (e.g., delay and loss); R_{pl} is hybrid QoS requirements described in preference. PLM groups flows F with hybrid requirements R_{pl} into dynamic classes Q under variable parameters P. Here, Q_n is defined as

$$Q_n := \{(p_j, d(p_j)), j \in Z_d^+\} \tag{2}$$

where $d(p_j)$ is the demand value of p_j for class Q_n. As shown in Table IV, the demand values for class Q_n are designated according to the existing international standards such as IETF (Internet Engineering Task Force).

Based on (1a) and (1b), the proposed aggregation scheme consists of four main components (see Fig. 2): First, based on P, the hybrid QoS requirements R_{pl} of flows are modeled in Section III-B, and preference conflicts in R_{pl} are eliminated in Section III-D; Next, the optimal candidates that best satisfy R_{pl} are derived from Q in Section III-C; Finally, Section III-E makes the final selection from the optimal candidates. The important symbols are listed in Table III for the ease of reading. The symbols with capital italic letters represent sets.

B. Modeling the QoS Requirements in Qualitative Manner

QoS requirements are usually described in quantitative weight-based models [16], e.g., $f_1 = \{134$ kbps, 46ms, 0.0015, 57ms$\}$ (the QoS parameters are bandwidth, delay, loss and jitter, respectively), and their weights are [0.5, 0.25, 0.15, 0.1]. Here, the weights are computed according to the relationship between the QoS parameters and QoE. QoE reflects the overall experience of QoS in network environment [29]. Thus, the influence of each QoS parameter on QoE can indicate their importance [30]. More concretely, the weights are obtained as follows: i) First, collect the values of QoS parameters and QoE, such as $\{pd(1) = 58ms, pl(1) = 0.001, \ldots, qoe(1) = 3\}$, $\{pd(2) = 42ms, pl(2) = 0.001, \ldots, qoe(2) = 4\}$, where $pd(i), pl(i), qoe(i) \ (i = 1, 2, \ldots, I)$ represent the ith observation of delay, loss and QoE, respectively. ii) Next, model the importance of the QoS parameters by quantitative fitting methods [16], [31], and thus obtain the precise weights. However, in reality, it's hard to determine how much bandwidth is more important than delay, which is difficult to describe by an accurate number.

In this paper, we use qualitative R_{pl} to describe the QoS requirements, e.g., $R_{pl} = \{ban_{134}{}^{\nabla} >^{\nabla} del_{46}{}^{\Delta} >^{\nabla} Sec_{High}{}^{\Delta} >^{\Delta} Use_{Gold}\}$. Here, R_{pl} is also derived based on the above QoS and QoE. Note that the scales of these data are different, which need to be normalized:

$$\bar{p}_k(i) = \frac{p_k(i)}{\sum_i p_k(i)} \quad (i = 1, 2, \ldots, I) \tag{3}$$

where $p_k \in P$ represents the QoS parameter such as delay and loss. The values of $qoe(i)$ are also transformed to the non-dimensional form by (3). Based on GRA (Grey Relational Analysis) [32], the correlation between p_k and QoE is defined as

$$r_{p_k} = \left[\frac{1}{I}\sum_{i=1}^{I} \Upsilon_{p_k}(i)\right] \tag{4a}$$

$$\text{s.t. } \Upsilon_{p_k}(i) = \frac{\underset{i,k}{min}\{\lambda x_k(i)\} + \frac{1}{2}\underset{i,k}{max}\{\lambda x_k(i)\}}{\lambda x_k(i) + \frac{1}{2}\underset{i,k}{max}\{\lambda x_k(i)\}}, \tag{4b}$$

$$\lambda x_k(i) = |qoe(i) - \bar{p}_k(i)|. \tag{4c}$$

GRA is used in various research fields (e.g., medicine [33] and chemistry [34]) to measure the relationship between different factors. It has mainly four steps: i) All the values are to be normalized as in Eq. (3); ii) Obtain the absolute difference between the standard sequence and correlation sequences as in Eq. (4c); iii) Calculate the correlation coefficient between each column of the standard sequence and correlation sequences as in Eq. (4b); iv) Calculate the total correlation degree between the standard sequence and correlation sequences as in Eq. (4a).

The range of r_{p_k} is $(0, 1)$. The larger the value of r_{p_k}, the greater the influence of parameter p_k on QoE. Preference logic defines four types of preferences. To determine which preference the relationship between p_j and p_k belongs to, we define the distinction index $qan_{p_j - p_k}$ as

$$qan_{p_j - p_k} = \left|\mathbf{k} * (\vec{e})^{1-(r_{p_j} - r_{p_k})^2}\right| \tag{5}$$

where $\vec{e}$ is a difference vector, specified as $(e^{1-(r_{p_j} - r_{p_k})^2}, 1 - (r_{p_j} - r_{p_k})^2)$, and $\mathbf{k}$ is affine matrix. We expect the range of $qan_{p_j - p_k}$ to fall into $(0, 4)$, which exactly correspond to the four preferences. Therefore, we define $\mathbf{k}$ as $\mathbf{k} = 4/(\vec{e})^*$, where

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

TABLE III
NOTATIONS

Symbol	Definition	Symbol	Definition
$c_i(i \in z_d^+)$	A preference description	qoe	QoE
$d(p_j)$	The demand value for p_j	R_{pl}	Set of preferences
$E, E_l(l \in z_d^+)$	Sets in solution space	$R_{xy}(c_i)$	The right set of c_i
$F, f_m(m \in z_d^+)$	Set of flows and a single flow	r_{p_j}	Correlation degree of p_j
$L_{xy}(c_i)$	The left set of c_i	sd_{f_m}	QoE satisfaction degree for f_m
P	Set of QoS parameters	w, w'	Instances of classes
$p_j, p_k(j, k \in z_d^+)$	QoS parameters	$x, y \in \{\nabla, \Delta\}$	Preference relationships
pb, pd, pj, pl	Bandwidth, Delay, Jitter, Loss,	z_d^+	Natural number greater than 0
pp, ps, pu	Price, Security, User level	∇, Δ	Preference marks
osd	Overall QoE satisfaction degree	Ψ, Φ	Instances of preference relationship
$Q, Q_n(n \in z_d^+)$	Universal set of classes and a class	$\aleph_p, \aleph_c, \aleph_f$	The number of QoS parameters, classes and flows
$qan_{p_j - p_k}$	Distinction index between p_j and p_k		

$(\vec{e})^*$ means $\vec{e}$ is rotated to the polar coordinate, and then moved to the pole; $4/(\vec{e})^*$ means stretching to 4 times its length after the vector $\vec{e}$ is transformed by $(\vec{e})^*$. Thus, the length of vector $\vec{e}$ after being transformed by $\mathbf{k}$ is

$$qan_{p_j - p_k} = \frac{4}{e-1}\left(e^{1-(r_{p_j} - r_{p_k})^2} - 1\right). \tag{6}$$

In (6), if $r_{p_j} = r_{p_k}$, then there is no preference relationship between p_j and p_k. If $r_{p_j} < r_{p_k}$, exchange p_j and p_k. Finally, the preference between p_j and p_k is

$$p_j{}^x{>}^y\, p_k : (x, y) \stackrel{\Delta}{=} tr\left(\left\lfloor\, qan_{p_j - p_k} + 1 \right\rfloor\right) \tag{7}$$

where $x, y \in \{\nabla, \Delta\}$. $\lfloor qan_{p_j - p_k} + 1 \rfloor = i \in \{1, 2, 3, 4\}$, and $tr(i) \in \{(\nabla, \nabla), (\Delta, \nabla), (\nabla, \Delta), (\Delta, \Delta)\}$, corresponding to the four preferences. For example, $pd^{\nabla}{>}^{\Delta}\, pl$ means that delay is "careful" preferred to loss. The preferences for other parameters can also be deduced by Algorithm 1. These preferences are combined into R_{pl} to form the QoS requirements of f_m:

$$R_{pl} = \bigcup \{c_i = c_i^l{}^x{>}^y c_i^r\} \tag{8a}$$

$$\text{s.t. } c_i^l = p_{j-d(p_j)} \tag{8b}$$

$$c_i^r = p_{k-d(p_k)} \tag{8c}$$

where $d(p_j)$ and $d(p_k)$ represent the demand value for QoS parameters p_j and p_k, which are obtained by fixed observation [16] as shown in Section V-A.

C. Logic Reasoning

Based on R_{pl}, optimal candidates can be derived from the QoS classes: $E = \{w | M_{plm}, w \vdash R_{pl}, w \in Q\}$, where the logical symbol of $\vdash$ means that it is deducible. The QoS classes in E can best satisfy R_{pl} for flow f_m. In order to obtain E, two definitions are given below:

Definition 1: The ordered partition of Q. $(E_1, \ldots, E_n)$ is the ordered partition of Q, if and only if: (i) E_i $(i = 1, 2, \ldots, n)$ is non-empty set; (ii) $E_1 \cup E_2 \cup \cdots \cup E_n = Q$; (iii) $\exists i, j, E_i \cap E_j = \varnothing, i \neq j$. The ordered partition on Q corresponds to a full forward sequence, satisfying $\forall w, w' \in Q$, if $w \in E_i, w' \in E_j$, then $i \leq j$, if and only if $w \succ w'$.

Algorithm 1: Modeling QoS Requirements

Input: $p_j(i), p_k(i)$, and $qoe(i) (i = 1, 2, \ldots, I)$
Output: $p_{j\_d(p_j)}{}^x{>}^y\, p_{k\_d(p_k)}$
Observe $p_j(i)$, $p_k(i)$ to obtain $d(p_j)$, $d(p_k)$ as in [16];
Normalize p_j, p_k and qoe by (3);
for $p_a(a = j, k)$ **do**
  Compute $\Upsilon_{p_a}(i)$ and $\lambda x_a(i)$ by (4b) and (4c);
  Obtain the correlation degree r_{p_a} by (4a);
Compare r_{p_i} and r_{p_k}:
  if $r_{p_i} = r_{p_k}$ **then** terminate the algorithm;
  if $r_{p_i} < r_{p_k}$ **then** exchange p_j and p_k;
Compute the distinction index $qan_{p_j - p_k}$ by (6);
Define $q = \lfloor qan_{p_i - p_j} + 1 \rfloor$, and judge:
  if $q = 1$ **then** $x = \nabla, y = \nabla$/*$p_{i\_d(p_i)}{}^{\nabla}{>}^{\nabla}\, p_{k\_d(p_k)}$*/;
  if $q = 2$ **then** $x = \Delta, y = \nabla$/*$p_{i\_d(p_i)}{}^{\Delta}{>}^{\nabla}\, p_{k\_d(p_k)}$*/;
  if $q = 3$ **then** $x = \nabla, y = \Delta$/*$p_{i\_d(p_i)}{}^{\nabla}{>}^{\Delta}\, p_{k\_d(p_k)}$*/;
  if $q = 4$ **then** $x = \Delta, y = \Delta$/*$p_{i\_d(p_i)}{}^{\Delta}{>}^{\Delta}\, p_{k\_d(p_k)}$*/;

Definition 2: The pairs of $L_{xy}(c_i)$ and $R_{xy}(c_i), x, y \in \{\nabla, \Delta\}$. The former is called the left set and the latter is the right set. $R_{pl} = \bigcup c_i$ $(i = 1, 2, \ldots, c)$ is the QoS requirements of flow f_m. $\bigcup_i \{w | M_{plm}, w \vdash c_i, w \in Q\}$ is the output of R_{pl}, which is converted to $\bigcup_i (L_{xy}(c_i), R_{xy}(c_i))$. According to different preferences, the left and right sets are defined below.

$$\begin{aligned} {}^{\nabla}{>}^{\nabla} : L_{\nabla\nabla}(c_i) &= \{w | M_{plm}, w \vdash c_i^l \wedge \neg c_i^r, w \in Q\} \\ R_{\nabla\nabla}(c_i) &= \{w | M_{plm}, w \vdash \neg c_i^l \wedge \neg c_i^r, w \in Q\}. \end{aligned} \tag{9a}$$

$$\begin{aligned} {}^{\Delta}{>}^{\nabla} : L_{\Delta\nabla}(c_i) &= \{w | M_{plm}, w \vdash c_i^l, w \in Q\} \\ R_{\Delta\nabla}(c_i) &= \{w | M_{plm}, w \vdash \neg c_i^l \wedge \neg c_i^r, w \in Q\}. \end{aligned} \tag{9b}$$

$$\begin{aligned} {}^{\nabla}{>}^{r} : L_{\nabla\Delta}(c_i) &= \{w | M_{plm}, w \vdash c_i^l \wedge c_i^r, w \in Q\} \\ R_{\nabla\Delta}(c_i) &= \{w | M_{plm}, w \vdash \neg c_i^r, w \in Q\}. \end{aligned} \tag{9c}$$

$$\begin{aligned} {}^{\Delta}{>}^{\Delta} : L_{\Delta\Delta}(c_i) &= \{w | M_{plm}, w \vdash c_i^l, w \in Q\} \\ R_{\Delta\Delta}(c_i) &= \{w | M_{plm}, w \vdash \neg c_i^r, w \in Q\}. \end{aligned} \tag{9d}$$

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

Based on the above $L_{xy}(c_i)$ and $R_{xy}(c_i)$, the logic reasoning is carried out to obtain the ordered partition of Q: $(E_1, E_2, \ldots, E_n)$, where the QoS classes in E_1 are superior to that in E_2, and the QoS classes in E_2 are superior to that in E_3, and so on. In this paper, f_m is aggregated into the QoS class that best meets R_{pl}. Therefore, only E_1 is kept, while $E_2, \cdots, E_n$ can be ignored. Thus, we reconstructed the reasoning steps in [35] to form the formula

$$E=\begin{cases} w|\forall\left(L_{xy}(c_i)-E, R_{xy}(c_i)\right), \\ \qquad w\notin R_{xy}(c_i)\vdash xy:=\nabla\nabla\cup'\Delta\nabla \\ w|\forall\left(L_{xy}(c_i), R_{xy}(c_i)-E\right), \\ \qquad w\notin L_{xy}(c_i)\vdash xy:=\Delta\Delta\cup'\nabla\Delta \\ w|\forall\left(L_{xy}(c_i)-E, R_{xy}(c_i)-E\right), \\ \qquad w\notin R_{xy}(c_i)\wedge L_{xy}(c_i)\vdash else. \end{cases} \tag{10}$$

Eq. (10) contains three cases:

i) When $xy:=\nabla\nabla\cup'\Delta\nabla$, i.e., R_{pl} contains only $\{^{\nabla}>^{\nabla}\}$ or $\{^{\nabla}>^{\nabla}\}\cup\{^{\Delta}>^{\nabla}\}$, E_1 will be forward reasoned from $L_{xy}(c_i)$. Forward reasoning means: E_1 that best meets the requirements is first reasoned out, then suboptimal E_2, then E_3, and so on (see Algorithm 2). We only need E_1 in this paper. Therefore, the end condition of Algorithm 2: $Q \neq \varnothing$ is revised to $E_1 \neq \varnothing$, and the reasoning is stopped after E_1 is derived.

ii) When $xy:=\Delta\Delta\cup'\nabla\Delta$, i.e., R_{pl} contains only $\{^{\Delta}>^{\Delta}\}$ or $\{^{\Delta}>^{\Delta}\}\cup\{^{\nabla}>^{\Delta}\}$, E_1 will be backward reasoned from $R_{xy}(c_i)$ (see Algorithm 3). Backward reasoning means: E_1, that is most unable to meet the requirements, is reasoned out first, then E_2, and so on. We change the order of E_i: $E'_j = E_{i-j+1}$ to ensure E_1 is optimal.

iii) R_{pl} contains neither $\nabla\nabla\cup'\Delta\nabla$ nor $\Delta\Delta\cup'\nabla\Delta$, the forward and backward reasoning are combined: i) first, the backward reasoning from $R_{xy}(c_i)$ is carried out; ii) the classes produced by the backward reasoning will be deleted from Q; iii) for the rest of the classes in Q, the forward reasoning is carried out.

Algorithm 2: Forward Preference Reasoning

Input: R_{pl}, Q
Output: E
for *each* $c_i \in R_{pl}$ **do**
  Obtain $(L_{xy}(c_i), R_{xy}(c_i))$ by (9a)–(9d);
$l \leftarrow 0$;
while *($Q \neq \varnothing$)* **do**
  $l \leftarrow l+1$;
  $E_l = \{w|\forall(L(c_i), R(c_i)) \in R_{pl}, w \notin R_{xy}(c_i)\}$;
  if $E_l = \varnothing$ **then** terminate to check the conflicts;
  Delete the elements in E_l from Q;
  if $L_{xy}(c_i) = \varnothing$ **then** delete $(L_{xy}(c_i), R_{xy}(c_i))$;
  Replace $(L_{xy}(c_i), R_{xy}(c_i))$ by $(L_{xy}(c_i)$-$E_l, R_{xy}(c_i))$;
$E = E_1$;

Here, the following three points need to be emphasized:

i) The number of iterations. As shown in Algorithms 2 and 3, the iteration terminating condition is $L_{xy}(c_i) - E = \varnothing$ or $R_{xy}(c_i) - E = \varnothing$. Therefore, even in the worst case, the calculation will end up in $\min(\aleph_p, \aleph_c)$ iterations, where $\aleph_p$ and $\aleph_c$ refer to the number of parameters and classes.

ii) By (10), we can calculate the QoS parameter that flow f_m is insensitive to. When parameter p_j changes, we get set E'. If $E = E'$, and no matter how p_j changes, E remains unchanged, then flow f_m is insensitive to p_j. Therefore, when the network environment changes, our proposed method will only update the flows which are sensitive to parameter p_j. Thus PLM provides local adjustment, which is a breakthrough over some of the existing aggregation methods with global adjustment as shown in Sections V-C and V-D.

iii) In Algorithms 2 and 3, when set E_i is empty, it indicates that there are some preference conflicts in R_{pl}. Conflicts in R_{pl} may cause E_i to be empty, and as a result, the logic reasoning would terminate. Therefore, these conflicts should be eliminated.

Algorithm 3: Backward Preference Reasoning

Input: R_{pl}, Q
Output: E
for *each* $c_i \in R_{pl}$ **do**
  Obtain $(L_{xy}(c_i), R_{xy}(c_i))$ by (9a)–(9d);
$l \leftarrow 0$;
while *($Q \neq \varnothing$)* **do**
  $l \leftarrow l+1$;
  $E_l = \{w|\forall(L_{xy}(c_i), R_{xy}(c_i)) \in R_{pl}, w \notin L_{xy}(c_i)\}$;
  if $E_l = \varnothing$ **then** terminate to check the conflicts;
  Delete the elements in E_l from Q;
  Replace $(L_{xy}(c_i), R_{xy}(c_i))$ by $(L_{xy}(c_i), R_{xy}(c_i) - E_l)$;
  if $R_{xy}(c_i) = \varnothing$ **then** delete $(L_{xy}(c_i), R_{xy}(c_i))$;
$E'_j = E_{l-j+1}$; $E = E'_1$;

D. Elimination of Preference Conflicts

As shown in Section III-B, the user's subjective experience may vary sometime due to the variation of feelings, surroundings, etc., which would have direct effect on QoE, and consequently the preferences may be changed. In another case, when a user moves from one place to another, if there are new QoS metrics to be taken into account, then the preferences may also be changed. When the newly generated preferences are combined into the existing ones as in Eqs. (16a)–(16d), conflicts may possibly be aroused. For example, suppose that the combined set of preferences contains: $R_{pl} = \{pd_{100}{}^{\nabla}>^{\Delta}pl_{0.01}, pl_{0.02}{}^{\nabla}>^{\Delta}pb_{100}, pb_{50}{}^{\nabla}>^{\Delta}pd_{200}\}$, where we can find a loop as shown in Fig. 3. Note that the probability of conflicts is very low. However, conflicts must be paid enough attention to because they may cause E_i to be empty and consequently the algorithm would be terminated. Only when conflicts are eliminated, can the system continue to work. The simplest way to eliminate conflicts is to break such loops. In practice, we just delete a preference to achieve

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

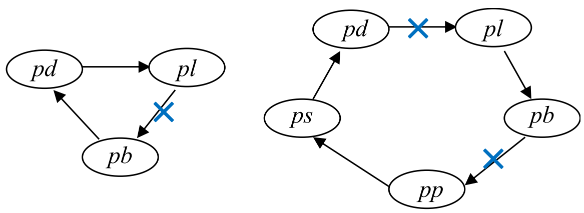

Fig. 3. Distribution of flows for fixed requirements.

this goal, i.e., two preferences are merged into one:

$$p_{i\_d(p_i)}\{\Psi\}p_{j\_d(p_j)} \wedge p_{j\_d(p_j)}\{\Phi\}p_{k\_d(p_k)} \rightarrow p_{i\_d(p_i)}\{\Psi;\Phi\}p_{k\_d(p_k)} \tag{11}$$

where Ψ and Φ represent two preference relationships. For example, the above preferences $\{pd_{100}{}^{\nabla}{>}^{\Delta}pl_{0.01}$, $pl_{0.02}{}^{\nabla}{>}^{\Delta}pb_{100}, pb_{50}{}^{R}{>}^{\Delta}pd_{200}\}$ are merged into $\{pl_{0.02}{}^{\nabla}{>}^{\Delta}pb_{100}, pb_{50}{}^{\nabla}{>}^{\Delta}pd_{200}\}$.

Here, we prove the rationality of Eq. (11). Certainly, it can only assure that conflicts are eliminated, but cannot provide the optimal solution to eliminate conflicts since there is couple of ways to break the loop as shown in Fig. 3, which will be further studied in our future work.

Proof of rationality:

Define $H_{pl}(\Psi,\succ) = \{w|M_{plm}, w \vdash \Psi$, and for any $w' \in Q$, if $M_{plm}, w' \vdash \Psi$, then $w \succ w'\}$; $h_{pl}(\Psi,\succ) = \{w|M_{plm}, w \vdash \Psi$, and for any $w' \in Q$, if $M_{plm}, w' \vdash \Psi$, then $w' \succ w\}$.

According to the definition, $H_{pl}(\Psi \wedge \Phi,\succ) = \{w|M_{plm}, w \vdash \Psi \wedge \Phi$, and for any $w' \in Q$, if $M_{plm}, w' \vdash \Psi \wedge \Phi$, then $w' \succ w$. Furthermore, because $\succ$ is reflexive, complete, and transmitted, there must be $\forall w_2 \in H_{pl}(\Psi,\succ)$, $\forall w_3 \in H_{pl}(\Psi,\succ)$ satisfying $w_3 \succ w_2$ or $w_2 \succ w_3$, and, if $w_3 \succ w_2$, then $H_{pl}(\Psi\wedge\Phi,\succ) \subseteq H_{pl}(\Psi,\succ)$; if $w_2 \succ w_3$, then $H_{pl}(\Psi\wedge\Phi,\succ) \subseteq H_{pl}(\Phi,\succ)$.

In the same way, $h_{pl}(\Psi \wedge \Phi,\succ) = \{w|M_{plm}, w \vdash \Psi \wedge \Phi$, and for any $w' \in Q$, if $M_{plm}, w' \vdash \Psi \wedge \Phi$, then $w' \succ w$. Furthermore, because $\succ$ is reflexive, complete, and transmitted, there must be $\forall w_2 \in h_{pl}(\Psi,\succ)$, $\forall w_3 \in h_{pl}(\Psi,\succ)$ satisfying $w_3 \succ w_2$ or $w_2 \succ w_3$, and, if $w_3 \succ w_2$, then $h_{pl}(\Psi \wedge \Phi,\succ) \subseteq h_{pl}(\Psi,\succ)$;if $w_2 \succ w_3$, then $h_{pl}(\Psi\wedge\Phi,\succ) \subseteq h_{pl}(\Phi,\succ)$.

Hence, the result of preference reasoning for $\Psi\wedge\Phi$ is a subset of that of Ψ or Φ. That is, the merge of the conflicting preferences $p_{i\_d(p_i)}\{\Psi\}p_{j\_d(p_j)}$ and $p_{j\_d(p_j)}\{\Phi\}p_{k\_d(p_k)}$ by (11) can guarantee the reasoning result of $p_{i\_d(p_i)}\{\Psi;\Phi\}p_{k\_d(p_k)}$ is a subset of that of $p_{i\_d(p_i)}\{\Psi\}p_{j\_d(p_j)}$ or $p_{j\_d(p_j)}\{\Phi\}p_{k\_d(p_k)}$. Accordingly, the conflict treatment (11) assures E_i not to be empty and thus the preference reasoning avoids to be terminated.

E. Final Selection

When there are several optimal options in E, we use the decision theory to implement the final selection [36]. Therefore, we define the matrix as:

$$\mathbf{A} = (a_{ij})_{\aleph_d\times\aleph_d}|a_{ij} = qan_{crt(i)-crt(j)} \tag{12a}$$

$$s.t.\ a_{ij} > 0;\quad a_{ji} = 1/a_{ij};\ a_{ii} = 1 \tag{12b}$$

$$\mathbf{A}\vec{v} = \xi_{max}\vec{v} \tag{12c}$$

where $crt(i)$ and $crt(j)$ represent decision criteria, e.g., price and resource utilization. $\aleph_d$ is the number of decision criteria. The calculation method of distinction index $qan_{crt(i)-crt(j)}$ can be found in Section III-B. ξ_{max} is the largest characteristic root of $\mathbf{A}$. $\vec{v}$ is the characteristic vector. Based on $\vec{v}$, the following decision function is defined to aggregate flow f_m into class Q_n:

$$Q_n \triangleq \arg\min\{N_{Q_n}\log\vec{v}\vec{y}_{Q_n}|_{(Q_n\in E)}\} \tag{13}$$

where $\vec{y}_{Q_n}$ is the evaluation vector which includes the components of the price, the resource utilization for class Q_n, etc. N_{Q_n} is the number of flows in class Q_n. If there is only one criterion (e.g., price) in Eq. (12a) and the prices for $\{Q_n \in E\}$ are the same, then Eq. (13) can be simplified as $Q_n \triangleq \arg\min\{N_{Q_n}\}$. In this case, the flow would be grouped into the queue with the least number of flows.

F. Complexity Analysis

As shown in Fig. 2, the proposed aggregation scheme PLM consists of four main components: modeling QoS requirements (Section III-B), eliminating preference conflicts (Section III-D), logic reasoning (Section III-C) and final selection (Section III-E).

In this paper, preferences are used to model the imprecise QoS requirements of different flows. As shown in Algorithm 2, the complexity of establishing the preference between p_j and p_k is $\mathrm{O}(\aleph_f)$, where $\aleph_f$ is the number of the flows. If there are $\aleph_p$ aggregation criteria (e.g., jitter and break), then the time complexity to achieve preferences is $\mathrm{O}(\aleph_p\aleph_f)$.

Furthermore, as described in Section III-D, the probability of preference conflicts is very small. Therefore, the computation costs of PLM are mainly produced by the part on logic reasoning. As shown in Algorithms 2 and 3, in some cases (e.g., the forward preference reasoning), the logic reasoning would stop just after the first iteration. Even in the worst case, the logic reasoning would end up in $\min(\aleph_p,\aleph_c)$ iterations, where $\aleph_c$ is the number of the QoS classes. The complexity of each iteration is $\mathrm{O}(\aleph_c)$. Thus, the complexity for logic reasoning is no more than $\mathrm{O}(\min(\aleph_p,\aleph_c)\aleph_c)$. When $\aleph_f$ flows are aggregated, the time complexity is $\mathrm{O}(\min(\aleph_p,\aleph_c)\aleph_c\aleph_f)$.

Note that there may be several QoS classes in set E, so the final selection is made by Eq. (13). In practice, we usually select the class with more free resources (i.e., the queue that has the least flows) from E as shown in Section III-E. Thus, the complexity for final selection is no more than $\mathrm{O}(\aleph_c)$, which can be neglected when compared with that of logic reasoning.

As a result, the total time complexity of PLM is $\mathrm{O}(\min(\aleph_p,\aleph_c)\aleph_c\aleph_f+\aleph_p\aleph_f)$. It can be seen that the algorithm complexity is a linear function of $\aleph_p$ and $\aleph_f$. The computation costs of the proposed method are relatively small, which will continue to be evaluated in subsequent Section V-G.

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
PARAMETERS FOR 6 QOS CLASSES

Code	Classes	Typical Case	Delay (ms)	Bandwidth (kbps)	Loss	Jitter (ms)
Q_1	Realtime	Mikogo	50	128	10^{-3}	50
Q_2	Streaming	Tudou	100	128	10^{-1}	100
Q_3	Interactive	Game	100	64	10^{-3}	50
Q_4	Background	FTP	250	64	10^{-1}	50
Q_5	Voice	VoIP	50	16	10^{-2}	10
Q_6	Text	WWW	500	32	10^{-3}	100

IV. DATASETS AND PERFORMANCE METRICS

A. Datasets

By analyzing the existing QoS framework adopted by international standard organizations, such as IETF and NGN, we summarized the six QoS classes of ITU-T Y.1541 as shown in Table IV. Besides, there are 3 datasets used in this paper:

- NJUPT dataset [37]. It contains many types of flows, which were captured by Wireshark in the campus network of Nanjing University of Posts and Telecommunications in 2018.
- ISP dataset. It contains some types of flows not available in NJUPT, such as the monitoring systems, teleconference, and e-commerce, which were collected in an ISP data center located in a southern city of China in 2017 (names are omitted due to privacy reason).
- UNB ISCX Network Traffic (VPN-nonVPN) trace 2016 (short for UNB dataset). We downloaded 28G network traffic data from the official website [38].

B. Performance Metric

Flows are aggregated into the QoS classes in Table IV. Under the QoS framework, the aggregation performance is reflected in whether the QoS requirements of flows are met. Referring to the preference models in economics [28], we define the QoE satisfaction degree as

$$sd_{F_m} = \sum_{p_k \in P} w(p_k) \cdot sat(p_k) \tag{14}$$

where $sat(p_k) = \{0,1\}$ is a two-value function. If p_k of flow f_m is within or close to the QoS class (the range is set within 5% in this paper), then $sat(p_k) = 1$, which means the requirement for p_k is met. $w(p_k)$ is the weight for QoS parameters p_k. Therefore, the range of sd_{f_m} is $[0, 1]$. If there are $\aleph_f$ flows, then the overall QoE satisfaction degree is

$$osd = \frac{1}{\aleph_f} \sum_m sd_{f_m}. \tag{15}$$

V. EVALUATION

A. Aggregating a Single Flow to QoS Class

In this subsection, the video conferencing flows from the NJUPT dataset are selected. Four QoS parameters in Table IV are studied here, including delay, bandwidth, loss and jitter. The aggregation process is demonstrated as follows.

1) Data Preparation: Traffic flow contains a lot of information, such as the arrival time of each packet and packet size. In fact, all flows in the datasets have such information, through which we can get the parameters of delay, jitter, bandwidth, and loss. The values of QoE are recorded with the method in [31]. According to the work, users' QoE are given in 5 levels, i.e., {1, 2, 3, 4, 5}. Here, 5 means users get the best experience and 1 means the worst experience. If 9 conferencing flows are sampled, then we have:

$$pd(i) = \{72, 59, 34, 41, 65, 51, 37, 86, 46\} \tag{16a}$$
$$pl(i) = \{0.0025, 0.002, 0.001, 0.001, 0.0015, 0.002, \tag{16b}$$
$$0.0015, 0.002, 0.001\} \tag{16c}$$
$$qoe(i) = \{3, 4, 5, 5, 4, 4, 5, 3, 5\}. \tag{16d}$$

2) Modeling QoS Requirements With Preferences: As shown in Algorithm 1, the first step of modeling QoS requirements is to scan through $pd(i)$ and $pl(i)$ to obtain the demand value for delay and loss. In this paper, the demand values for QoS parameters are measured by fixed observation [16], i.e., selecting the maximum/minimum when QoE reaches the 5th level as the demand value. Therefore, $d(pd) = 46ms$ and $d(pl) = 0.0015$. Next, r_{p_k} is calculated with (4a) and thus $r_{pd} = 0.892$ and $r_{pl} = 0.247$. Then, compute the distinction index qan_{pd-pl} to be 0.287. Finally, the preference is obtained according to Eq. (7) and it is $pd_{46}{}^{\nabla}{>}^{\nabla}pl_{0.0015}$. The preferences for other parameters can be obtained by the same steps. Thus, the preferences of conferencing flows are $R_{pl} = \{c_1 = pd_{46}{}^{\Delta}{>}^{\nabla}pb_{134}, c_2 = pb_{134}{}^{\nabla}{>}^{\nabla}pl_{0.0015}, c_3 = pl_{0.0015}{}^{\Delta}{>}^{\nabla}pj_{47}\}$.

3) Logic Reasoning: In this case, there are only ${}^{\Delta}{>}^{\nabla}$ and ${}^{\nabla}{>}^{\nabla}$ in R_{pl}, so we use Algorithm 2 to identify the most suitable class from $Q = \{Q_1, Q_2, Q_3, Q_4, Q_5, Q_6\}$ (if R_{pl} contains ${}^{\Delta}{>}^{\Delta}$, then we will use Algorithm 3). The first part of Algorithm 2 is to convert R_{pl} to sets $L_{xy}(c_i)$ and $R_{xy}(c_i)$:

- $L_{\Delta\nabla}(c_1) = \{w|M_{plm}, w \vdash c_1^l\}$, where $c_1^l = pd_{46}$. From Table IV, find the classes that can satisfy pd_{46}. Therefore, $L_{\Delta\nabla}(c_1) = \{Q_1, Q_5\}$.
- $R_{\Delta\nabla}(c_1) = \{w|M_{plm}, w \vdash \neg c_1^l \wedge \neg c_1^r\}$, where $c_1^l = pd_{46}$, and $c_1^r = pb_{134}$. Find the classes that cannot satisfy pd_{46} and pb_{134}, yielding $R_{\Delta\nabla}(c_1) = \{Q_3, Q_4, Q_6\}$.
- $L_{\nabla\nabla}(c_2) = \{w|M_{plm}, w \vdash c_2^l \wedge c_1^r\}$, where $c_2^l = pb_{134}$, and $c_2^r = pl_{0.0015}$. Find the classes that can satisfy pb_{134} and $pl_{0.0015}$. Therefore, $L_{\nabla\nabla}(c_2)$ are $\{Q_1\}$.

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

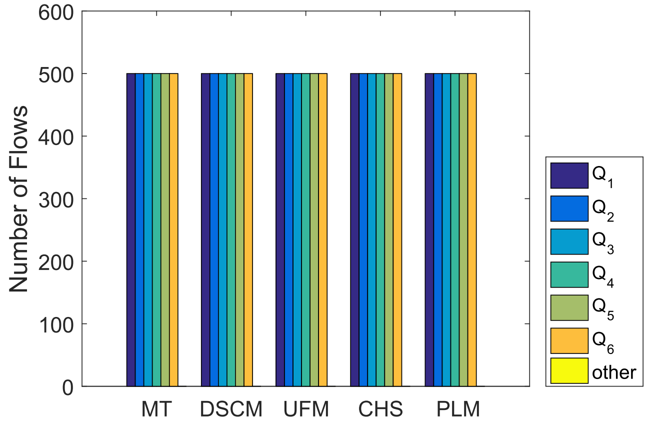

Fig. 4. Distribution of flows for fixed requirements.

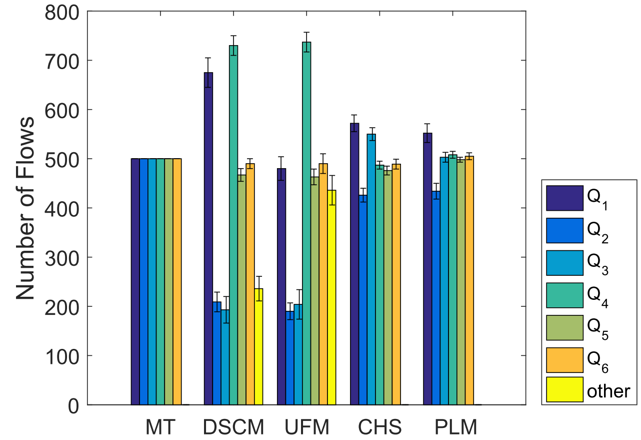

Fig. 5. Distribution of flows for variable requirements.

- $R_{\nabla\nabla}(c_2) = \{w|M_{plm}, w \vdash \neg c_2^l \wedge \neg c_2^r\}$. Therefore, $R_{\nabla\nabla}(c_2)$ are $\{Q_4, Q_5\}$.
- $L_{\Delta\nabla}(c_3) = \{w|M_{plm}, w \vdash c_3^l\} = \{Q_1, Q_3, Q_6\}$;
- $R_{\Delta\nabla}(c_3) = \{w|M_{plm}, w \vdash \neg c_3^l \wedge \neg c_3^r\} = \{Q_2, Q_4\}$.

The second part of Algorithm 2 is to reason out the optimal class for the flow with several iterations. As explained in Section III-F, in the worst case, the logic reasoning would require $\min(\aleph_p, \aleph_c)$ iterations. Therefore, with Algorithm 2, we get an ordered partition of $Q : \{\{Q_1\}, \{Q_5\}, \{Q_3, Q_6\}, \{Q_2, Q_4\}\}$, where $E_1 = \{Q_1\}$, $E_2 = \{Q_5\}$, $E_3 = \{Q_3, Q_6\}$, $E_4 = \{Q_2,\ Q_4\}$. The classes in E_1 best meet the QoS requirements, the classes in E_2 are suboptimal, and the classes in E_4 are the worst. Thus the conferencing flows are aggregated into Q_1.

Considering two new parameters (e.g., price and even qualitative security) are added into aggregation, the preferences would be obtained in the same way by Algorithm 1 as above. The mapping from flows to QoS classes is also in the same way by logic reasoning as above. We do not need to make any adjustment to the algorithms of PLM. Here, we can see that:

- PLM can effectively handle hybrid requirements considering both quantitative and qualitative parameters, such as pb_{128} and ps_{High}, while the existing aggregation methods cannot.
- In PLM, parameters p_k can be deleted or added without affecting the reasoning algorithm, which is different from the existing methods, such as the parametric model $f(\theta, \lambda, r)$ by Purwanto *et al.* [42], in which parameters $\{\theta, \lambda, r\}$ cannot be changed at all.

B. Distribution of Flows After Aggregation

In this subsection, i) 3000 flows are randomly selected, and each class (from Q_1 to Q_6) consists of 500 flows; ii) only 4 QoS parameters and 6 QoS classes are considered as shown in Table IV; iii) assume the length of the queues for all QoS classes (see Fig. 1) is infinite.

Several aggregation methods are compared, including MT [39], UFM [40], DSCM [16], and CHS [41]. In [39], a flow mapping table (MT) is manually obtained to group the flows into the classes. Jin *et al.* [40] developed the utility function model (UFM) to control the traffic aggregation. In [16], Wang *et al.* proposed a dynamic service class mapping scheme (DSCM) for transporting traffic over HNs. Wu *et al.* [41] presented multi class aggregation structure called the chain and hierarchical structure (CHS) based on the near neighbor classifier.

For all methods, the system parameters, thresholds, weights, etc., are trained and tuned to the optimal state. Therefore, flows can be aggregated into the right QoS classes as shown in Fig. 4. However, in reality, the QoS requirements, parameters, and classes are often changing at the edge in HNs, as described in Section II. Therefore, in the following sections, we will discuss the impact of variable requirements R_{pl}, parameters P, and classes Q on flow aggregation.

C. Variable QoS Requirements

In this subsection, some flows with variable QoS requirements are selected, such as the Youku videos and SDO games. For other flows, we add a random quantity to the QoS requirements to simulate changes. Specifically, for flows of class Q_2, we increase their QoS requirements by a random quantity. For that of Q_3, we decrease their QoS requirements by a random quantity. The range of the quantity is set within 5%. 3000 flows (500 flows for each class) are re-aggregated and the distribution is presented in Fig. 5. We repeat this experiment 10 times, and the differences among experiments are also demonstrated in the figure.

It can be seen that when the QoS requirements change, MT is completely incapable of adaptation. All flows are aggregated in the same way as before, regardless of whether QoS requirements are changed or not.

UFM and DSCM are sensitive to changing environment, but the results are not as good as expected. For UFM and DSCM, the QoS requirements of flows are represented by numerical values, which exceed the thresholds of Q_2 and Q_3, and happen to fall into the thresholds of Q_1 and Q_4. Therefore, the queues Q_2 and Q_3 are almost empty, while Q_1 and Q_4 are too long. In real networks, long queue or full queue will cause an increase in delay and packet loss. In general, it is the sensitivity to threshold that results in imbalance in flow

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

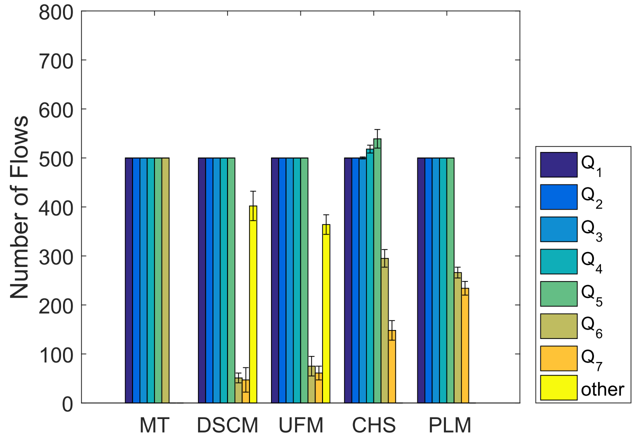

Fig. 6. Distribution of flows for changing classes.

aggregation [43]. However, if the value does not fall within the threshold of any QoS class, the flow would be aggregated into queue "other". In real networks, flows in queue "other" will not be allocated with appropriate system resources.

CHS and PLM present a good adaptability. They can effectively respond to the changes. For PLM, according to the algorithms in Sections III-B and III-C, changes in the QoS requirements will lead to the changes in $c_i = p_{j\_d(p_j)}{}^{x}{>}^{y} p_{k\_d(p_k)}$, and consequently result in changes in set $E := \{w \vdash R_{pl}\}$. PLM can respond to the changes, and it does not overreact like UFM and DSCM do. According to (9a)-(9d), the reasoning result of $L_{xy}(c_i)$ and $R_{xy}(c_i)$ depends on the relative positions of p_j and p_k in the QoS class. In most cases, the absolute variation of p_j and p_k does not affect these relative positions, and thus the logic reasoning results remain unchanged. In general, PLM can effectively respond to the changes of QoS requirements without overreaction. Similarly, for CHS, there is no overreaction either. When the QoS requirements change, the valid range $\min(D_j^*)$, $\max(D_j^{\sim})$ is properly adjusted, where $\min(D_j^*)$ and $\max(D_j^{\sim})$ are the maximum and minimum Euclidean distances from sample to cluster point j. Thus it can be seen that CHS does not depend on thresholds and avoid overreaction.

D. Variable QoS Classes

In this subsection, we create a new class Q_7 and expect that the flows originally belonging to Q_6 can be partially re-grouped into queue Q_7. Thus: i) Class Q_6 in Table IV is properly modified by randomly increasing the values within the range of 5%. ii) The difference between Q_6 and Q_7 is less than 10%, and the differences between Q_7 and other QoS classes are more than 30%. The QoS classes exactly refer to the aggregates in Fig. 1. Therefore, when the QoS classes are added or deleted, we just increase or decrease the number of aggregates accordingly. After the QoS classes are revised, the above 3000 flows in Section V-C are re-aggregated. The distribution is shown in Fig. 6.

Apparently, MT still cannot respond to changes. UFM and DSCM can respond to changes, but they do not work very well. In theory, we expect that flows originally belonging to Q_6 can be partially aggregated into Q_7, but in fact, most of them fall into queue "other". The reason is that UFM and DSCM are sensitive to the threshold. Taking UFM as an example, the threshold is set at $D(\lambda) = \max L(F, y)$. Thus flow F_m would be aggregated into class y if and only if the utility function $L(\cdot)$ does not exceed the threshold. Note that the threshold $D(\lambda)$ for class Q_1 is 2.53, and it is 0.87 for Q_6. The threshold for new class is set as the lowest one in this system, i.e., $D(\lambda) = \min_x \max L(x, y) = 0.87$ for Q_7. $L(\cdot)$ would easily exceed the threshold of 0.87, which causes these flows to fall into queue "other".

CHS dynamically adjusts the aggregation by the cluster centers when the QoS classes change. Therefore, CHS is insensitive to threshold, but sensitive to cluster centers. On the positive side, flows will not be aggregated into queue "other". On the negative side, the changes of Q_6 and Q_7 will have global impact on other queues as shown in Fig. 6. The reasons are as follows: The model of CHS is described as $A^* = [V_1, \ldots, V_m]$, where V_i represents the QoS class. All flows are aggregated into the QoS classes based on the cluster center V_i. Therefore, CHS would not map the flows into queue "other". However, CHS highly depends on the entire sample population. When a cluster point changes, especially when a new cluster point is added, all flows need to be recalculated to adjust the valid range. For CHS, a small change may lead to global influence on the system. As shown in Fig. 6, Q_4 and Q_5 are affected by the changes of Q_6 and Q_7.

PLM overcomes the shortcomings of the above aggregation methods which are sensitive to either threshold or cluster centers. Thus, PLM will not aggregate flows into queue "other", and when QoS classes changes locally, it won't generate global influence like CHS. The reasons are: i) According to Definition 1, PLM implements the ordered partition of $Q = \{Q_1, \ldots, Q_7\}$, resulting in $(E_1, \ldots, E_n)$, $E_1 \cup E_2 \cup \cdots \cup E_n = Q$. E_1 is bound to be a subset of Q. Therefore, flows would not be aggregated into queue "other". ii) E_1 is also a subset of E_1^*, where E_1 is obtained before the QoS classes change, while E_1^* is obtained after the QoS classes change. Taking the QQ flows in Section V-A as an example, we get $E_1 = \{Q_1\}$, which means that Q_1 can best meet the QoS requirements of QQ flows, while $Q_2, \ldots, Q_6$ are not able to meet the requirements well. When Q_6 changes and Q_7 is added, apparently, $Q_2, \ldots, Q_5$ are still unable to meet the requirements. Therefore, we only need to discuss Q_6 and Q_7, i.e., $Q = \{Q_1, Q_6, Q_7\} \vdash E_1^*$. Thus, for PLM, the local change of classes only has local influence.

E. QoE Satisfaction Degree

We expect most of the flows to be properly aggregated into the QoS classes, i.e., the flows' requirements are maximally satisfied. Thus, we use the satisfaction degree to evaluate the performance of aggregation methods. In this subsection, we obtain the statistics of satisfaction degree based on the results of Sections V-B, V-C, and V-D, as shown in Table V. The final result was obtained by averaging the results of 10 runs.

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

TABLE V
OVERALL SATISFACTION DEGREE

Methods	V-B: Fixed requirements	V-C: Variable requirements	V-D: Variable classes
MT	0.927	0.897	0.860
DSCM	0.927	0.855	0.836
UFM	0.927	0.821	0.843
CHS	0.927	0.919	0.914
PLM	0.927	0.913	0.918

Under fixed requirements as in V-B, the system parameters, thresholds, weights, etc., are trained and tuned to the optimal state. Therefore, for all methods, including MT, UFM, DSCM, CHS, and PLM, flows can be aggregated into the right QoS classes, see Fig. 4. Thus their Osd are all the same. However, they are not 1, because some requirements of some flows are not met. For example, a type of video monitoring flow $f_1 = \{25ms, 130kbps, 0.0015, 57ms\}$ is aggregated into $Q_1 = \{50ms, 128kbps, 0.001, 50ms\}$. Obviously, the requirement for delay is not met.

Under variable QoS requirements as in V-C and variable QoS classes as in V-D, the flows are re-aggregated into different QoS classes as shown in Fig. 5 and Fig. 6 respectively. For MT, all flows are aggregated in the same way as before, regardless of whether the QoS requirements or classes have changed, which results in the drop of osd. As for UFM and DSCM, The decline of osd is very significant. This is mainly because some of the flows fall into the "other" class. The flows in queue "other" will not be allocated with appropriate system resources. For CHS and PLM, they can effectively adjust and respond to the changes of QoS requirements and classes, so they have good performance, as shown in Table V.

Discussions: i) Note that the "other" class is defined by the average value in Table IV. If it is defined by the minimum or maximum value, the statistical values of osd will be different for UFM and DSCM. But we have verified that even the "other" class is defined by the maximum value, the osd for UFM turns out to be less than 0.9, which is still lower than that of CHS and PLM. ii) Aggregation of flows also aims to enhance the efficiency of scheduling and transmission. Thus, it is obviously inappropriate to concentrate only on whether flows' requirements are met. Therefore, in next subsection, we use throughput as criteria to further test the performance.

F. Throughput Analysis

In this experiment, the dynamic round robin (DRR) scheduling is exploited to schedule the aggregates [44]. As shown in Fig. 1, flows with the same QoS requirements are grouped into one aggregate. The DRR scheduling will allocate different resources for QoS classes, including the queue length, bandwidth, etc. The DRR scheduling model has n aggregates in total, and each aggregate has m sub-queues to carry m flows. The total length of queue is set to $8m$ times the average length of flows. Here, two issues need to be illustrated.

- Why *the need of the average length*? Because the lengths of flows vary significantly. Short flows (e.g., VoIP) have only hundreds of Byte. Many text flows are below 1MB. Long flows are as many as several MB. For longer flows (e.g., streaming data), we only take 3 minutes of data volume due to limited hardware conditions in the experiment.
- Why is it $8m$ times? There are 6 QoS classes (see Table IV) and plus the "other" class and Q_7, so we have 8 classes in total, corresponding to 8 aggregates in Fig. 1. Each aggregate accommodates m flows. Thus the total length is $8m$ times the average length of flows.

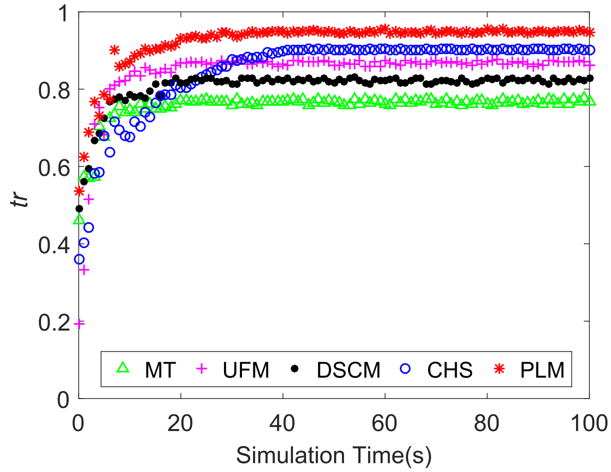

Fig. 7. Throughput for the ELD requirements.

All samples used in this experiment come from Section V-D. The traffic are generated from the samples according to Poisson distribution with intensity $\lambda_p = 90$, which are used to simulate the heavy-load network (in the case of low load, there is no obvious difference among the methods in throughput). The throughput here refers to the normalized throughput: $tr = d_o/d_i$, where d_i is the input data rate, which is measured when the flows enter the queues, while d_o is the output data rate, which is measured when the flows leave the queues.

In Fig. 7, MT, UFM and DSCM show poorer performance in throughput. This is due to: i) For MT, flows are always aggregated according to the original rules, resulting in a higher packet loss. What's more, queue Q_7 is empty, and the scheduling time allocated to Q_7 is completely wasted, thus resulting in low overall throughput. ii) For UFM and DSCM, the queue "other" is too full, which causes the packet loss, and thus the performance in throughput is not good.

CHS is slow in the start-up phase, while other methods can achieve their maximum throughput quickly, which is mainly attributed to its high computation complexity. When class Q_6 changes and class Q_7 is added, the distances between all cluster points and flows are reanalyzed to update the valid range for classes. The complicated computations have significant impact on the throughput at the start-up phase.

PLM performs well in throughput. This is because: i) For the changed QoS classes, PLM only makes local adjustment. Taking QQ flows as an example, the optimal class for QQ is $E_1 = \{Q_1\}$ as shown in Section V-A. When Q_6 changes and Q_7 is added, we can find the optimal class in

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

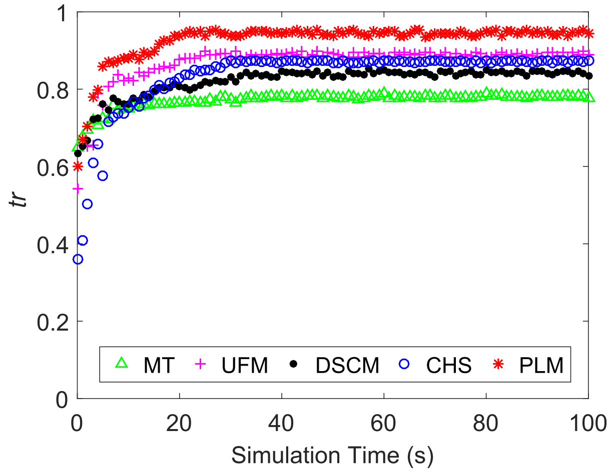

Fig. 8. Throughput for the ELD requirements.

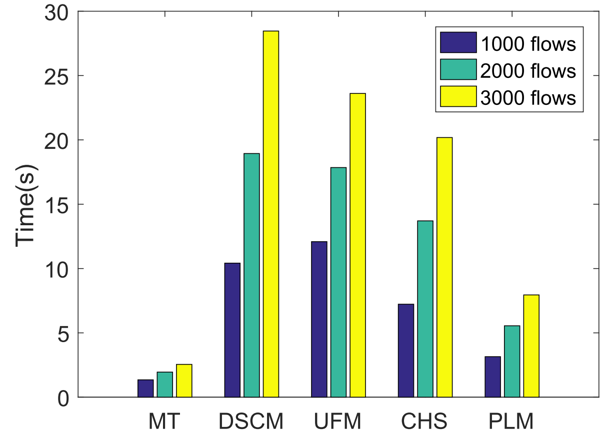

Fig. 9. Time complexity under $\aleph_p = 4$.

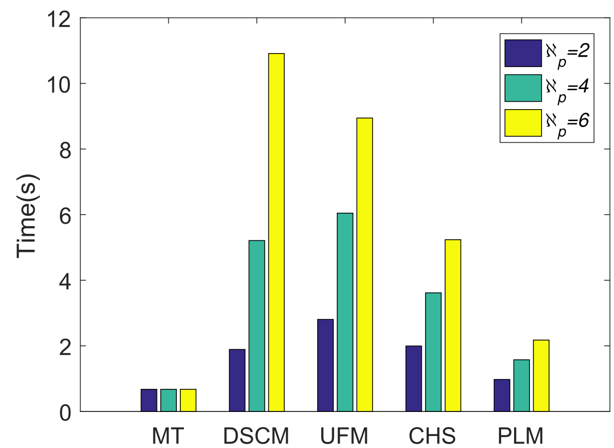

Fig. 10. Time complexity under $\aleph_f = 1000$.

$Q = \{Q_1, Q_6, Q_7\}$. There is no need to update calculation for all classes, resulting in a relatively smaller computation cost in comparison with CHS. ii) According to the algorithm in Section III-D, the final selection of QoS class from E will be properly balanced by $Q_n \triangleq \arg\min\{N_{Q_n}\}$. Therefore, PLM tends to have a balanced distribution of flows, and there will be no imbalanced situation like that occurred in the UFM and DSCM methods. One-sided aggregation in UFM and DSCM cause the corresponding queue to be full, which leads to increase of delay and loss, and in turn reduces the throughput.

For flows that need ELD (below 5 ms) in B5G systems, a minor modification is made on the parameters in Table I. For example, $d(pd)$ of flow F_1 changes to 3, where $d(pd)$ refers to the ELD requirement. There is no need to make any adjustment on PLM algorithm. These flows are re-aggregated by PLM and thus new results on throughput are obtained as shown in Fig. 8. PLM still demonstrates better throughput performance in comparison with the other methods. The reasons mainly lie in that PLM is a local adjustment scheme. Only those sets containing varied $d(pd)$ need to be updated while others remain the same. Therefore, there is no need to update all calculations for flows. That is, it only needs a small computation cost which results in higher throughput.

G. Computational and Space Complexity

In this subsection, we compared the time and space complexities of PLM with the existing works as shown in Table VI. Note that: i) $\aleph_c = 6$ and $\aleph_p = 4$ (see Table IV). That is, $\aleph_p < \aleph_c$, so the time complexity of PLM is $\mathrm{O}(\aleph_p \aleph_c \aleph_f)$. ii) Some of the methods need to be re-trained (e.g., CHS) and re-designed (e.g., DSCM) under varied parameters, so the performance of QoE satisfaction degree and throughput would not be discussed in this section.

From Table VI, it can be seen that the time and space complexities of MT are extremely low. MT maps flows into the target classes according to the flow table, which is designed based on experience. Therefore, the time and space complexities of mapping $\aleph_f$ flows into $\aleph_c$ classes are $\mathrm{O}(\aleph_c \aleph_f)$ and $\mathrm{O}(\aleph_f)$ respectively. As for the DSCM method, Wang *et al.* proposed the rate-delay model based on a series of convolution operations to achieve aggregation. The time complexity of a single convolution operation for rate and delay is $\mathrm{O}({\aleph_p}^2)$. Thus, the total time complexity of $\aleph_c$ convolution operations for $\aleph_f$ flows is $\mathrm{O}({\aleph_p}^2 \aleph_c \aleph_f)$. Based on the utility function, UFM requires storage for all training and testing flows. Therefore, its space complexity is $\mathrm{O}(\aleph_p(\aleph_m + \aleph_f))$. Its time complexity is mainly caused by the calculation of decision process based on the utility function. CHS adopts the nearest neighbor rule, which also needs to store all samples, so the space complexity of CHS is the same as that of UFM. We know that KNN groups flows with $\mathrm{O}({\aleph_p}^2 \aleph_c \aleph_m \aleph_f)$ comparisons. However, CHS combines several KNN classifiers to implement classification. That is, it divides the sample flows into $\aleph_c$ KNN classifiers, thus reducing the number of comparisons to $\mathrm{O}(\aleph_p \aleph_c \log(\aleph_m) \aleph_f)$. PLM does not need to store the training flows, and thus its space complexity is the same as that of DSCM.

In order to verify the theoretical comparisons given in Table IV, we select 1000, 2000, and 3000 flows to calculate the aggregation time, respectively. We run the simulation with MATLAB R2016a on a laptop computer with Win7 professional (64bit/SP1) operating system, Intel I CoITM i5-4210M @ 2.60 GHz, 2 GB memory. As shown in Fig. 9, it takes the proposed PLM method 3.152s to aggregate 1000 flows,

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
COMPARISON OF TIME AND SPACE COMPLEXITY

	Time complexity	Space complexity
MT [39]	$O(\aleph_c \aleph_f)$	$O(\aleph_f)$
DSCM [16]	$O({\aleph_p}^2 \aleph_c \aleph_f)$	$O(\aleph_p \aleph_f)$
UFM [40]	$O({\aleph_p}^2 \aleph_c (\aleph_m + \aleph_f))$	$O(\aleph_p (\aleph_m + \aleph_f))$
CHS [41]	$O(\aleph_p \aleph_c \log(\aleph_m) \aleph_f)$	$O(\aleph_p (\aleph_m + \aleph_f))$
PLM	$O(\aleph_p \aleph_c \aleph_f)$	$O(\aleph_p \aleph_f)$
Parameters	$\aleph_p$: no. of parameters $\aleph_m$: no. of sample flows	$\aleph_c$: no. of classes $\aleph_f$: no. of testing flows

5.561s for 2000 flows, and 7.947s for 3000 flows. MT has the smallest time and space complexities. However, as analyzed in Sections V-E and V-F, MT has the worst performance of QoE satisfaction and throughput. Among the other four methods, the proposed PLM has the best performance in terms of the time and space complexities. The results illustrated in Fig. 9 agree with the theoretical analysis in Table IV.

As analyzed in Section III-F, the total time complexity of PLM is O$(\min(\aleph_p, \aleph_c)\aleph_c\aleph_f + \aleph_p\aleph_f)$. The algorithm complexity is a linear function of $\aleph_p$ and $\aleph_f$. Therefore, when $\aleph_p$ is fixed to 4 and $\aleph_f$ is increased from 1000 to 3000, or $\aleph_f$ is fixed to 1000 and $\aleph_p$ is increased from 2 to 6, the aggregation time of PLM grows linearly as shown in Figs. 9 and 10. Compared with other methods (except MT), The growth rate of PLM is relatively small.

VI. Conclusion

Diffserv for massive heterogeneous traffic in edge computing is challenging. The existing schemes require quantitative QoS requirements, fixed QoS parameters, and static QoS classes, which are typically hybrid, changeable, and dynamic in B5G. Therefore, a novel aggregation method PLM is proposed in this paper. In PLM, the hybrid requirements of flows are modeled in preference, and the most suitable QoS class can be derived by logic reasoning without using a strict threshold. In the dynamic environment with high variability, when the QoS requirements change, or the QoS parameters and even the QoS classes change, PLM can locally adjust the aggregation with low computational cost and make the best use of limited system resources. It has better performance in terms of QoE satisfaction and throughput than the existing hard aggregation methods. Based on the preference logic, PLM provides a general solution to aggregate flows in heterogeneous networks and is expected to improve Diffserv/QoE in B5G.

There is still the need for some potential improvements to our work, e.g., the granularity of QoS classes. The number of classes is an important factor in determining the performance of aggregation. WiMAX (Worldwide Inter Operability of Microwave Access) defines 6 classes including unsolicited grant service, real-time polling service, non-real-time polling service, etc. Wi-Fi and LTE designate 4 and 9 classes respectively. However, there is no standard on QoS classes in B5G. The issue of aggregation granularity needs to be studied in depth along with future B5G technologies.

References

[1] P. Tang, Y. Dong, and S. Mao, "Soft aggregation of multimedia flows based on QoS classes," in *Proc. IEEE 4th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Singapore, Feb. 2019, pp. 599–603.

[2] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019.

[3] Y. He *et al.*, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10433–10445, Nov. 2017.

[4] E. Fitzgerald, M. Pióro, and A. Tomaszwski, "Energy-optimal data aggregation and dissemination for the Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 955–969, Apr. 2018.

[5] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[6] P. Bellavista, A. Corradi, L. Foschini, and D. Scotece, "Differentiated service/data migration for edge services leveraging container characteristics," *IEEE Access*, vol. 7, pp. 139746–139758, Sep. 2019.

[7] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge intelligence and blockchain empowered 5G beyond for the industrial Internet of Things," *IEEE Netw.*, vol. 33, no. 5, pp. 12–19, Sep. 2019.

[8] A. Libri, A. Bartolini, and L. Benini, "PAElla: Edge AI-based real-time malware detection in data centers," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9589–9599, Oct. 2020.

[9] E. Peltonen *et al.*, "6G white paper on edge intelligence," Apr. 2020, *arXiv:2004.14850*. [Online]. Available: http://arxiv.org/abs/2004.14850

[10] Z. Wu, Y. Dong, W. Tian, and J. Jin, "Enhanced rough K-means based flow aggregation for QoS mapping in heterogeneous network environments," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 1197–1210, Jun. 2020.

[11] K. Wang, W. Zhou, and S. Mao, "On joint BBU/RRH resource allocation in heterogeneous cloud-RANs," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 749–759, Jun. 2017.

[12] M. Chen, O. Semiari, W. Saad, X. Liu, and C. Yin, "Federated echo state learning for minimizing breaks in presence in wireless virtual reality networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 177–191, Jan. 2020.

[13] L. Yao, A. Chen, J. Deng, J. Wang, and G. Wu, "A cooperative caching scheme based on mobility prediction in vehicular content centric networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5435–5444, Jun. 2018.

[14] A. T. Z. Kasgari, W. Saad, and M. Debbah, "Human-in-the-loop wireless communications: Machine learning and brain-aware resource management," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7727–7743, Nov. 2019.

[15] Z. Zhou, C. Gao, C. Xu, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Social big-data-based content dissemination in Internet of vehicles," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 768–777, Feb. 2018.

[16] Z. J. Wang, Y. N. Dong, and X. H. Wang, "A dynamic service class mapping scheme for different QoS domains using flow aggregation," *IEEE Syst. J.*, vol. 9, no. 4, pp. 1299–1310, Dec. 2015.

[17] M. Hijazi *et al.*, "Network traffic classification based on class weight based K-NN classifier (CWK-NN)," in *Proc. BDCSIntell*, Versailles, France, Dec. 2019, pp. 105–112.

[18] Q. Wu, Z. Du, P. Yang, Y.-D. Yao, and J. Wang, "Traffic-aware online network selection in heterogeneous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 1, pp. 381–397, Jan. 2016.

[19] L. He, D. Jiang, and C. Wei, "A QoE-based dynamic energy-efficient network selection algorithm," *Wireless Netw.*, Jan. 2020, doi: 10.1007/s11276-019-02231-z.

[20] A. Morell, A. Correa, M. Barceló, and J. L. Vicario, "Data aggregation and principal component analysis in WSNs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3908–3919, Jun. 2016.

[21] Q. Si, Z. Cheng, Y. Lin, L. Huang, and Y. Tang, "Network selection in heterogeneous vehicular network: A one-to-many matching approach," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, Antwerp, Belgium, May 2020, pp. 1–5.

[22] N. Ghosh, S. K. Ghosh, and S. K. Das, "SelCSP: A framework to facilitate selection of cloud service providers," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 66–79, Jan. 2015.

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.

[23] W. Gao, J. Cao, and Y. Xiong, "A novel QoS mapping mechanism in integrated UMTS/WLANs," *Wireless Pers. Commun.*, vol. 81, no. 3, pp. 1101–1116, Apr. 2015.

[24] A. M. Alkharasani, M. Othman, A. Abdullah, and K. Y. Lun, "An improved quality of service performance using RED's active queue management flow control in classifying networks," *IEEE Access*, vol. 5, pp. 24467–24478, 2017.

[25] M. Levorato, "Cognitive networking with dynamic traffic classification and QoS constraints," in *Proc. IEEE WCNC*, San Francisco, CA, USA, Mar. 2017, pp. 1–6.

[26] Y. C. Wang and S. Y. Hsieh, "Service-differentiated downlink flow scheduling to support QoS in long term evolution," *Comput. Netw.*, vol. 94, pp. 344–359, Nov. 2016.

[27] G. H. Von Wright, "The logic of preference reconsidered," *Theory Decis.*, vol. 3, no. 2, pp. 140–169, Dec. 1972.

[28] Q. Xu, J. Xiong, X. Cao, Q. Huang, and Y. Yao, "From social to individuals: A parsimonious path of multi-level models for crowdsourced preference aggregation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 844–856, Apr. 2019.

[29] K. Xiao, S. Mao, and J. K. Tugnait, "Robust QoE-driven DASH over OFDMA networks," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 474–486, Feb. 2020.

[30] J. Song, F. Yang, Y. Zhou, S. Wan, and H. R. Wu, "QoE evaluation of multimedia services based on audiovisual quality and user interest," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 444–457, Mar. 2016.

[31] L. Yang, Y. Dong, M. S. Rana, and Z. Wang, "Fine-grained video traffic classification based on QoE values," *Wireless Pers. Commun.*, vol. 103, no. 2, pp. 1481–1498, May 2018.

[32] M. Han, R. Zhang, T. Qiu, M. Xu, and W. Ren, "Multivariate chaotic time series prediction based on improved grey relational analysis," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 10, pp. 2144–2154, Oct. 2019.

[33] T.-C. Chang, "Detection of exercise fatigue using neural network with grey relational analysis from HRV signal," in *Proc. IEEE Int. Conf. Comput., Commun. Eng. (ICCCE)*, Fujian, China, Nov. 2019, pp. 87–90.

[34] L.-W. Hourng, C.-Y. Hsieh, and Z.-W. Fen, "The choice of optional working condition in electrochemical machining by grey relational analysis," in *Proc. IEEE Int. Conf. Appl. Syst. Invention (ICASI)*, Chiba, Japan, Apr. 2018, pp. 366–369.

[35] S. Kaci and L. V. Torre, "Algorithms for a nonmonotonic logic of preferences," *Comput. Sci.*, vol. 3571, pp. 281–292, Jul. 2005.

[36] H. Liao, X. Wu, X. Liang, J. Xu, and F. Herrera, "A new hesitant fuzzy linguistic ORESTE method for hybrid multicriteria decision making," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3793–3807, Dec. 2018.

[37] *NJUPT Dataset*. Accessed: Feb. 2019. [Online]. Available: https://pan.baidu.com/disk/home#/all?path=%2F&vmode=list

[38] *UNB ISCX VPN-nonVPN Traffic Dataset*. Accessed: Apr. 2018. [Online]. Available: http://www.unb.ca/cic/research/datasets/vpn.html

[39] J. Domzal, P. Jurkiewicz, P. Gawlowicz, and R. Wojcik, "Flow aggregation mechanism for flow-aware multi-topology adaptive routing," *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2582–2585, Dec. 2017.

[40] J. Jin, M. Palaniswami, D. Yuan, Y.-N. Dong, and K. Moessner, "Priority service provisioning and max–min fairness: A utility-based flow control approach," *J. Netw. Syst. Manage.*, vol. 25, no. 2, pp. 397–415, Oct. 2016.

[41] Z. Wu, Y. Dong, L. Yang, and P. Tang, "A new structure for Internet video traffic classification using machine learning," in *Proc. 4th IEEE CBD*, Lanzhou, China, Aug. 2018, pp. 322–327.

[42] Y. Purwanto, Kuspriyanto, Hendrawan, and B. Rahardjo, "Time based anomaly detection using residual polynomial fitting on aggregate traffic statistic," in *Proc. 2nd IEEE ICWT*, Manado, Indonesia, Apr. 2016, pp. 1–5.

[43] O. B. Rhaiem and L. C. Fourati, "QoS improvement for multicast video streaming: A new cross-layer scheme-based mapping, scheduling and dynamic arbitration algorithms for multicast scalable video coding (CMSAA-MSVC)," in *Proc. 4th IEEE WINCOM*, Marrakech, Morocco, Jan. 2016, pp. 1–6.

[44] H. H. Yang, Y. Wang, and T. Q. S. Quek, "Delay analysis of random scheduling and round robin in small cell networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 978–981, Dec. 2018.

Authorized licensed use limited to: Auburn University. Downloaded on September 10,2022 at 03:19:57 UTC from IEEE Xplore. Restrictions apply.