Optimal Coded Caching in 5G Information-Centric Device-to-Device Communications

Xingyan Chen*, Changqiao Xu*, Mu Wang*, Tengfei Cao*, Lujie Zhong† and Gabriel-Miro Muntean‡
*State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China
†Information Engineering College, Capital Normal University, Beijing 100089, China
‡ School of Electronic Engineering, Dublin City University , Dublin 9, Ireland
Email: {chenxingyan, cqxu, wangmu, caotf}@bupt.edu.cn, zljict@gmail.com, gabriel.muntean@dcu.ie

***Abstract*—As one of the key technologies for future 5G, Device-to-Device communications (D2D) offloads traffic to local by enabling mobile equipment directly communicating with each other, which perfectly supporting distributed applications and IoT scenarios. Integrating Information-centric networking (ICN) with D2D is becoming an attractive trend because of the superior advantages of inherent support of caching and name-based routing. Nevertheless, efficient caching in ICN D2D still remain problematic due to the low utilization of caching space and multicast feature of wireless scenarios. In this paper, we propose a novel optimal coded content caching mechanism for ICN-based 5G D2D. We first building a fluid-based model to describe how the roles of mobile nodes evolve with the user behaviors and caching strategy. We then accordingly formulate the coded caching problem as an optimization problem, which mainly considers the tradeoff between delivery latency and energy consumption. The existence of optimal solutions is proved theoretically. We further propose a** $Learn\ Tree\text{-}based\ Code\ Content$ **(**$LTCC$**) mechanism to cluster the contents for content coding selection and an** $Optimal\ Coded\ Content\ Caching$ **(**$O3C$**) algorithm to solve coded content caching problem. Finally, we conduct massive simulation tests to validate the performance of the proposed algorithm against the state-of-art solutions.**

I. Introduction

With the fast development of wireless communication technologies which have instinct the explosive of smart mobile devices and mobile applications, recent mobile data networks have experienced drastic growth. According to Cisco VNI prediction [1], global mobile video traffic reached 4.2 exabytes per month at the end of 2016, and will increase by 8 times between 2016 and 2021. Such a large amount of traffic presents a dual challenge to core network load and spectrum resource utilization. Besides, ever increasing mobile data traffic has brought huge pressure to the network infrastructure. In this context, the design of the future fifth generation (5G) network [2] must be able to accommodate the overwhelming demand for mobile communications. Device-to-Device communication (D2D) offloads the traffic pressures to local by enabling users to communicate each without the coordinate of Base Stations [3], which increases the network capacity and alleviate the pressures of backhaul. Hence, D2D has been treated as one of the key technologies in future 5G. However, introducing mobile service into such scenarios still remain problematic due to the mismatch between host-entric design of conventional IP architecture and content-centric requirement of mobile service [4]. By naming the content instead of host in network, information-centric networking (ICN) shifts the network design focus from conventional host communication to content distribution, which inherently supports ubiquitous caching and multicast delivery. Such salient features of ICN make it a very promising solution for improving mobile service over 5G D2D communications.

In-networking is a key feature of ICN which attempts to alleviate the traffic pressure of core network and reduce the data transmission delay. Thus, it critical to study how to improve the caching performance for ICN. However, conventional uncoded caching schemes [5][6] in ICN D2D cache the content independently, namely, requests arrived simultaneously can only be served by unicast transmission separately. Due to the failure of utilizing multicasting feature of wireless channel, network capacity of such solutions is limited.

Different from uncoded solutions, coded caching scheme [7] synchronously satisfies the different requests by coding asked contents together and delivering them via broadcast channel. The gain of multicasting opportunity by coding content not only improves the transmission capacity but also reduces the delivery latency. Hence, coding caching has become an attractive trend for caching problem in ICN D2D [8][9]. Although, caching coded content in wireless scenarios has been studied by several literatures, problem still remains. Content in [7] will not be encoded until being transmitted, which still suffers from the low caching utilization due to the tremendous caching redundancy. Literature [10] treats each content equally when encoding them together. The feature of neglecting the content demand triggers the imbalance between demand and supply.

In this paper, we propose an optimal coded caching scheme for ICN-based 5G D2D. We first propose a fluid-based model to describe the dynamic evolution of the network. Based on the models, we formulate the coded caching problem as an optimization problem which balances the tradeoff between delivery latency and energy consumption. An $Optimal\ Coded\ Content\ Caching$ ($O3C$) algorithm to solve the problem of joint optimization of delay and energy consumption. We also conduct experiments to verify the accuracy of the model, and performance of our algorithm. The main contributions of this paper are:

978-1-5386-4727-1/18/$31.00 ©2018 IEEE

1) We consider ICN-based 5G D2D network as a dynamical system and build a fluid-based model to describe the mobile node role evolving with the network system status(the arrival rate of content requests and coded content cache strategy). We also verified the accuracy of the model theoretical results by experiments.
2) Based on the proposed model, we first give the object function of joint optimization of delay and consumption problem, and design the $LTCC$ mechanism to accomplish the coded content selection. Then, we propose the $O3C$ algorithm to solve optimal coded caching problem.
3) We design a series of simulations to compare our algorithm with the state-of-art uncoded content caching approaches. Results demonstrate our algorithm outperforms state-of-the-art solutions [11][12] in terms of average number of hops, latency and energy consumption.

II. System Model

In this section, we will first give some reasonable assumptions, and then introduce a fluid-based model to describe the dynamics of ICN-based 5G D2D network with coded content caching.

A. Assumptions

Before introducing our model, we make the following assumptions.

Similar as our previous work in [12][13] where a Named Data Networking [14] is employed for future 5G-D2D scenarios[1]. Each ICN node is equipped with 5G-D2D interface, which makes it possible to build a pure ICN-based 5G D2D network. According to [12], such mobile node will play any of the three roles: content $consumer$, $producer$ and $forwarder$, each role follows the same definition. Besides, we also assume the contents are divided into multiple chunks with equal size. The requesting rate $q(i)$ of each chunk follows the Zipf distribution[15] with parameter ρ. The Random Way Point (RWP) model as described in [16] is employed for mobility description.

B. Coded File Reconstruction in Mobile ICN

For capacity analysis, the total m contents represented by a set $\boldsymbol{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_m\}$, where each content contains k number of chunks as $\boldsymbol{f}_i = \{c_i^1, c_i^2, ..., c_i^k\}$. We use $\boldsymbol{L} = \{\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_{K_c}\}$ to denote the set of classes and $\boldsymbol{l}_j = \{l_j^1, l_j^2, ..., l_j^{|\boldsymbol{l}_j|} | l_j \in \boldsymbol{f}\}$ to record the set of chunks in class j. According to [10], for each class, randomly selecting each one of the chunks from the set $\boldsymbol{l}$ with probability $\frac{1}{2}$ and then mix the selected chunks via the XOR operation to create one encode file. The i-th encoded cached file of class j can be represented as:

$$g_j^i = \sum_{n=1}^{|\boldsymbol{l}_j|} a_n^{ij} l_n = \boldsymbol{a}_j^i \boldsymbol{l}_j, \qquad (1)$$

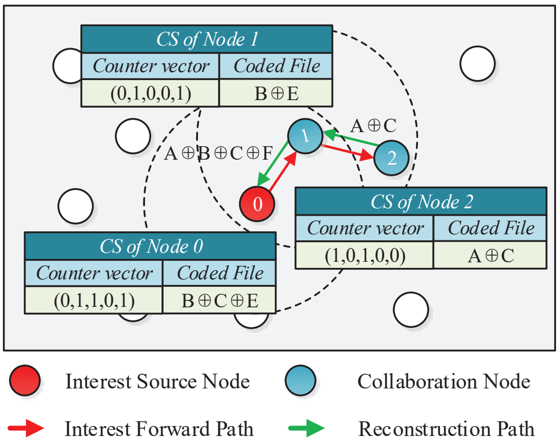

Fig. 1. An example of decoding in ICN-based 5G D2D network

where $\boldsymbol{a} = [a_1, a_2, .., a_{|\boldsymbol{l}_j|}]$ is an uniformly distributed vector of encoding counters with binary elements and the summation is carried over GF(2). In this way, if the mobile nodes find a set of nodes in the forward path with enough encoded files containing $|\boldsymbol{l}_j|$ linearly independent encoded vectors, they can collaborate to decode the files. As Fig. 1 shows, considering the content set of current network is $\{A, B, C, D, E\}$, we assuming that node n_0 contains the coded content $\{B \oplus C \oplus E\}$. If n_0 requests content A, it will need uncoded content packet A or $\{A \oplus B \oplus C \oplus E\}$ for decoding A. Thus, n_0 sends out the interest packet containing the $Counter\ Vector$ $(1,0,0,0,0)$ and $(1,1,1,0,1)$ to the neighbor n_1. The receiving node n_1 does not contain the asked packets. Since n_1 contains $\{B \oplus E\}$, decoding the requested packet will only need any one of the packets A, $\{A \oplus B \oplus C \oplus E\}$, $\{A \oplus B \oplus E\}$ and $\{A \oplus C\}$. Thus, n_1 forwards the request with vectors $(1,0,0,0,0)$, $(1,1,1,0,1)$, $(1,1,0,0,1)$ and $(1,0,1,0,0)$ to next hop n_2 which has coding packet $\{A \oplus C\}$. n_2 passes the $\{A \oplus C\}$ to n_1, and n_1 obtains the $\{A \oplus B \oplus C \oplus E\}$ by combining the $\{A \oplus C\}$ and $\{B \oplus E\}$. Then, n_1 returns the asked $\{A \oplus B \oplus C \oplus E\}$ to n_0 and n_0 decodes the A with $\{A \oplus B \oplus C \oplus E\}$ and $\{B \oplus C \oplus E\}$.

Based on above discussion, each mobile node can decodes the k-th content via N linear independent packets, which can be represented by $\left(\sum_{i=1}^{R}\sum_{j=1}^{N} b_j^i \boldsymbol{a}_j^i\right)\boldsymbol{l_j}$, where $b_j^i \in GF(2)$ denotes the XOR operation over the reconstruction path, and we have vector $\boldsymbol{v}_k$ denoted by $\sum_{i=1}^{R}\sum_{j=1}^{N} b_j^i \boldsymbol{a}_j^i = \boldsymbol{v_k}$. Intuitively, for k-th component in $\boldsymbol{v}_k$ and 0 for other components.

C. Fluid-based Model

We discuss the basic concepts of the fluid-based model for a given class j in ICN-based 5G D2D. Inspired by our early work [12], we firstly define four roles for mobile nodes: **Consumer**, the node which issues an $Interest$ packet for specific encode chunk in class j; **Relay**, the intermediate node which receives, extends and forwards the $Interest$ packet since it does not contain them in its local CS ; **Provider**, the node which holds one of the demand chunks; **Ordinary**, the node which does not belong to any role above. To further describe, we introduce the following four bits:

[1] Without loss of generality, we use NDN as an example. Since content caching is a common feature of any ICN architecture, our solution can be also extend to ICNs other than NDN

- **Request bit** ($\mathcal{R}$): 1 if the node is a consumer for the chunk, 0 otherwise.
- **Forward bit** ($\mathcal{F}$): 1 if the node is a relay for the chunk, 0 otherwise.
- **Spread bit** ($\mathcal{S}$): 1 if the node is able to spread the request to neighbor node, 0 otherwise.
- **Have bit** ($\mathcal{H}$): 1 if the node is capable of decoding the corresponding content, 0 otherwise.

Unlike the [12] which consider the non-coded scenarios, we in this paper model 6 possible node states during the content sharing and Each state can be described as follows:

- **Ordinary state $\boldsymbol{A}$** (R=0, F=0, S=0, H=0): The node in this state is an ordinary node for specific encode chunk, let $A(t)$ be the population fraction of nodes in this state at time t.
- **Activated consumer state $\boldsymbol{D}$** ($\mathcal{R}$=1, $\mathcal{F}$=0, $\mathcal{S}$=1, $\mathcal{H}$=0): This state indicates that the requesting node is sending out the request, Let $D(t)$ denotes the population fraction of nodes in state $\boldsymbol{D}$ at time t.
- **Inactivated consumer state $\boldsymbol{B}$** ($\mathcal{R}$=1, $\mathcal{F}$=0, $\mathcal{S}$=0, $\mathcal{H}$=0): The node in this state is a requester that have already forwarded the request and is waiting for the finish of decoding over the forwarding path. Let $B(t)$ be node scale in state $\boldsymbol{B}$ at time t.
- **Consumer satisfied state $\boldsymbol{X}$** ($\mathcal{R}$=1, $\mathcal{F}$=0, $\mathcal{S}$=0, $\mathcal{H}$=1): In this state, the consumers have already obtained the enough packets for decoding the demand chunk. Denote $X(t)$ scale of nodes in $\boldsymbol{X}$ at time t.
- **Activated relay state $\boldsymbol{D'}$**($\mathcal{R}$=0, $\mathcal{F}$=1, $\mathcal{S}$=1, $\mathcal{H}$=0): The node enters in this state when it receives a request and prepares to forward it. This state node cannot decode the requested content and is preparing to forward the request to next hop. Given $D'(t)$ as the population fraction in $\boldsymbol{D'}$ at time t.
- **Inactivated relay state $\boldsymbol{B'}$** ($\mathcal{R}$=0, $\mathcal{F}$=1, $\mathcal{S}$=0, $\mathcal{H}$=0): The nodes in this state are the relay nodes that have already forwarded the request and waited for any of the requested coded content back. We define $B'(t)$as the population fraction of $\boldsymbol{B'}$at time t.

Each node in networks is one of the six states, hence the sum of all nodes of each state is constantly equal to the total number of nodes in networks, namely:

$$A(t) + D(t) + B(t) + X(t) + D'(t) + B'(t) = 1 \quad (2)$$

Further, we build a fluid-based model to describe the dynamics of $A_j(t)$, $D_j(t)$, $B_j(t)$, $X_j(t)$, $D'_j(t)$, $B'_j(t)$ in class j. The transition of above 6 states and 8 possible types of state transitions (represented by the dotted/solid line with arrow) can be interpreted as Fig. 2. The details of each transition are shown below:

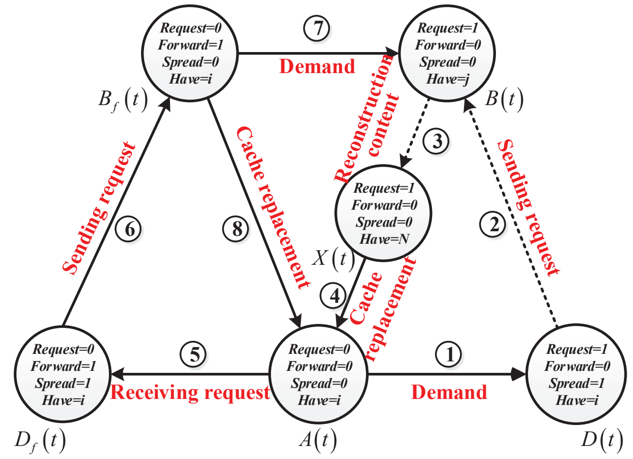

Fig. 2. Possible state transitions of nodes in ICN-based 5G D2D network with coded content caching

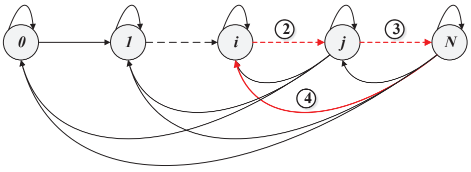

Fig. 3. The state space of the Markov chain for decoding

- **Transition 1**: This transition follows the same definition in [12], which indicates that the node in state $\boldsymbol{A}_j$ becomes interested in l_j^i a chunk i in class j. The probability of each node becomes interested in l_j^i follows the Poisson distribution with parameter β_j^i. The conversion rate of transition 1 can be denoted by $\beta_j A_j(t) = \sum_{i=1}^{|\boldsymbol{l}_i|} \beta_j^i A_j(t)$. When the Δt is small enough, we can assume that $\beta_j \Delta t \approx \beta_j dt$.
- **Transition 2**: This transition indicates that activated consumer converts from state $\boldsymbol{D}_j$ to inactivated state $\boldsymbol{B}_j$ after sending out the request. Since all activated consumers are converted to inactivated consumers after forwarding the $interest$ packet, the conversion rate is $D_j(t)$.
- **Transition 3**: In coded caching case where enough encode packets are received for decoding l_j^i, inactivated consumers in state $\boldsymbol{B}_j$ will become satisfied consumers $\boldsymbol{X}_j$. As mentioned earlier, the consumers get the demand chunk by spanning the entire message space during reconstruction path. In this way, the state transition here can be seen as the Markov chain shown in fig. 3. According to the [10], when the consumer has h linearly independent encoded chunks, the consumer will needs to find $|\boldsymbol{l}_i| + \gamma - h$, ($\gamma = \sum_{i=1}^{|\boldsymbol{l}_i|} \frac{1}{2^i - 1} \approx 1.6067$) new encoded chunks to decode the required chunk on average. When the mobile nodes have cached an average of h encoded chunks that are linearly independent, it needs $\mathbb{H} = \left\lceil \frac{|\boldsymbol{l}_i| + \gamma - h}{h} \right\rceil$ hops (cooperation nodes) to decode the content on average. Here, we believe that request will be spread once at an interval Δt , and the average conversion rate from the state $\boldsymbol{B}_j$ to $\boldsymbol{X}_j$ is $B_j(t)/\mathbb{H}$.
- **Transition 4**: After submitting content to upperlayer, satisfied consumers in state $\boldsymbol{X}_j$ will become ordinary state by converting from state $\boldsymbol{X}_j$ to $\boldsymbol{A}_j$. Unlike uncoded caching replacement the whole chunk, mobile nodes need to modify the encoded chunks according to the coding rules and caching strategy. Redundant coded chunks against the encoding rules and caching strategy will be replace by suitable one. Since the XOR coded is flexible and the popularity varies slowly. Thus, the encoded content caching can self-adapt to the dynamic environment without requiring specific

caching replacement. The conversion rate is $X_j(t)$.

- **Transition 5**: When an ordinary node receives an $Interest$ packet, it will become to an activated relay, namely conversion from state $\boldsymbol{A}_j$ to $\boldsymbol{D}'_j$. The forwarding of interest packets can be regarded as an epidemic process(EP) [17][18] since the activated relays' mission is transforming one of their neighbors into an activated relay. As the unicast-based forwarding is considered, the conversion rate of this transition can be represented by the following equation according to[17]:

$$A_j(t)(D_j(t) + D'_j(t)) \tag{3}$$

- **Transition 6**:After sending out the $interest$ packet of chunk l_j^i, an activated relay is converted from state $\boldsymbol{D}'_j$ to inactivated state $\boldsymbol{B}'_j$, and waiting for the encoded chunk to return. Similarly to transition 2, all activated relays will convert to inactivated relays after spreading the $interest$ packet and hence the conversion rate is $D'_j(t)$.
- **Transition 7**: Since the relay nodes in state $\boldsymbol{B}'_j$ may become interest in l_j^i as same as the ordinary nodes. They may convert to $\boldsymbol{B}_j$. Similarly to transition 1, the conversion rate of transaction 7 is equal to $\beta_j B'_j(t) = \sum_{i=1}^{|\boldsymbol{l}_i|} \beta_j^i B'_j(t)$.
- **Transition 8**: When a relay receives the demand coded chunks, it needs to adjust the coded content cache based on network coding rules and caching strategy. Replace the redundant coded chunks that do not conform to the encoding rules and caching strategy with new encoded chunks, namely conversion from state $\boldsymbol{B}'_j$ to $\boldsymbol{A}_j$. The average conversion rate of this transition is $2B'_j(t)/\mathbb{H}$. Because the first relay needs average $\mathbb{H} - 1$ hops to obtain the demand chunks on the reconstruction path. The second relay needs $\mathbb{H} - 2$. By that analogy, the i-th relay needs $\mathbb{H} - i$, $i < \mathbb{H}$. The average hop of all relays is $\mathbb{H}/2$.

The conversion rate of above 8 transition process is summarized in table 1. According to fig. 2 and table 1, we can obtain the following O.D.E functions:

$$\begin{aligned}
\dot{A}_j &= X_j(t) + 2B'_j(t)/\mathbb{H} - \beta_j A_j(t) \\
&\quad - A_j(t)(D_j(t) + D'_j(t)) && (4) \\
\dot{D}_j &= \beta_j A_j(t) - D_j(t) && (5) \\
\dot{B}_j &= D_j(t) - B_j(t)/\mathbb{H} + \beta_j B'_j(t) && (6) \\
\dot{X}_j &= B_j(t)/\mathbb{H} - X_j(t) && (7) \\
\dot{D'}_j &= A_j(t)(D_j(t) + D'_j(t)) - D'_j(t) && (8) \\
\dot{B'}_j &= D'_j(t) - \beta_j B'_j(t) - 2B'_j(t)/\mathbb{H} && (9) \\
\boldsymbol{U}_j|_{t=t_0} &= U_{j,t_0} && (10)
\end{aligned}$$

Where $\boldsymbol{U}_j$ represents the system state of the class j. Given the initial value U_{j,t_0} by

$$U_{j,t_0} = (A_j(t_0), D_j(t_0), B_j(t_0), X_j(t_0), D'_j(t_0), B'_j(t_0)).$$

D. Accuracy of the O.D.E Approximation

To validate the accuracy of proposed O.D.E function, we use the ndnSIM based on NS3 to simulate the real NDN environments and conduct a series of tests. Specifically, we build a NDN network with 1000 mobile nodes moving arbitrarily over a 3000 × 3000. The RWP mobility model is used and its velocity range is set to $[3, 10]\, m/s$. Consider 10 different videos of which are 10s long and consists of 5 chunks. All chunks are encoded randomly. The caching capacity of each mobile node in NDN is set to 5 chunks large. To eliminate the randomness, we repeat 20 times with different random seeds and take the average of the results. Fig.4 shows the comparison between theoretical and experimental values of the system evolution. As figure shows, our model converges well to the simulation results. Hence, we can conclude that our model indeed describe the realistic system evolving.

TABLE I
STATE UPDATE AND CONVERSION RATE OF TRANSITIONS

Transition	State update	Conversion rate
1	(0,0,0,0) → (1,0,1,0)	$\beta_j A_j(t)$
2	(1,0,1,0) → (1,0,0,0)	$D_j(t)$
3	(1,0,0,0) → (1,0,0,1)	$B_j(t)/\mathbb{H}$
4	(1,0,0,1) → (0,0,0,0)	$X_j(t)$
5	(0,0,0,0) → (0,1,1,0)	$A_j(t)(D_j(t) + D'_j(t))$
6	(0,1,1,0) → (0,1,0,0)	$D'_j(t)$
7	(0,1,0,0) → (1,0,0,0)	$\beta_j B'_j(t)$
8	(0,1,0,0) → (0,0,0,0)	$2B'_j(t)/\mathbb{H}$

III. CACHING OPTIMIZATION

In this section, we will introduce the optimize caching policy based on the fluid-based model.

A. Problem Formulation

Unlike traditional uncoded caching mechanisms storing continuous chunks, coded caching decouples the correlations between contents and divides the content into different categories. Contents in each category are stored in the form of linear independent packets.

Instead of placing content as conventional noncoded solutions, caching how many linear independent packets for each category should be considered. Recall that the request distribution is Zipf-like, each mobile node in network can be equally considered since they follow the same requesting distribution. For each mobile node i in D2D scenarios, let $\eta = \ e_1^i, e_2^i, \ldots, e_j^i, \ldots, e_{K_c}^i$ caching policy, where e_j indicates the number of coding packets for j-th category. The total cached content M_i for i equals to $\sum_{j=1}^{K_c} e_j^i$.

According to the discussion in the previous section, the more linear independent packets be cached in each node, the less hop and delay of content retrieve. However, due to the constraints of caching space and available energy, it is impossible to infinitely cache the packets. Thus, the objective function for caching optimization problem for coded caching can be formulated as the tradeoff between waiting delay and energy cost(caching and forwarding), which by :

$$J_\eta = \sum_{j=1}^{K_c} J_\eta^j = \sum_{j=1}^{K_c} (\phi B_{j,\eta}(T_{j,\eta}) + \varphi B_{j,\eta}(T_{j,\eta}) + \psi e_j) \tag{11}$$

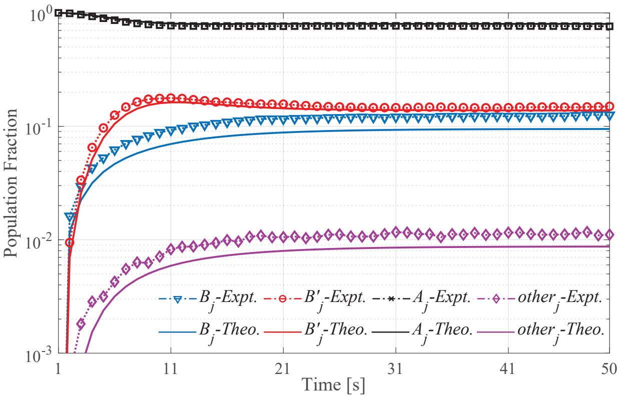

Fig. 4. Population Fraction vs. Time

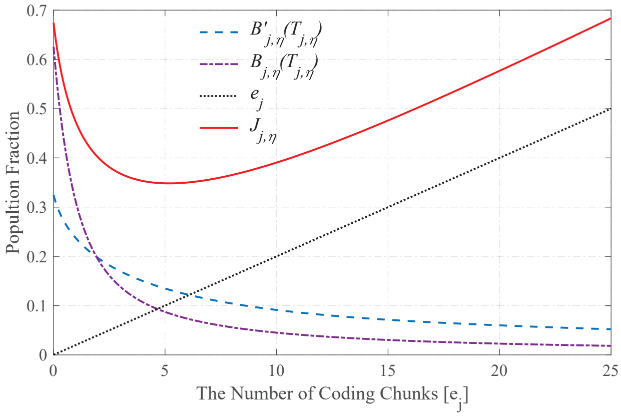

Fig. 5. Population Fraction vs. the number of coding chunks e_j

Where $B_{j,\eta}\left(j,\eta\right)$ indicates that the value of $B_j\left(t\right)$ under the caching operation η and initial condition U when time is $T_{j,\eta}$. Let $T_{j,\eta}$ be the time when $B_{j,\eta}\left(t\right)$ reaches the peak value, namely, $\{T_{j,\eta}|B_{j,\eta}\left(T_{j,\eta}\right) = \max_T B_{j,\eta}\}$. Similarity, we have $T_{j,\eta}$ with $\{T_{j,\eta}|B_{j,\eta}\left(T_{j,\eta}\right) = \max_T B_{j,\eta}\}$. e_j is the total scale of coded content in category j. α, β and γ are the weight parameters where $\phi+\varphi+\psi = 1$. Thus, we can formulate the optimization as follows.

$$\mathbf{min} \quad J_\eta \tag{12}$$

$$\boldsymbol{s.t} \quad \sum_{j=1}^{K_c} e_j^i < C \tag{13}$$

Where C indicates the caching capacity limits.

B. Optimal Control

In this subsection, we will discuss how to optimize the caching control parameter η to minimize the objective J_η. Due to the J_η is separable at caching category, we consider for each category j, we have the J_η^j optimization objective.

$$J_\eta^j = \phi B_{j,\eta}(T_{j,\eta}) + \varphi B_{j,\eta}(T_{j,\eta}) + \psi e_j \tag{14}$$

Since $B_{j,\eta}(T_{j,\eta})$ and $B_{j,\eta}(T_{j,\eta})$ are continuous with the respect to e_j, then we can have an optimal e_j that minimize the J_η^j.

$Proof$: To simplify the description of O.D.E function(4)-(10), we rephrase the population fraction of each state with corresponding lowercase letter, such as, b for $B(t)$, $\dot{b}$ for $\dot{B}(t)$.

$$\begin{aligned} \dot{a}_j &= x_j(t) + \alpha_j b'_j(t) - \beta_j a_j(t) \\ &\quad - a_j(t)(d_j(t) + d'_j(t)) \end{aligned} \tag{15}$$

$$\dot{d}_j = \beta_j a_j(t) - d_j(t) \tag{16}$$

$$\dot{b}_j = d_j(t) - \gamma_j b_j(t) + \beta_j b'_j(t) \tag{17}$$

$$\dot{x}_j = \gamma_j b_j(t) - x_j(t) \tag{18}$$

$$\dot{d'}_j = a_j(t)(d_j(t) + d'_j(t)) - d'_j(t) \tag{19}$$

$$\dot{b'}_j = d'_j(t) - \beta_j b'_j(t) - \alpha_j b'_j(t) \tag{20}$$

where $\alpha_j = \frac{(\mathbb{H}-1)}{\sum_{i=1}^{\mathbb{H}-1} i}$ and $\gamma_j = \frac{1}{\mathbb{H}}$. Let $\boldsymbol{f}_j$ denotes as $(f_{a_j}, f_{d_j}, f_{b_j}, f_{x_j}, f_{d'_j}, f_{b'_j})$, which are the right side of the (15)-(20). According to (3), we have $(a_j, d_j, b_j, x_j, d'_j, b'_j) \in [0,1]^6$. Therefore the Jacobian of $\boldsymbol{f}_j$ is bounded and satisfy the Lipchitz condition.

According to (14), we can limit our focus on state evaluation of $(b_j(t)$,$b'_j(t))$

$$\dot{b}_j = d_j(t) - \gamma_j b_j(t) + \beta_j b'_j(t) \tag{21}$$

$$\dot{b'}_j = d'_j(t) - \beta_j b'_j(t) - \alpha_j b'_j(t) \tag{22}$$

where β_j is constant and only related to the popularity of $\boldsymbol{l}_j$. As we can see, $b_j(t)$ and $b'_j(t)$ are the monotonically decreasing in terms of γ_j and α_j respectively. Hence, we know that $b_j(t)$ and $b'_j(t)$ are monotonically decreasing with the respect to e_j.

Assume $\boldsymbol{\omega}_j^{(1)}(t) = (b_j^{(1)}(t)$,$b_j'^{(1)}(t))$ and $\boldsymbol{\omega}_j^{(2)}(t) = (b_j^{(2)}(t)$,$b_j'^{(2)}(t))$ are the trajectories of equation (21) and (22) under different caching policies η_1 with $e_j^{(1)}$ and η_2 with $e_j^{(2)}$ respectively. With loss of generality, we assume $e_j^{(1)} > e_j^{(2)}$, and denote $\Delta\gamma_j$ =$\gamma_j^{(1)} - \gamma_j^{(2)}$ and $\Delta\alpha_j$ =$\alpha_j^{(1)} - \alpha_j^{(2)}$. According to (14), we have following inequality

$$||\boldsymbol{\omega}_j^{(1)}(T_{j,\eta}) - \boldsymbol{\omega}_j^{(2)}(T_{j,\eta})||_2 \leq CT_{j,\eta}\sqrt{\Delta\gamma_j^2 + \Delta\alpha_j^2}$$

Where $C = \sqrt{2}(2+\beta_j)$. Since $\boldsymbol{f}_j$ is bounded, and $T_{j,\eta}$ is finite, we thereby have the continuous of $b_j\left(T_{j,\eta}\right)$ and $b_j\left(T_{j,\eta}\right)$ with e_j, where γ and α are continuous for e_j.

Fig. 5 shows how $B'_{j,\eta}(T_{j,\eta})$, $B_{j,\eta}(T_{j,\eta})$ and J_η^j vary with e_j, where ϕ=0.1, φ=0.5 and ψ=0.4. The setting of experimental parameters is the same as in Section 2.E. We can see $B'_{j,\eta}(T_{j,\eta})$ and $B_{j,\eta}(T_{j,\eta})$ are the monotone decreasing with e_j, which also proves the previous conclusion. This is a good explanation for the fact that the more cached coded chunks are in the mobile nodes, the less consumers and forwarders in the network. For the value of cost function J_η^j, we can see the corresponding curve firstly experience a decrease trend and then increase with the e_j. This is because as the cache fraction increases, the cost of network cache increases gradually, resulting in the overall cost increase. Besides, we can find the minimum value of J_η^j through numerical calculation.

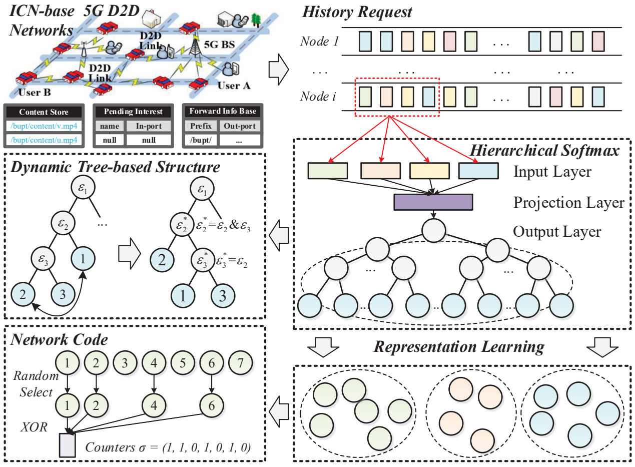

Fig. 6. Architecture of the $Learn\ Tree\text{-}based\ Code\ Content$ Mechanism

C. Practical Algorithm

In the following part, we will introduce the $LTCC$ mechanism based on hierarchical softmax structure and the design of word2vec [19]. This famous natural language processing (NLP) structure has been widely used in cluster analysis in many fields [20][21]. Hierarchical softmax-based model is the core of $LTCC$ which consists of three layers, input layer, projection layer and output layer. And the design of the model is based on CBOW and Skip-gram in word2vec. The principle is to calculate the next-word's probability in a given context, which is derived as

$$P(w|context) = \prod_{x=1}^{n} P(b = b_j(w)|l_x(w), context) \tag{23}$$

where w is the target word, and the right-hand side represents the formula for calculating the probability through the Hierarchical softmax. Due to the limited space, we omit the introduction that can find in [20]. In order to adapt to content coding selection, we improve the original hierarchical softmax by using dynamic Tree-based structure to calculate the conditional probability requesting event under the condition that another content is reqeusted at set time interval T_{ir}. Once the representations are learned by hierarchical softmax, we use k-means algorithm to divide the contents into K_c classes.

As aforementioned, each mobile node periodically uploads users' request records to cloud. The cloud proxy maintains the request history of each node and sort them by time. Further, the request history set is divided into a series of parts with the maximum request interval T_{ir} between the current records and predicted record within request history. Therefore, the cloud server can calculate the optimal cache policy according to network state $\boldsymbol{U}_j$. After obtaining the optimal η and coding rule, the cloud server broadcasts them to all nodes through the base station, and then the mobile nodes perform cache replacement and recoding based on η and coding rule. Meanwhile, the mobile nodes upload their own status periodically. The above process is implemented by the pseudo code of **Algorithm 1**.

Algorithm 1 $O3C$ – **Coded Content Caching Policy**

```
Cloud Server side:
  for each time-slot $T_{ir}$:
    collect network status $\boldsymbol{U}_j$ and mobile nodes request records
    for each class $j$:
      according to request history of mobile nodes
      using the $LTCC$ to calculate the coding rule;
      calculate the optimal cache policy $e_j$ with $\boldsymbol{U}_j$.
    end for
  end for

Mobile Nodes side:
  for each time-slot $T_{ir}$:
    upload $\boldsymbol{U}_j$ and request records
    for each class $j$:
      recoding based on the new coding rule ;
      caching replacement based on the cache policy $e_j$.
    end for
  end for
```

IV. Performance Evaluation

In this section, we design a series of experiments to compare the performance of our algorithm with two state-of-art ICN D2D-based caching strategies, ς^*-OCP [12], GrIMS [11]. The parameters setting of RWP model are the same as before. We deployed a 2000*2000 m^2 quadrate simulation scenario with 200 mobile nodes, and set the simulation time to 1000s. Besides, we adopt 1000 different videos and each of the videos contain 10 chunks. The length of each chunk is 2s. The user's request for content follows the Zipf distribution with parameter ρ = 0.8. The caching size of each node is set to 1×10^5 MTUs, where each MTU equals to 1500B. In such cases, mobile node can store 300 content chunks at most. We employ the $LTCC$ to cluster the content into 100 categories, and use the central controller such as BSs to broadcast the categories and coding strategy η to all mobile nodes. We also deploy two mobile nodes as the initial content providers which disseminate the original content in network.

A. Simulation Tests

For each caching strategy, three caching performance factors are tested. **Average hit hop** (**AHH**): Traditionally, AHH indicates how many hops have been passed when request is satisfied. For coded caching, the AHH is defined by the average hop of finding all packets for decoding. **Average downloading time** (**ADT**): ADL represents the average time interval between sending out the request and obtaining the enough coding packets. **Energy Consumption** (**EC**): **EC** indicates the total energy consumption of all forwarding and caching operations during the simulation.

According to fig. 6, we can see that $O3C$ works best on **AHH**, comparing to the other two algorithms, and the **AHH** decreases by about 2 hops. The superiority of $O3C$ is because the coded content caching reduces the likelihood of redundant replica. Intuitively, assuming that there are five different content chunks in the network and each node can cache one chunk. In the uncoded caching, the probability of two nodes to caches the same content is $1/5$. However, in the coded content caching, there are 2^5 linearly independent coded contents, the probability is 1/2^5.

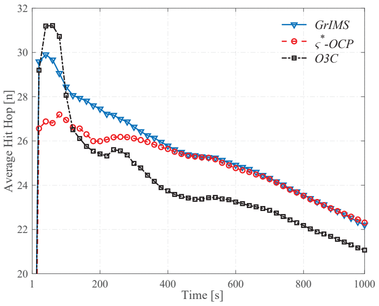

Fig. 7. Average Hit Hop vs. Time

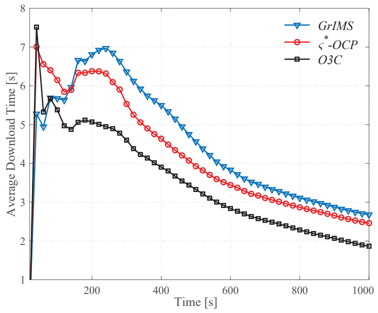

Fig. 8. Average Download Time vs. Time

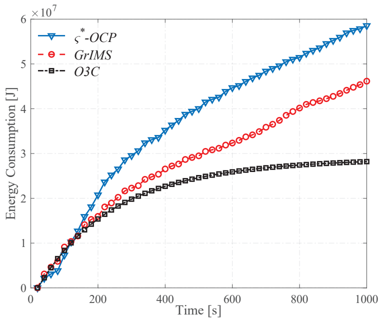

Fig. 9. Energy Consumption vs. Time

Fig. 7 shows how the **ADL** of three caching solutions varying with the simulation time. All curves begin with a brief period of shock and then slow decreases over time. This is because the node movement causes the system to be unstable at the beginning. As the time pass by, the cache space is gradually being optimized for full utilization and cache configuration, which reduces the **ADT**. From Fig. 7 we can also see $O3C$ has the lowest **ADT** among three solutions.

In addition, fig. 8 shows the **EC** of three solutions. Because the decrease in **AHH** reduces the energy consumed by forwarding, we can see $O3C$ outperforms than ς^*-OCP and GrIMS at 1000s by about 24% and 47%. The results show that compared with the traditional caching strategy, the cache space can be more effectively utilized by caching encoded content.

V. Conclusion and Future Work

This paper proposed a coded caching scheme for joint optimization of delivery latency and energy consumption. We first model the network status evolving with user behaviors and caching operations as a fluid-based model. We formulate the optimization problem for coded caching based on this fluid model, focusing on the tradeoff between delivery latency and caching cost. The existence of optimal solutions is proved. To practically solve this problem, a *Learn Tree-based Coded Content* (*LTCC*) mechanism is designed in order to cluster the contents in to different categories, and an *Optimal Coded Content Caching* (*O3C*) algorithm is also proposed for optimally scheduling the coded caching chunks over all mobile nodes. A series of simulation results have also been conduct and prove the dominance of our algorithm. Future work will include how to design the coded-oriented caching replacement and scenarios with time varying of user behaviors.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61522103;

References

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast*, Cisco, San Jose, CA, USA, 2017.

[2] J. Qiao, Y. He and X. S. Shen, "Proactive Caching for Mobile Video Streaming in Millimeter Wave 5G Networks," *IEEE Trans. on Wireless Commun.*, vol.15, no.10, pp.7187-7198, Oct. 2016.

[3] A. Moubayed, A. Shami and H. Lutfiyya, "Power-Aware Wireless Virtualized Resource Allocation with D2D Communication Underlaying LTE Network," *in Proc. IEEE Global Commun. Conf. (GLOBECOM)*, pp.1-6, 2016.

[4] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher and B. Ohlman, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no.7, pp.26-36, July 2012.

[5] G. Deng and L. Wang et al. "Distributed Probabilistic Caching strategy in VANETs through Named Data Networking," *in Proc. IEEE Comput. Commun. Soc. Workshops (INFOCOM WKSHPS)*, pp.314-319, 2016.

[6] C .Xu, W. Quan, V. A. Vasilakos, and H. Zhang, "Informationcentric cost-efficient optimization for multimedia content delivery in mobile vehicular networks," *Comput. Commun.*, vol.99, no.1, pp.93-106, Feb. 2017.

[7] Y. Fadlallah, A. M. Tulino, D. Barone, G. Vettigli, J. Llorca and J. M. Gorce, "Coding for Caching in 5G Networks," *in Proc. IEEE Commun. Magazine*, vol.55, no.2, pp.106-113, Feb. 2017.

[8] M. K. Kiskani and H. Sadjadpour, "Application of index coding in information-centric networks," *in Proc. Inter. Conf. on Comput., Net. and Commun. (ICNC)*, pp.977-983, 2015.

[9] J. Saltarin, E. Bourtsoulatze, N. Thomos and T. Braun, "NetCodCCN: A network coding approach for content-centric networks," *in Proc. IEEE Comput. Commun. Soc. (INFOCOM)*, pp.1-9, 2016.

[10] M. K. Kiskani and H. R. Sadjadpour, "Throughput Analysis of Decentralized Coded Content Caching in Cellular Networks,"*IEEE Trans. on Wireless Commun.*, vol.16, no.1, pp.663-672, Jan. 2017.

[11] C. Xu, W. Quan, H. Zhang, and L. A. Grieco, "GrIMS: Green Information-Centric Multimedia Streaming Framework in Vehicular Ad Hoc Networks," *IEEE Trans. on Cir. & Sys. for Vid. Technol.*, vol.28, no.2, pp.483-498, Sept. 2016.

[12] C. Xu, M. Wang, X. Chen, L. Zhong and A. L. Grieco, "Optimal Information Centric Caching in 5G Device-to-Device Communications," *IEEE Trans. on Mobile Compu.*, vol.17, no.9, pp.2114-2126, Sept. 2018.

[13] X. Chen, M. Wang, S. Jia, C. Xu, "Energy-Aware Fast Interest Forwarding for Multimedia Streaming over ICN 5G-D2D," *International Conference on Image and Graphics (ICIG)* , pp.353-365, 2017.

[14] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa and L. Zhang, "VANET via Named Data Networking," *in Proc. IEEE Comput. Commun. Soc. Workshops (INFOCOM WKSHPS)*, pp.410-415, 2014.

[15] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," *in Proc. IEEE Comput. Commun. Soc. (INFOCOM)*, pp.126-134, 1999.

[16] C. Bettstetter H. Hartenstein and X. Perez-Costa, "Stochastic Properties of the Random Waypoint Mobility Model," *ACM/Kluwer Wireless Networks*, vol.10, no.5, Sept. 2004.

[17] A. Barrat, M. Barthelemy, and A. Vespignani, "Dynamical processes on complex networks,"*Cambridge University Press*, 2008.

[18] B. Chen, L. Liu, H. Wang and H. Ma, "On Content Diffusion Modelling in Information-Centric Networks," *in Proc. IEEE Global Commun. Conf. (GLOBECOM)*, pp.1-6, 2017.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint* , arXiv:1301.3781 2013a.

[20] H. Zhu, P. Zhang, G. Li, J. He, H. Li, K. Gai, "Learning Tree-based Deep Model for Recommender Systems," *arXiv preprint* , arXiv:1801.02294 2018.

[21] R. Gonzalez, F. Manco, A. Garcia-Duran, J. Mendes, F. Huici, S. Niccolini, M. Niepert et al. "Net2Vec: Deep Learning for the Network," *Proc. of ACM Spec. Interest Gr. on Data Comm. Workshop (SIGCOMM WKSHPS)*, pp.13-18, 2017.