Broadcast-Based Peer-to-Peer Collaborative Video Streaming Among Mobiles

Man-Fung Leung and S.-H. Gary Chan

***Abstract*—In traditional mobile streaming networks such as 3G cellular networks, all users pull streams from a server. Such pull model leads to high streaming cost and problem in system scalability. In this paper, we propose and investigate a fully distributed, scalable, and cost-effective protocol to distribute multimedia content to mobiles in a peer-to-peer manner. Our protocol, termed Collaborative Streaming among Mobiles (COSMOS), makes use of broadcasting and data sharing to achieve high performance (in terms of delay, cost fairness, stream continuity, etc.). In COSMOS, only a few peers pull video descriptions from base stations. Using a free broadcast channel (such as Wi-Fi and Bluetooth), they share the streams to nearby neighbors. As a result, COSMOS greatly reduces the streaming cost and cellular bandwidth requirement. Furthermore, as video streams are supplied by multiple peers, COSMOS is robust to peer failure. Since broadcasting is used to distribute video data, COSMOS is highly scalable to large number of users. In COSMOS, peers autonomously determine whether to broadcast packets or not in order to efficiently use of the channel bandwidth. By taking turns to pull descriptions, peers can effectively share, and hence substantially reduce, streaming cost. As broadcast scope is small and peers can often obtain a number of streams from its neighbors, COSMOS achieves low delay and excellent stream continuity.**

***Index Terms*—Ad-hoc, collaborative streaming, mobiles, peer-to-peer, video broadcasting.**

I. INTRODUCTION

IN TRADITIONAL mobile streaming system such as 3G cellular network [1], users in the range of a base station "*pull*" streams from a remote server.[1] Depending on the amount of data streamed, they are charged for a certain streaming fee. As cellular channels are precious and limited in number, such approach is usually costly and is not scalable in terms of user capacity. In addition, users have to be in the coverage of the base station in order to be served. This greatly limits the pervasive deployment of multimedia streaming services.

With the evolution of mobile technology, we have witnessed in recent years great improvement in processing capability, battery power, and memory of mobile devices such as cellular phones and PDAs. Many mobile devices nowadays can exchange data using some secondary wireless channels such as IEEE 802.11 or Bluetooth [2], [3]. These secondary channels can often be turned on simultaneously with the cellular channel. As these channels are broadcasting and free in nature, we can make use of them to achieve cost-effective collaborative peer-to-peer streaming. As a matter of fact, peer-to-peer architecture has been shown to be an effective way for data delivery, especially for media streaming [4]. It reduces the bandwidth cost and increases the scalability of the system.

In this paper, we propose and study a fully distributed, scalable and cost-effective protocol called **Co**llaborative **S**treaming among **Mo**bile**s** (COSMOS) to share multimedia stream.[2] We make use of the secondary free channels already available in mobile devices today to form a broadcast-based mobile network, where mobiles share their multimedia streams among their peers in near vicinity. In COSMOS, some mobiles are "pullers," who pull streams from the base station and hence are the payers in the system. By taking turns in being the pullers, the mobiles can share streaming cost more fairly. Such approach is particularly attractive for popular multimedia streams such as live sport events and breaking news, which may be of interest to many users in the vicinity of each other.

COSMOS may incorporate with multiple description coding (MDC) to improve its failure resilience, where a video stream is coded into multiple independent "descriptions" which may be arbitrarily combined and played back.[3] The more descriptions one receives, the higher the video quality is. A peer in COSMOS randomly selects and downloads a video description through the cellular channel, and broadcasts to its nearby neighbors using the secondary wireless channel.

There are two approaches for multimedia data broadcasting in COSMOS. The first one is that a description is broadcasted with a certain fixed broadcast scope or hop from a puller. For the second one, the broadcast scope is dynamic depending on node density. A node determines whether video broadcast is necessary so as to efficiently use the secondary channel bandwidth.

In COSMOS, a peer may collect more than one description while paying only for one. A peer does not pull any description if it has already received full set of descriptions from the broadcast channel. In COSMOS, we hence have two types of mobiles at any instant of time: *pullers* who pay for their streams, and *passive receivers* who do not need to pay any at that time. Clearly, as opposed to the traditional "what-you-pull-is-what-you-get" approach, our system is of much lower cost and higher user capacity. Its performance improves, rather than decreases, with the number of peers (in terms of delay, cost, fault resilience, etc.).

Manuscript received July 1, 2006; revised November 14, 2006. This work was supported, in part, by Direct Allocation Grant (DAG05/06.EG10) of HKUST and Hong Kong Innovation Technology Fund (GHP/045/05).

The authors are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: csfung@cse.ust.hk; gchan@cse.ust.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2006.889093

[1]In this paper, we use "peers," "mobiles," "nodes," and "users" interchangeably.

[2]We focus on only the sharing mechanism of COSMOS here. Issues such as payment security and authentication, right management, and attacks are beyond the scope of this paper.

[3]For generality, we assume in the following, MDC is used.

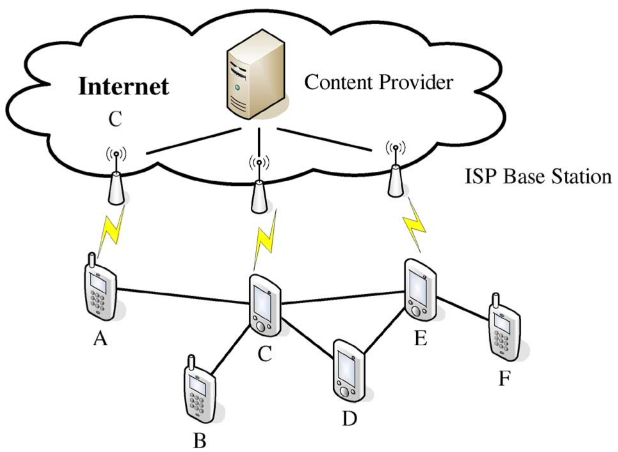


Fig. 1. A COSMOS system in which mobile devices form a peer-to-peer network. They cooperatively download and share video descriptions pulled from a content provider.

We use the following example to illustrate the basic concept of COSMOS. Fig. 1 shows six mobile devices labeled from A to F forming a wireless broadcast network using their secondary channel. The solid lines indicate the reachability of the data through the broadcast channel. In the network, by broadcasting beacon messages, each peer learns which descriptions its neighbors are receiving. Mobiles A, C, and E pull different video descriptions from the content provider. They relay the multimedia data received to others by rebroadcasting using the secondary channel, hence providing service to B, D, and F. Note that because some nodes (B, D, and F in the figure) may not need to download data, the total streaming cost is reduced.

Clearly, COSMOS enjoys the following strengths:

- *Reduction in streaming cost and bandwidth requirement per node:* Because a peer may need to pull only a video description while enjoying the full video quality, COSMOS achieves much lower cellular bandwidth requirement and streaming cost.
- *Low delay:* The delay depends on the broadcast hop. In COSMOS, we found that one to two broadcast hops would be enough to achieve good performance. This leads to low video delay.
- *High scalability:* COSMOS is a fully distributed broadcast protocol. It is simple to implement, and each mobile may needs to keep some simple and partial information on its neighbors and does not need to know any network topology. Therefore, COSMOS is light-weight with low exchange overhead in terms of control messaging and membership maintenance. As a single broadcast may cover a large number of users without the need of any mobile forwarding, it is scalable to large group.
- *Robustness to peer dynamics and failures:* Each mobile may receive multiple descriptions at the same time. Therefore, the departure or failure of a node would not break video continuity. Furthermore, some peers may enjoy replicated description depending on their network locations. Such redundancy greatly improves the robustness of the system.

In this paper, we address several important issues in COSMOS. The first one is how peers pull different video descriptions in a distributed manner so as to achieve good video quality and channel utilization even in the presence of network dynamics. The second issue is the protocol on how peers take turns in pulling descriptions to achieve good fairness in cost sharing. We also propose a generic architecture framework for the implementation of COSMOS peer.

We have conducted simulation to evaluate COSMOS performance and compare it with a recently proposed scheme, CHUM [5], [6]. Our results show that COSMOS achieves better performance in delay, cost fairness, video bitrate achieved, and resilience to peer failure, with some sacrifice in cost.

The rest of the paper is organized as follows. In Section II, we overview relevant previous work. We present in Section III the COSMOS problem formulation. The details of COSMOS protocol is discussed in Section IV. In Section V, we present a generic framework of COSMOS. We show our simulation results and the comparison with CHUM in Section VI. In Section VII, we conclude with some future work.

II. Related Work

We briefly discuss previous work as follows. A recent work, CHUM (Cooperating ad-hoc networking to support messaging), which shares multimedia data among mobile devices in an ad-hoc manner [5], [6]. To minimize the streaming cost, one of the peers pulls multimedia content and shares it to their peers. To the best of our knowledge, it is the only recent work closely related to our work; therefore, we will compare our scheme with that. However, CHUM is based on a tree topology based on point-to-point communication, while COSMOS is based on a mesh topology with broadcasting. Therefore, COSMOS has lower delay, better fault resilience, lower processing requirement and lower maintenance overhead at nodes.

Using a secondary channel for mobile data delivery has been investigated in iCAR (Integrated cellular and ad-hoc relaying systems), which integrates cellular system with ad-hoc network [7], [8]. However, while the previous work focuses mainly on how to relay data from a mobile to the base station, COSMOS does not have complicated routing issues. Furthermore, as opposed to COSMOS, there is no fault and user dynamic issues in iCAR (as the relay points are stationary and reliable). Another work, SPAWN (Swarming protocol for vehicular ad-hoc networks), considers content delivery in wireless (vehicular) networks [9], [10]. The protocol uses a gossip mechanism to exchange information among users. COSMOS, on the other hand, does not need point-to-point communication, and hence is much simpler and achieves lower processing and exchange overhead.

We extend the work in [11] by considering dynamic, rather than fixed, broadcast scope where each peer determines whether to broadcast a packet depending on its local density. This greatly reduces flooding and the channel redundancy. In addition, we present a framework for the implementation of COSMOS.

WIANI (Wireless infrastructure and ad-hoc network integration) is a multi-hop WLAN which makes use of ad-hoc channel for scalable content delivery [12]. As compared to WIANI, COSMOS routing is simpler as it is based on broadcasting and mobiles do not need to know the addresses of each other (no end-to-end communication and routing is necessary). Furthermore, as opposed to WIANI, COSMOS mobiles use a secondary channel for peer-to-peer broadcasting and sealed to consider cost issues in stream sharing.

MACA (Mobile-assisted connection-admission) and MADF (Mobile-assisted data forwarding) construct an ad-hoc overlay network with a secondary channel on fixed infrastructure to achieve load balancing in a cellular network [13], [14]. An user in a hot cell connects to its neighbor cold cell through other users. While the work is based on unicast, COSMOS uses broadcasting to distribute data to achieve higher user scalability.

There has been much work on video broadcasting (see, for example, [15]–[18] and references therein). However, these work are related to wired broadcast network. In mobile broadcasting, such as the case we are considering, we need to consider *whom* the broadcaster should be and the *scope* of the broadcast. Our system does not need the information of peer location as proposed in [19].

COSMOS may use MDC, which is a compression scheme where a video clip is coded into multiple independent descriptions. MDC has been extensively studied before (see, for example, [20]–[22]). Different schemes provide different tradeoffs in terms of compression performance and error resilience. The use of MDC for video streaming has been widely studied. Padmanabhan *et al.* propose that introducing redundancy can provide robustness in media streaming [23]. They use multiple distribution trees for data delivery, and MDC to provide redundancies in network paths and data. However, the tree management of this protocol is centralized, while in COSMOS, peers determine whether data broadcasting is necessary in a distributed manner. The authors in [24] investigate the advantage of path diversity to reduce the probability of simultaneous loss in the paths using Multiple Description Content Delivery Network (MD-CDN). In a more recent work [25], Mobile Streaming Media CDN (MSM-CDN) is used to overcome delivery challenges such as mobility, wireless, and user scalability. However, they mainly consider that each user downloads a full set of descriptions using point-to-point connections to servers, while we discuss in COSMOS how to *collaboratively* pull a stream and share data among the peers to reduce telecommunication cost.

Previous research has extensively studied distributed algorithms for selecting the rebroadcasting peers for data delivery using local neighbor information in wireless network [26]–[30]. However, the work mainly focuses on static data delivery. We need to address the dynamic selection of pullers and video descriptions so as to achieve good video quality and better fault resilience.

In a graph, a dominating set is a subset of nodes such that a node is either in the dominating set or a direct neighbor of a node in the dominating set. The minimum dominating set (MDS) problem is to compute a dominating set of minimum size. An algorithm for MDS can be used to find a minimum set of pullers in COSMOS network. Approximate algorithms for selecting a dominating set are investigated in [31], [32]. However, the approximation MDS algorithms mainly focus on the running time and approximation efficiency of the algorithms. The dominating set selected does not change if the graph is fixed. In COSMOS, we need to study a mechanism for peers to take turns in pulling data so as to achieve fairness in streaming cost sharing.

III. Problem Formulation

In this section, we formulate the broadcast problem of COSMOS to illustrate the complexity of the problem. As there is no polynomial time algorithm for the problem recently, we present our distributed heuristic in next section.

As wireless broadcasting may cause the duplication of packets and the contention of the wireless medium, it is important to construct a broadcast tree for COSMOS such that the number of broadcasters is minimized. We simplify the problem by considering one puller in the system since the simplified problem is sufficient to show the hardness of COSMOS broadcast problem. Let $G(V,E)$ represents the mobile network, where V is the set of mobile nodes and $E = \{(i,j)\}$ is the set of edges where $i \neq j$ and $(i,j) = 1$ if node i can reach node j in its power range (assuming equal power for all nodes). In order to minimize the cost of COSMOS, we need to find a minimum connected set B such that

$$B \subseteq V, \tag{1}$$

and, for any node, say u, not in B,

$$\exists v \in B \text{ s.t.} (u,v) \in E. \tag{2}$$

We can select any one node in set B as the puller of the network.

This is the well-known minimum connected dominating set problem and has been proved to be NP-hard [33]. To share the workload of the puller, we need to consider that there are more than one puller in the system. This makes the problem harder than the problem formulated above. Therefore, in COSMOS, we need to propose a heuristic to select peers for video data rebroadcasting in a distributed manner. This heuristic is presented in the next section. Due to the complexity and dynamic of the system, the analysis of the system is intractable and hence we will use simulation to study our scheme.

Before we proceed, we define the following notations which will be used for the remainder of this paper:

- $N(u)$: The set of all neighbors of node u.
- $S(u,d)$: The set of neighbors of node u receiving description d from u.
- $W(u,d)$: The set of neighbors of node u in which they are not served with description d.
- $U(x)$: A function returning a random number uniformly distributed between 0 and x.

IV. COSMOS Description

In COSMOS, there is a video server where the video is encoded into D independent video descriptions using MDC ($D \geq 1$). A peer randomly chooses and pulls a description through the cellular link, which provides some basic video quality. To improve its video quality, a peer may pull other descriptions or be helped from its neighbors through the broadcast channel.

Given above, we present the details of COSMOS protocol. We first discuss how a peer broadcasts its video descriptions with a fixed broadcast scope in Section IV-A. Then, we present an algorithm with dynamic broadcast scope in Section IV-B. In Section IV-C, we discuss how peers take turns to pull video data so as to achieve cost fairness, and how to handle peer dynamics and failures.

A. Video Distribution With Fixed Broadcast Scope

In this protocol, pullers simply broadcast the video data they pull to their neighbors with a certain broadcast scope. As the delay increases with the broadcast scope, the scope would be limited to a certain value S. Each broadcast packet has a *Time-to-Live* (TTL) field which is initialized to S (e.g. $S = 2$). When a peer receives the packet, it decrements TTL by 1. Peers rebroadcast the packet so long as TTL is greater than 0.

As an example, refer to Fig. 1 again with $S = 1$. Mobiles A, C, and E pull different video descriptions from the content provider. As S is one, each description is broadcasted in one hop from the corresponding puller. Therefore, while pullers A, C, and E pay one description, they enjoy two, three, and two descriptions, respectively. Nodes B, D, and F are passive receivers enjoying one, two, and one description, respectively.

B. Video Distribution With Dynamic Broadcast Scope

When user density is high, using a fixed broadcast scope leads to high packet redundancy in the wireless channel. On the other hand, when user density is low, it may be beneficial to extend broadcast scope to reduce the streaming cost. Therefore, we propose here a distributed broadcast algorithm with dynamic broadcast scope which effectively distributes the descriptions among peers of different local density without wasting the channel bandwidth. The main idea of the algorithm is that a node does not rebroadcast a video packet if many of its neighbors are being served with that description. To achieve this, a node collects the local information of its neighbors to determine whether it should rebroadcast a packet or not.

To achieve dynamic scope, peers exchange local information with their neighbors (we refer neighbors of a node as the mobiles within the power range of the node). Fig. 2 shows the format of a beacon packet. Each node periodically broadcasts beacon packets with only one hop. The packet contains *Sender_Node_ID* which is the source ID of beacon packet. The $\langle Received\_Description\_Index, Upstream\_Node\_ID \rangle$ pair indicates which descriptions are received by the sender of the beacon and ID of its direct neighbor that the description is received from. Since a peer has only one upstream node for a description received, a *Received_Description_Index* is unique and does not equal another one in a beacon packet.

The format of the video packet is shown in Fig. 3. By the fields *Description_Index* and *Upstream_Node_ID*, a node knows which of its neighbor (i.e. the "upstream node") broadcasting the description. If a node receives a description from multiple neighbors, the upstream node of the earliest packet is put in the *Upstream_Node_ID*.

Note that the upstream node of a video packet is either a source (puller) or an intermediate rebroadcasting node. If it is the source and has some neighbors (as known from beacon messages), it broadcasts the video packets pulled. On the other hand, if it is not a puller, it determines whether to rebroadcast the packet or not according to the following.

32 Bits

Sender_Node_ID	Unused / Reserved
Received_Description_Index_1	Upstream_Node_ID_1
Received_Description_Index_2	Upstream_Node_ID_2
Received_Description_Index_3	Upstream_Node_ID_3
⋮	
Received_Description_Index_n	Upstream_Node_ID_n

Fig. 2. The format of beacon packet. Each node broadcasts beacon packets to its one-hop neighbors periodically. By exchanging beacon packets, peers can learn the local service information and the local topology within one hop.

32 Bits

Puller_Node_ID	Upstream_Node_ID	
Time_to_Live	Number_of_Neighbors	
Description_Index	Switch	RPT
Sequence_Number		
Video_Data		

Fig. 3. The format of video packet used during broadcasting video data.

Suppose it is the first time for a non-puller, say u, receives a video packet of description d. If more than a certain fraction of its neighbors have already received description d from nodes other than u, it does not rebroadcast the packet. In other words, u evaluates

$$\chi = \frac{|S(u,d) \cup W(u,d)|}{|N(u)|}, \tag{3}$$

which is the percentage of potential beneficiaries that can be served by node u. If χ is less than a certain threshold X_{Th} (a system parameter), u does not rebroadcast the packet. There are two extreme cases on the value of X_{Th}. If $X_{Th} = 0$, the system will work similar as COSMOS scheme with a fixed broadcast scope. Wireless medium contention may happen if network is too dense. On the contrary, if $X_{Th} = 1$, only pullers broadcast descriptions and there is no description rebroadcast. More pullers are required and hence the streaming cost increases. So, the setting of X_{Th} is a trade-off between wireless medium traffic and streaming cost. Note that $N(u)$, $S(u,d)$, and $W(u,d)$ may be a moving average as their values continuously change due to network dynamics. They can be obtained from the beacon messages, and do not need to be very accurate for the system to work.

A node to rebroadcast a video packet first waits by a random period to avoid collision. In general, we would like a node with a larger χ to rebroadcast its packets earlier, so that more nodes would be benefited by such broadcast. To achieve this, the rebroadcast delay T_D is calculated as

$$T_D = U\left(\eta \times (1 - \chi)\right), \tag{4}$$

where η is the maximum rebroadcast delay in the system. Upon receiving a packet, a node suppresses its rebroadcast schedule of the same description.

In order to limit the source-to-peer delay of the system, there is a maximum limit on the broadcast scope of a packet given by S. As in the case of fixed broadcast scope, *Time_to_Live* (TTL) is initialized to S and decremented by one on each hop. A node can only rebroadcast a packet if TTL is greater than zero and $\chi \geq X_{Th}$.

As an example, refer to Fig. 1 again with $S = 2$ and $X_{Th} = 0.5$. Mobiles A, C, and E pull different descriptions d_A, d_C, and d_E, respectively. When a peer receives a description, it determines whether to rebroadcast or not by evaluating χ. For example, C receives description d_A in the first broadcast (one hop) from puller A. From beacon messages, it knows that its neighbors B, D, and E are not receiving the description from any other peer. C hence gets $\chi = 0.75$, which is larger than X_{Th}. So, it rebroadcasts the description d_A. However, when D receives description d_C, it finds that C is the puller and E is already receiving the description transmitted from C. Therefore, D evaluates $\chi = 0$ and discards the rebroadcast operation to save some wireless channel bandwidth. As a result, A, C, and E are pullers paying one description and enjoying three descriptions, while B, D, and F are passive receivers enjoying three, three, and one descriptions, respectively.

C. Cost Sharing and Group Dynamics

Since peers randomly select video descriptions to pull, it may happen that two mobiles pull and broadcast the same description to each other. Though this redundancy leads to some failure resilience, it increases streaming cost and cellular bandwidth requirement. It is beneficial for one of them to pull another description to improve the video quality and bandwidth utilization. In general, we would like the node with more neighbors to keep sharing (broadcasting) its pulled description, as its description broadcasted is beneficial to more users. The *Number_of_Neighbors* field in the video packet is to resolve whom the puller should be. In case of a tie on the number of descriptions, the peer with the largest *Puller_Node_ID* would pull the description. The peers without the full video descriptions would randomly choose any of its missing descriptions to pull. If the peer finds that all the descriptions are already available from the broadcast secondary channel, it becomes a *passive receiver*.

As the number of peers increases, some may not need to pull any descriptions. To more fairly distribute the load and the streaming cost among the peers, COSMOS has a mechanism to exchange the roles between pullers and passive receivers when a puller has been downloading video data for some time.

A certain time T_S seconds before a puller would like to stop pulling, it sets the *Switch* flag of the video packet and set *RPT* (residual pull-time) as T_S to inform other peers on its intention of role switching. The packet is broadcasted within the scope S. A passive receiver who receives the video packet with *Switch* flag set starts a random timer with time $U(T_S)$. If by this time it does not receive the corresponding description from its neighbors, it becomes a puller of the description by broadcasting the description it pulls. In other words, the one with the earliest timer becomes the puller.

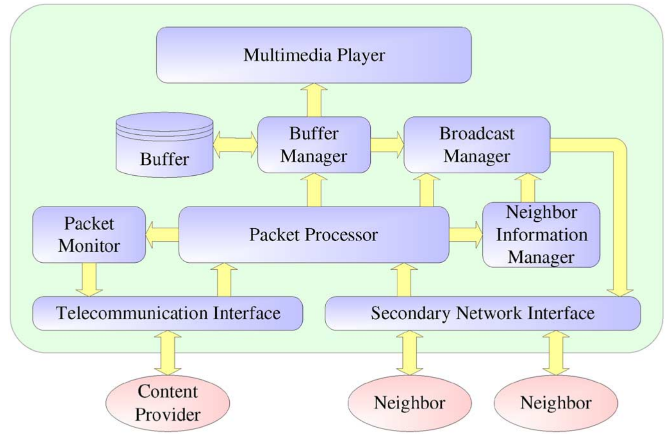


Fig. 4. A generic system architecture diagram for a COSMOS node.

Before a puller leaves the network, it notifies other peers so that they can contend to pull the description for sharing. A leaving puller sets the *Switch* flag of the video packet and set *RPT* to 0. The other peers, upon receiving the packet, start a random timer of value $U(\widehat{L})$ where $\widehat{L}$ is some constant. What follows is similar to the role switching mechanism.

COSMOS is robust since some descriptions may be duplicated. If a puller fails or a peer moves out of the coverage range of an upstream node, the same description can be supplied from other peers. Peers can buffer and order the video packets received according to their *Description_Index* and *Sequence_Number* of the packet. In this way, duplicate packets can be identified and removed. Then, video packets can be assembled together before presenting to the decoder. If a peer finds that some of its video descriptions have been missing for a time (due to, for example, node failures or out of coverage ranges of upstream nodes), it starts pulling and sharing the video description after a certain random timer $L+U(\widehat{L})$, where L and $\widehat{L}$ are some constants.

V. COSMOS Framework

In this section, we present a generic system architecture for the implementation of COSMOS. Fig. 4 depicts the system framework of a COSMOS node. *Telecommunication Interface* allows communication with a content provider for the peer to pull multimedia data. The *Secondary Network Interface* is used to broadcast and receive packets. *Multimedia Player* decodes multimedia data received and displays it on a mobile device. Apart from the above components, there are five key modules in the system.

1) *Packet Processor:* The functions of this unit are to extract information from packets (beacon packets and video packets) received from the *Telecommunication Interface* and the *Secondary Network Interface* and deliver it to other modules. Upon processing a beacon packet, the module provides neighborhood information to *Neighbor Information Manager*. If a video packet is received, it extracts the required information and passes to *Packet Monitor*, *Buffer Manager*, and *Broadcast Manager* modules.
2) *Buffer Manager:* This module manages video data received and stores it in *Buffer* memory. When a video

packet (Fig. 3) is received, *Packet Processor* module delivers *Video_Data* to *Buffer Manager*. It then stores and orders the data in *Buffer* memory according to *Description_Index* and *Sequence_Number*. In addition, this module supplies suitable video data to *Multimedia Player* and *Broadcast Manager* for display and broadcast, respectively.

3) *Neighbor Information Manager:* When *Packet Processor* module obtains a beacon packet (Fig. 2), it extracts *Sender_Node_ID* and pairs of $\langle Received\_Description\_Index, Upstream\_Node\_ID\rangle$. This neighbor information is then passed to *Neighbor Information Manager* for book-keeping, and to *Broadcast Manager* for data broadcast.

4) *Broadcast Manager:* The primary function of this module is to broadcast beacon packets periodically. It constructs a beacon packet by setting *Sender_Node_ID* and embedding a list $\langle Received\_Description\_Index, Upstream\_Node\_ID\rangle$ pairs, which indicate the descriptions received and the upstream nodes of the descriptions. Then, the packet is broadcasted through the *Secondary Network Interface*.
Moreover, *Broadcast Manager* handles broadcasting of video packets and constructs video packets for broadcast operations with suitable *Video_Data* supplied from *Buffer Manager*. If a video packet is received from the *Telecommunication Interface*, the peer is a puller. *Packet Processor* then requests *Broadcast Manager* to broadcast the packet. It proceeds the broadcast operation if there is some direct neighbors recorded in *Neighbor Information Manager*.
On the other hand, if a video packet is obtained through the *Secondary Network Interface*, it decides whether a rebroadcast is necessary by evaluating χ using the local information provided by *Neighbor Information Manager*. If it is needed, the rebroadcast operation is scheduled with a delay T_D given before.

5) *Packet Monitor: Packet Monitor* keeps track of video packets received by recording *Description_Index* and *Sequence_Number*. When *Packet Processor* receives a video packet with *Switch* flag set, this means that a puller would like to stop pulling. Therefore, *Packet Monitor* starts a random timer $U(T_S)$. If by this time the corresponding description is not received, it starts pulling video data. In addition, this unit monitors whether there is missing or redundant data. The peer will become a puller of multimedia data if the corresponding data have been missing for some time, or it stops pulling data when it receives the same description as it pulls.

VI. Illustrative Simulation Results

In this section, we describe the simulation used to compare the performance of COSMOS and CHUM. Peers are randomly placed in a 100 units $\times$ 100 units area with power range of 15 units. Peers enter the system with Poisson arrival rate λ (request/unit time). Each peer remains in the system with exponential time of mean $1/\mu$ unit time if it does not fail before then. A peer may fail with rate f (request/unit time). Therefore, at steady state, the average number of peers in the system is $N = \lambda/(\mu+f)$ and the probability that a peer fails is $f/(\mu+f)$. In this paper, we normalized the time such that $\mu = 1$ (request/unit time).

We consider that bandwidth is normalized such that a full stream of a video clip is of bandwidth 1. For both COSMOS and CHUM, it may incorporate with MDC of D descriptions $(D \geq 1)$ and each description is of lower bandwidth of a full stream. To account for MDC coding inefficiency, we consider that each description is of bandwidth $(1+\delta)/D$, where δ is a bandwidth dilation factor. We have implemented an event-driven simulator in C++ to study the system. All data are taken at steady state. In the simulation, pulling time and transmission time are assumed to be negligible.

A. Metrics

Consider peer i with its system time T_i. Let t be an instant of its lifetime in the system and N_{ex} be the number of peers examined. We consider the following performance metrics in our study.

1) *Receiving Channel Traffic:* Receiving channel traffic refers to the bandwidth required in receiving data for a user through the broadcast channel. Let $V_i(t)$ be the bandwidth that peer i receives at time t. We are interested in the average receiving channel traffic given by

$$\frac{\sum_i \int_{T_i} V_i(t)dt}{\sum_i T_i}. \tag{5}$$

2) *Total Channel Traffic:* Total channel traffic refers to the total bandwidth required for a user. Let $B_i(t)$ be the sum of bandwidth that peer i broadcasts and receives at time t. We define the average total channel traffic as

$$\frac{\sum_i \int_{T_i} B_i(t)dt}{\sum_i T_i}. \tag{6}$$

3) *Delay:* This is the minimum number of broadcasts before a peer receives a particular description. Let $H_i(t)$ be maximum delay of all its descriptions received at time t. We are interested in the average delay defined by

$$\frac{\sum_i \int_{T_i} H_i(t)dt}{\sum_i T_i}. \tag{7}$$

4) *Cost:* It refers to the streaming cost per unit time for a user. Let $P_i(t)$ be the total description bandwidth that peer i pulls at time t. We are interested in the average cost over all users defined as

$$\frac{\sum_i \int_{T_i} P_i(t)dt}{\sum_i T_i}. \tag{8}$$

5) *Cost Fairness:* We define streaming cost per unit time for peer i as

$$C_i = \frac{\int_{T_i} P_i(t)dt}{T_i}. \tag{9}$$

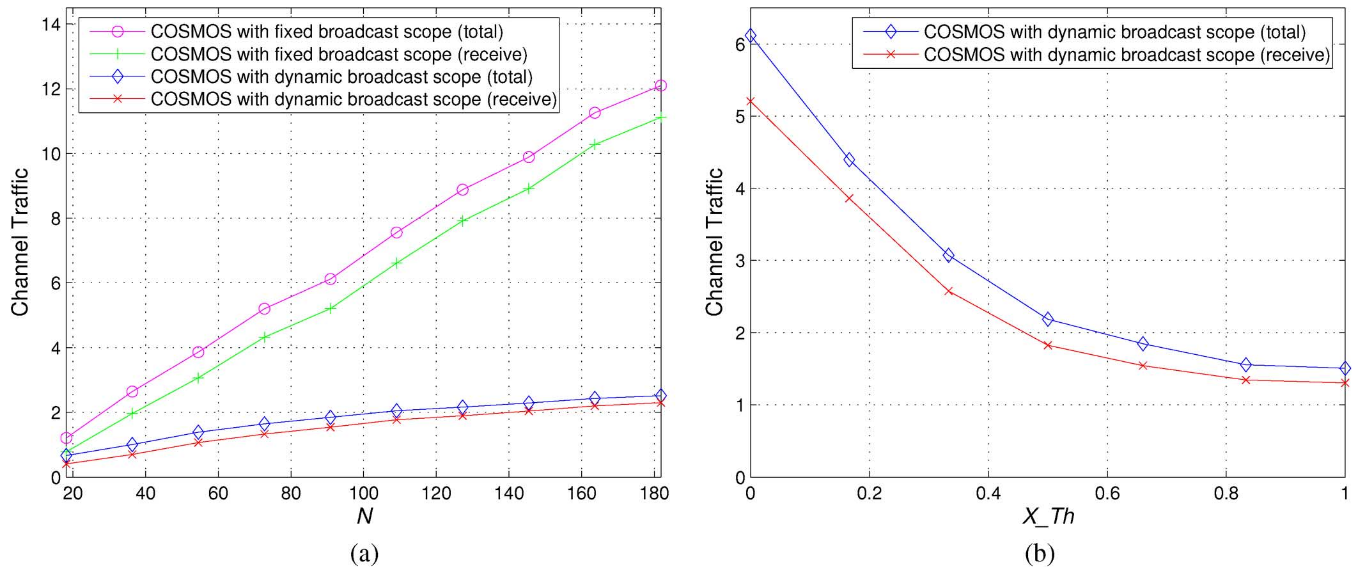


Fig. 5. Channel traffic for two COSMOS schemes (fixed broadcast scope and dynamic broadcast scope) with $D = 2$. (a) Channel traffic versus average number of peers. (b) Channel traffic versus broadcast threshold.

We further define cost fairness using Jain's fairness index as given by

$$\frac{(\sum_i C_i)^2}{N_{ex} \sum_i C_i^2}. \tag{10}$$

6) *Video Bitrate:* This refers to the (effective) description bandwidth received per unit time for a peer, excluding duplicate data received. Let $R_i(t)$ be the effective description bandwidth that peer i receives at time t. We are interested in the average video bitrate defined as

$$\frac{\sum_i \int_{T_i} R_i(t)dt}{\sum_i T_i}. \tag{11}$$

7) *Bitrate Fluctuation:* Let σ_{R_i} and μ_{R_i} be the standard deviation and the mean of $R_i(t)$, respectively. We define bitrate fluctuation of peer i as the coefficient of variation given by σ_{R_i}/μ_{R_i}. We are interested in the average bitrate fluctuation over all users as given by

$$\frac{\sum_i (\sigma_{R_i}/\mu_{R_i})}{N_{ex}}. \tag{12}$$

Unless otherwise stated, we set $f = 0.1$ (request/unit time), $\lambda = 100$ (request/unit time), broadcast scope $S = 2$, and broadcast threshold $X_{Th} = 0.66$ as baseline parameters. We consider $D = 1$ and $D = 2$.

B. Experiments

The channel traffic for COSMOS with a fixed broadcast scope and dynamic broadcast scope is plotted in Fig. 5(a). It illustrates that the total and receiving channel traffic for the scheme with a fixed broadcast scope raises much faster than that with dynamic broadcast scope as N increases. Note that, in general, the receiving channel traffic increases with the same rate as the total channel traffic. Most of the bandwidth is used on receiving packets broadcasted.

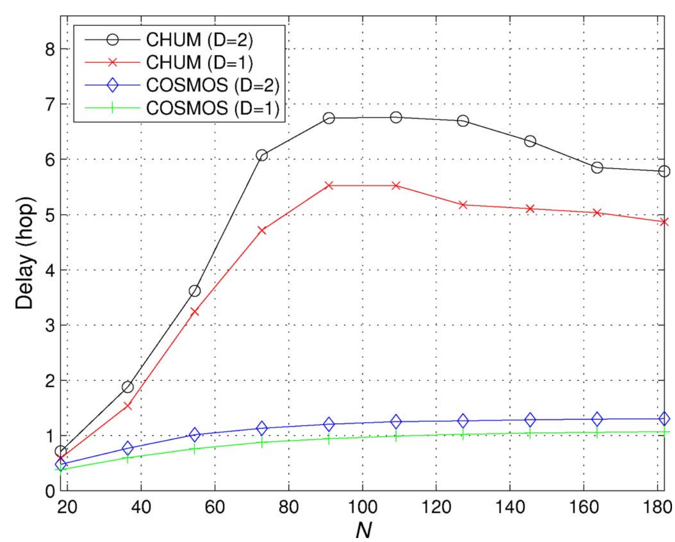


Fig. 6. Average delay versus average number of peers.

For COSMOS with a fixed broadcast scope, there are unnecessary broadcast operations when the number of peers is high. At that time, peers are located closely to each other and the user density is high. Some of them rebroadcast a description received to a group of users in which most of them may be covered by previous transmissions. Each peer receives many duplicate packets. In Fig. 5(a), the receiving channel traffic increases rapidly with the same rate as the total channel traffic. Most of the bandwidth is spent on receiving replicate packets. So, this wastes wireless channel bandwidth and may lead to wireless medium contention. On the other hand, with dynamic broadcast scope, a peer broadcasts a description if it can cover many potential beneficiaries who are not being served by other peers. This

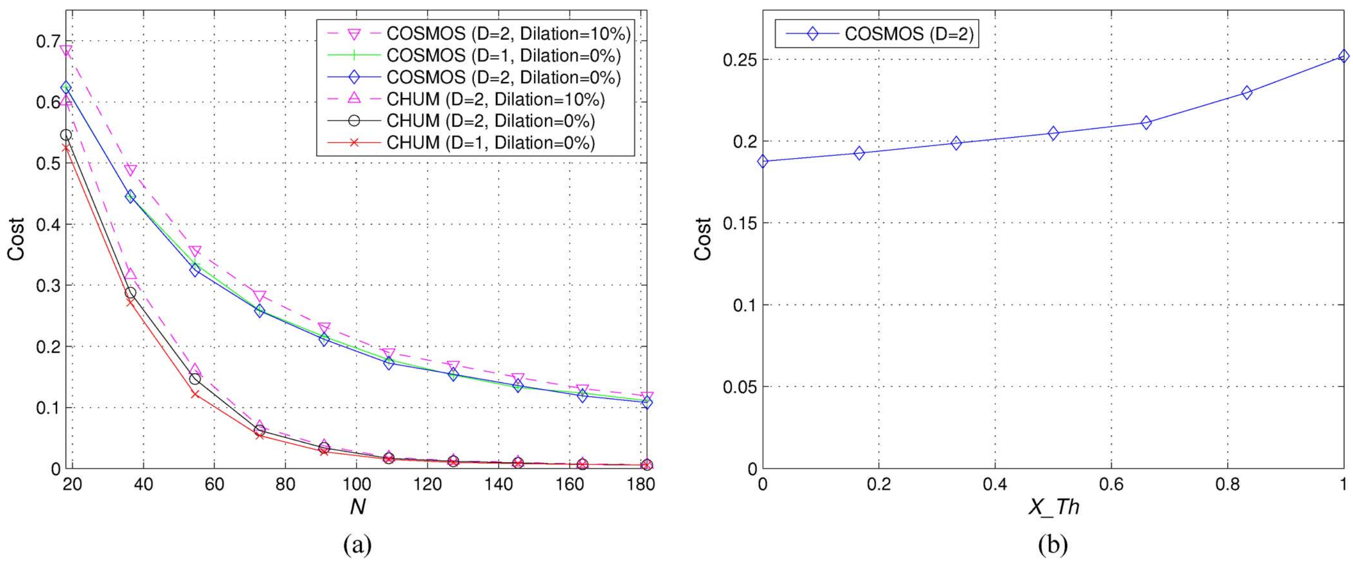


Fig. 7. Average cost. (a) Average cost versus average number of peers. (b) Average cost versus broadcast threshold.

can eliminate unnecessary rebroadcasts in the system. Therefore, the determination of broadcast necessity greatly reduces the usage of channel bandwidth.

The channel traffic of COSMOS with dynamic broadcast scope against broadcast threshold X_{Th} is plotted in Fig. 5(b). Channel traffic decreases while broadcast threshold X_{Th} increases. When $X_{Th} = 0$, the system works similar as COSMOS with a fixed broadcast scope. There are unnecessary description rebroadcasts and this results in high channel traffic. On the contrary, when $X_{Th} = 1$, only pullers broadcast descriptions and there is no description rebroadcast. So, the channel traffic is low.

In addition, we can deduce the bandwidth requirement of a peer from the figures. Let BW be the actual bandwidth of a video clip. Since the bandwidth of a full stream video clip is normalized to 1 in the simulation, the real bandwidth requirement of a peer equals to $(Total\ Channel\ Traffic \times BW)$. As the COSMOS scheme with a fixed broadcast scope has channel contention problem, we leave out the results of this scheme and consider the COSMOS scheme with dynamic broadcast scope for the following experiments.

Fig. 6 compares that the performance of COSMOS with CHUM in terms of delay as N increases. Clearly, COSMOS has a much lower delay. In COSMOS, the video packets are broadcasted within the scope S and hence delay is limited. For CHUM, the video data is forwarded until it reaches leaf nodes of forwarding tree. Accordingly, COSMOS has a better delay performance than CHUM.

When the CHUM network is small, its average tree height and the average delay increases with N. Nevertheless, average delay drops slightly beyond a certain value. This is because when the area becomes very crowded, the average path length (in terms of number of hops) to puller would not increase any further or even reduce due to shortest path routing. As a result, the delay of CHUM decreases when we take average over all users.

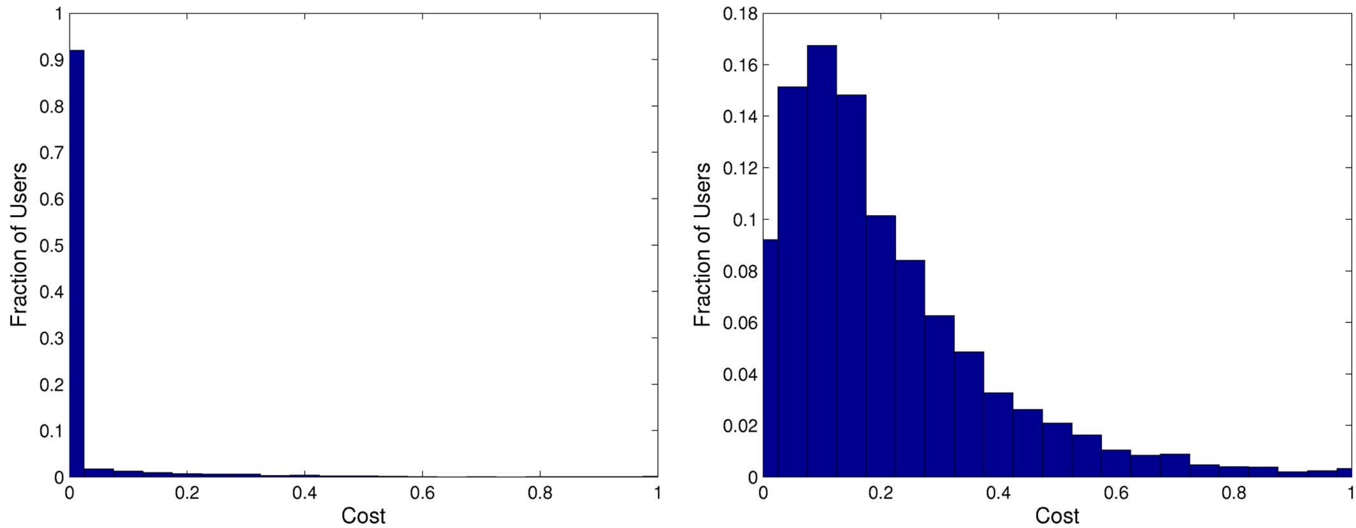
Furthermore, the number of descriptions D would affect the delay performance. This is because a node receives video descriptions from different pullers. The node may have different distances (in terms of number of hops) to the pullers. Since we consider the maximum delay of all descriptions received, more descriptions would lead to have a higher resultant delay. As a result, the average delay increases with the number of descriptions.

The average costs for COSMOS and CHUM are plotted in Fig. 7(a). As the inefficiency of MDC is significant for the cellular bandwidth pulled and affects the streaming cost, we also consider bandwidth dilation with dilation factor $\delta = 10\%$ for the schemes with number of descriptions $D = 2$. In general, the average cost increases by about 10% comparing to the schemes without bandwidth dilation.

When the number of users increases, more peers collaborate to pull video data and the cost is shared among them. At any instant of time, there is one peer pulls each video description in a CHUM network and it does not have duplicate packets. On the other hand, as COSMOS is a mesh approach to distribute multimedia data, some pullers may download replicate packets simultaneously. Consequently, the average cost of CHUM drops much faster than that of COSMOS as N increases.

Fig. 7(b) shows the average cost of COSMOS against broadcast threshold X_{Th}. The streaming cost increases with the broadcast threshold X_{Th}. When $X_{Th} = 1$, only pullers broadcast video descriptions and there is no description rebroadcast. So, it works similar as COSMOS with broadcast scope $S = 1$. More pullers are required and hence attaining higher streaming cost. On the other hand, for $X_{Th} = 0$, the system works like the COSMOS scheme with a fixed broadcast scope. Pullers broadcast descriptions which cover the nodes $S = 2$ hops away from the pullers. Therefore, the number of pullers and the streaming cost reduce.

However, since only one peer pulls a video description in CHUM network at the same time, the cost is distributed to a few of peers only. Some peers may not have chance to contribute and pull anything before leaving the system. The costs charged to peers are inconsistent. Fig. 8(a) shows a histogram to illus-



(a) (b)

Fig. 8. Cost distribution among peers in CHUM and COSMOS networks. (a) Cost distribution for CHUM with $D = 2$. (b) Cost distribution for COSMOS with $D = 2$.

trate the cost distribution in the CHUM network. The streaming cost for a peer is defined in (9). During the simulation period, the streaming cost is only distributed on a few of peers. Most of the peers need to pay for a very low cost only, or even do not need to pay, while only a small proportion of peers pay for most of the streaming cost of the entire system. On the contrary, in COSMOS system, many peers collaboratively pull video descriptions simultaneously. Thus, the streaming cost can be assigned to more peers, hence attaining higher fairness. The histogram for cost distribution in COSMOS network is plotted in Fig. 8(b). Most of the peers contribute on pulling descriptions for some time during their system time and hence the streaming cost can be fairly distributed to users.

Furthermore, with different average number of peers, we compare the fairness in cost sharing between CHUM and COSMOS. This is illustrated in Fig. 9. We use Jain's fairness index, which has value between 0 and 1, as an indicator for cost fairness. The larger the fairness index is, the fairer the streaming cost is distributed. Clearly, CHUM has a lower cost fairness than COSMOS, as the cellular streaming cost is more biased toward a few hosts. Moreover, the fairness in cost sharing changes with number of descriptions. When there are more descriptions in the system, more pullers are requires to pull a full set of descriptions from the content provider. Therefore, more peers contribute in pulling different descriptions and the streaming cost can be distributed to a larger number of users. As a result, the fairness in cost sharing increases with the number of descriptions D.

Despite our protocol to reduce description duplication in the network, there are unavoidable description redundancy in the network. This is in fact an advantage as it leads to higher failure tolerance. Furthermore, the use of MDC minimizes the stream disruption when peer fails. The neighbors of the failure peer would experience the loss of one description only rather than the entire video clip as in CHUM. Hence, the video bitrates of peers can be kept rather steady. In CHUM, once a peer fails, some downstream nodes may suffer stream discontinuity, which affects the bitrate adversely.

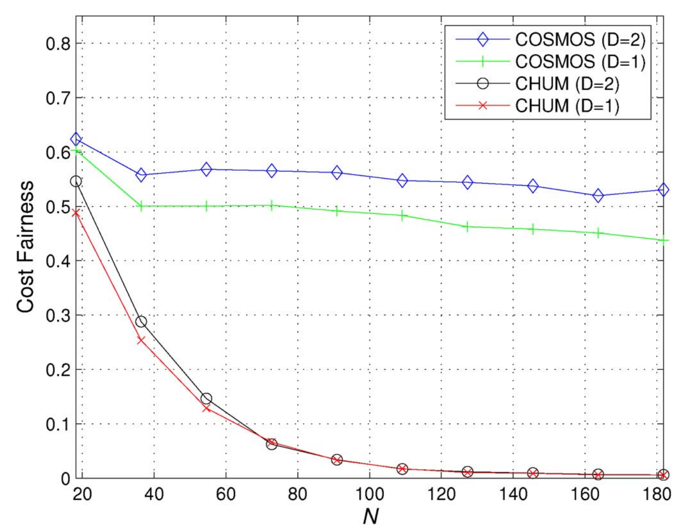


Fig. 9. Cost fairness versus average number of peers.

Fig. 10(a), Fig. 10(b), Fig. 10(c),and Fig. 10(d) plot four video bitrate profiles of peers in CHUM and COSMOS. The video bitrate obtained is the effective bitrate received for video decoding. The figures show that there are some bitrate drops which are due to peer leaves and peer failures. A peer leave causes a smaller gap while a peer failure results in a bigger gap. In general, CHUM peers have more bitrate drops during their system times. There is only one puller for each description. Due to a peer leave or failure, its downstream nodes cannot receive video data and they suffer some loss of video data. On the contrary, for COSMOS system, there are fewer bitrate drops because the video data delivery is based on mesh topology instead of tree topology as in CHUM. Description duplication reduces the effect of peer leaves and failures. Even though other nodes sometimes cannot supply duplicate description, the peer experience the loss of one description only. Especially, COSMOS scheme with $D = 2$ can bear the loss of a description. This is because the loss does not affect the continuity of the video since a usable

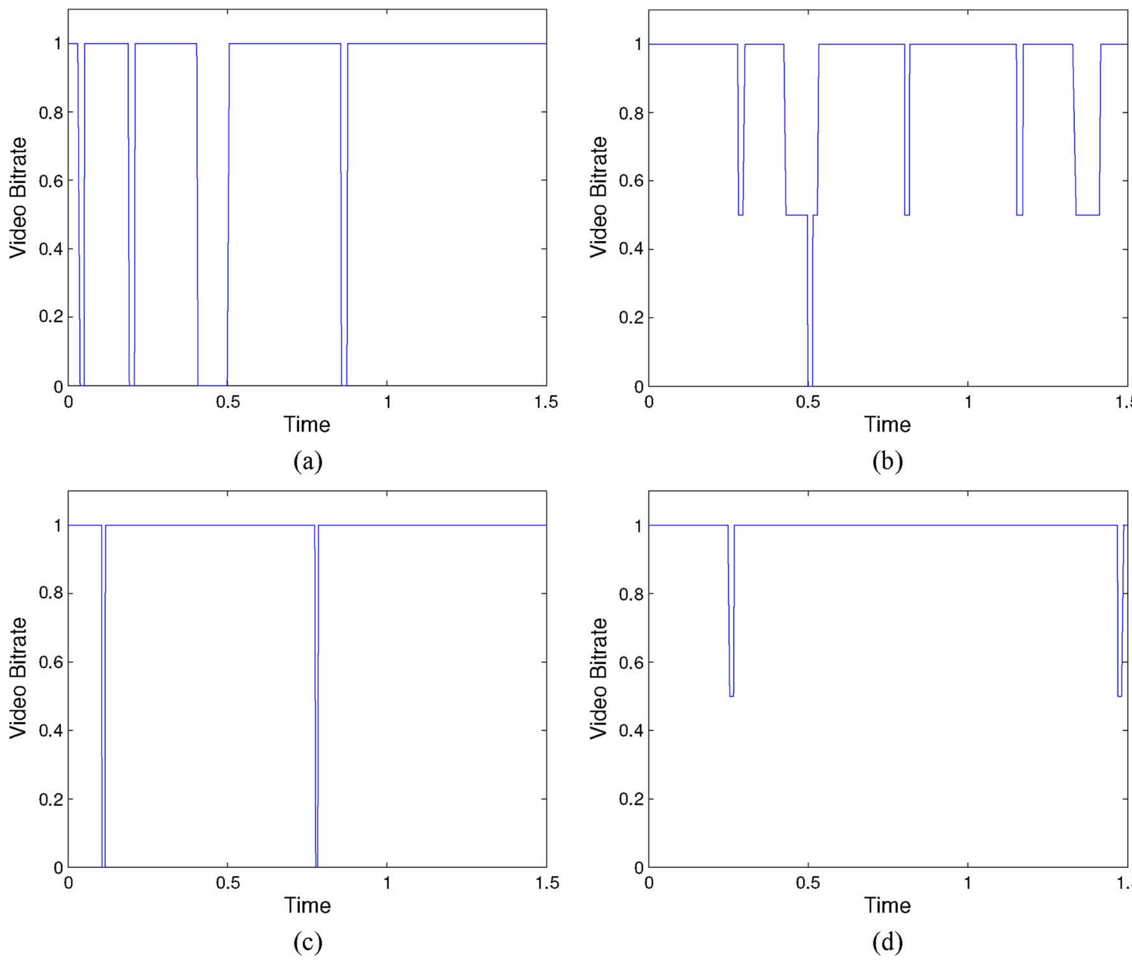


Fig. 10. Video bitrate profile of peers in CHUM and COSMOS networks. (a) CHUM peer with $D = 1$. (b)CHUM peer with $D = 2$. (c) COSMOS peer with $D = 1$. (d) COSMOS peer with $D = 2$.

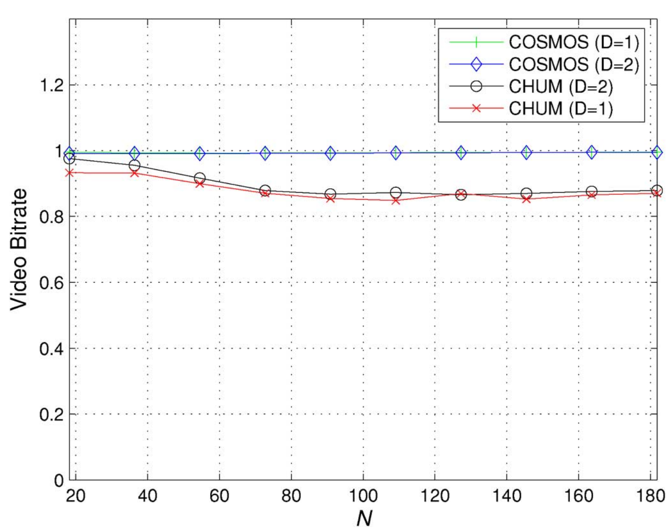


Fig. 11. Average video bitrate versus average number of peers.

quality is maintained whenever any description is correctly received. Hence, MDC would provide better failure resilience to the system.

We show in Fig. 11 the average bitrate received per node. Clearly, COSMOS achieves higher video bitrate than CHUM. This is because there is only one source (puller) for each video description in a CHUM network, while COSMOS is a mesh network with multiple pullers. COSMOS peers may receive duplicate descriptions and this reduces the effect of peer leaves and failures. Hence, higher effective video bitrate can be achieved. For COSMOS system, as the bandwidth of each description is defined as $1/D$, the number of descriptions received equals to $(D \times Video\ Bitrate)$. Furthermore, we show in Fig. 12 the bitrate variation as failure rate f increases. CHUM suffers from a larger bitrate fluctuation. The bitrate for COSMOS is more steady due to inheritance redundancy and MDC.

VII. Conclusion and Future Work

This paper proposes a scalable and cost-effective protocol called COSMOS (**Co**llaborative **S**treaming among **Mo**bile**s**) to distribute multimedia content to a group of mobile devices. COSMOS may incorporate with multiple description coding (MDC) for higher fault tolerance and stream stability. Each peer randomly selects and pulls an unavailable video description through the cellular link. It shares the description with its neighbors by broadcasting it so that its neighbors obtain more descriptions without increasing their streaming cost. We propose a broadcast algorithm with dynamic broadcast scope to effectively deliver multimedia data with good channel

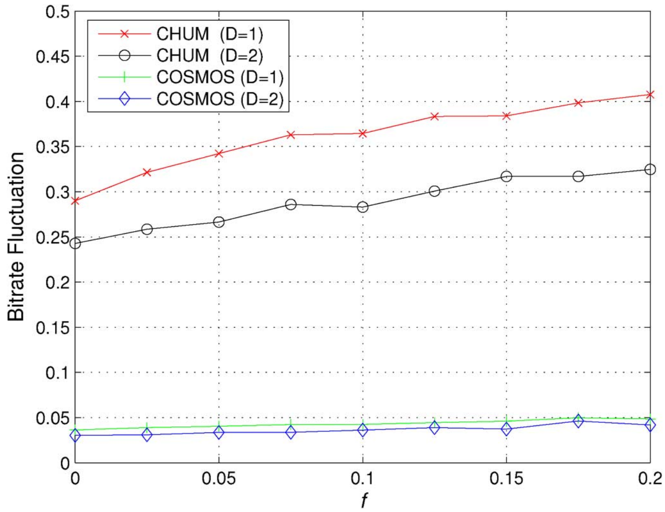


Fig. 12. Average bitrate fluctuation versus peer failure rate.

bandwidth utilization. Furthermore, peers take turns in pulling and hence the cost can be distributed more fairly. By controlling the broadcast scope, the delay of the system can be limited. In addition, we present a generic system framework of COSMOS for its implementation. Our simulation results show that COSMOS with dynamic broadcast scope substantially reduces the channel bandwidth usage (by more than 50% in our results). It achieves higher effective bitrate received (by as much as 20%), better fairness in cost sharing (by improving it more than 100%), higher fault tolerance, and more stable stream, as compared with a previous scheme (CHUM).

We are currently studying the heterogeneity of mobile devices in terms of their processing power and maximum number of descriptions they can decoded. We are also investigating the incentive issue so that peers may be willing to pull the streams and share them with neighbors.

Acknowledgment

The authors would like to thank Professor Oscar Au from the Department of Electrical and Computer Engineering at the HKUST for introducing the problem to them.

References

[1] L. Garber, "Will 3G really be the next big wireless technology?," *IEEE Computer*, vol. 35, pp. 26–32, Jan. 2002.

[2] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11, IEEE, Sept. 1999.

[3] P. Bhagwat, "Bluetooth: technology for short-range wireless apps," *IEEE Internet Computing*, vol. 5, pp. 96–103, 2001.

[4] D. Stolarz, "Peer-to-peer streaming media delivery," in *Proceedings of Peer-to-Peer Computing*, Aug. 2001, pp. 48–52.

[5] S. Kang and M. Mutka, "Mobile peer membership management to support multimedia streaming," in *Proceedings of ICDCS Workshop on Mobile and Wireless Networks*, May 2003, pp. 770–775.

[6] ——, "A mobile peer-to-peer approach for multimedia content sharing using 3G/WLAN dual mode channels," *Wireless Communications and Mobile Computing*, vol. 5, pp. 633–645, Sept. 2005.

[7] H. Wu, C. Qiao, S. De, and O. Tonguz, "Integrated cellular and ad hoc relaying systems: iCAR," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2105–2115, 2001.

[8] E. Yanmaz, O. Tonguz, S. Mishra, H. Wu, and C. Qiao, "Efficient dynamic load balancing algorithms using iCAR systems: a generalized framework," in *Proceedings of Vehicular Technology Conference, IEEE*, Fall 2002, pp. 586–590.

[9] S. Das, A. Nandan, G. Pau, M. Sanadidi, and M. Gerla, "SPAWN: a swarming protocol for vehicular ad-hoc wireless networks," in *Proceedings of ACM International Workshop on Vehicular ad hoc Networks*, 2004, pp. 93–94.

[10] A. Nandan, S. Das, G. Pau, M. Gerla, and M. Y. Sanadidi, "Co-operative downloading in vehicular ad-hoc wireless networks," in *Proceedings of Wireless On-demand Network Systems and Services*, 2005, pp. 32–41.

[11] M.-F. Leung, S.-H. G. Chan, and O. C. Au, "COSMOS: peer-to-peer collaborative streaming among mobiles," in *Proceedings of IEEE International Conference on Multimedia Expo (ICME) (invited)*, Toronto, Canada, July 2006, pp. 865–868.

[12] J.-C. Chen, S. Li, S.-H. Chan, and J.-Y. He, "WIANI: wireless infrastructure and ad-hoc network integration," in *Proceedings of IEEE International Conference on Communications (ICC)*, Korea, May 2005, pp. 3623–3627.

[13] X. Wu, B. Mukherjee, and S.-H. G. Chan, "MACA: an efficient channel allocation scheme in cellular networks," in *Proceedings of IEEE Globecom'00*, San Francisco, CA, Nov. 27–Dec. 1 2000, pp. 1385–1389.

[14] X. Wu, S.-H. G. Chan, and B. Mukherjee, "MADF: a novel approach to add an ad-hoc overlay on A fixed cellular infrastructure," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, Sept. 23–28, 2000, pp. 549–554.

[15] M. Tran and W. Tavanapong, "On the design, analysis, and implementation of a generalized periodic broadcast server," *IEEE Trans. Broadcasting*, 2006.

[16] W.-F. Poon, K.-T. Lo, and J. Feng, "Provision of continuous VCR functions in interactive broadcast VoD systems," *IEEE Trans. Broadcasting*, vol. 51, pp. 460–472, Dec. 2005.

[17] S.-H. Chan and S.-H. Yeung, "Broadcasting video with the knowledge of user delay preference," *IEEE Trans. Broadcasting*, vol. 49, no. 2, pp. 150–161, June 2003.

[18] ——, "Client buffering techniques for scalable video broadcasting over broadband networks with low user delay," *IEEE Trans. Broadcasting*, vol. 48, no. 1, pp. 19–26, Mar. 2002.

[19] D. Morris and A. H. Aghvami, "A novel location management scheme for cellular overlay networks," *IEEE Trans. Broadcasting*, vol. 52, pp. 108–115, Mar. 2006.

[20] J. G. Apostolopoulos, "Error-resilient video compression through the use of multiple states," in *Proceedings of IEEE ICIP*, Sept. 2000, vol. 3, pp. 352–355.

[21] A. R. Reibman, H. Jafarkhani, Y. Wang, M. T. Orchard, and R. Puri, "Multiple description coding for video using motion compensated prediction," in *Proceedings of IEEE ICIP*, Oct. 1999, pp. 837–841.

[22] V. A. Vaishampayan and S. John, "Balanced interframe multiple description video compression," in *Proceedings of ICIP*, Oct. 1999, vol. 3, pp. 812–816.

[23] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *Proceedings of Network Protocols*, Nov. 2003, pp. 16–27.

[24] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, "On multiple description streaming with content delivery networks," in *Proceedings of IEEE INFOCOM*, June 2002, pp. 1736–1745.

[25] S. Wee, J. Apostolopoulos, W. tian Tan, and S. Roy, "Research and design of a mobile streaming media content delivery network," in *Proceedings of IEEE International Conference on Multimedia and Expo*, July 2003, vol. 1, pp. 5–8.

[26] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proceedings of ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems*, 2000, pp. 61–68.

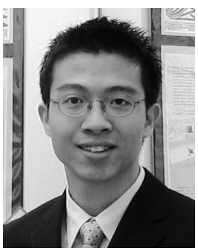
[27] G. Calinescu, I. I. Mandoiu, P.-J. Wan, and A. Z. Zelikovsky, "Selecting forwarding neighbors in wireless ad hoc networks," *ACM Mobile Networks and Applications*, vol. 9, no. 2, pp. 101–111, Apr. 2004.

[28] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *Proceedings of Hawaii International Conference on System Sciences (HICSS)*, Jan. 2002, pp. 3866–3875.

[29] W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of Mobile and Ad Hoc Networking and Computing*, Aug. 2000, pp. 129–130.

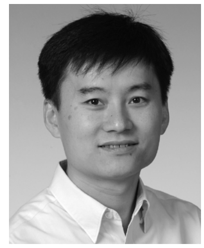
[30] ——, "AHBP: an efficient broadcast protocol for mobile ad-hoc networks," *Journal of Science and Technology (JCST)*, vol. 16, pp. 114–125, 2001.

[31] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing*, vol. 5, no. 2, pp. 193–204, Apr. 2002.
[32] F. Kuhn and R. Wattenhofer, "Constant-time distributed dominating set approximation," in *Proceedings of ACM Int. Symposium on the Principles of Distributed Computting (PODC)*, 2003, pp. 25–32.
[33] D. Lichtenstein, "Planar formulae and their uses," *SIAM Journal on Computing*, vol. 11, no. 2, pp. 329–343, 1982.



Man-Fung Leung received the B.Eng. and M.Phil. degrees in computer science from The Hong Kong University of Science and Technology, Hong Kong, in 2004 and 2006, respectively.

His research interests are in wireless peer-to-peer networks and multimedia networking.



S.-H. Gary Chan (S'89–M'98–SM'03) received MSE and PhD degrees in Electrical Engineering from Stanford University, Stanford, CA, in 1994 and 1999, respectively, with a minor in business administration. He obtained his B.S.E. degree (highest honor) in Electrical Engineering from Princeton University, Princeton, NJ, in 1993, with certificates in applied and computational mathematics, engineering physics, and engineering and management systems.

He is currently an Associate Professor with the Department of Computer Science and Engineering and Director of Computer Engineering Program, The Hong Kong University of Science and Technology, Hong Kong. He is also an Adjunct Researcher with Microsoft Research Asia in Beijing. He was a Visiting Assistant Professor in networking at the Department of Computer Science, University of California at Davis, CA, from 1998 to 1999. During 1992-93, he was a Research Intern at the NEC Research Institute, Princeton, NJ.

Dr. Chan served as a Vice-Chair of IEEE COMSOC Multimedia Communications Technical Committee (MMTC) from 2003 to 2006, and is currently a Vice-Chair of Peer-to-Peer Networking and Communications Technical Sub-Committee, IEEE Comsoc Emerging Technologies Committee. He is a guest editor for special issues on "Peer-to-Peer Multimedia Streaming" in IEEE Communication Magazine (2007) and "Advances in Consumer Communications and Networking" in Springer Multimedia Tools and Applications (2007). He has been a co-chair of multimedia symposium in IEEE Globecom (2007 and 2006) and IEEE ICC (2007 and 2005), and for the workshop on "Advances in Peer-to-Peer Multimedia Streaming" in ACM Multimedia Conference (2005).

Dr. Chan is a member of Tau Beta Pi, Sigma Xi, and Phi Beta Kappa. He was a William and Leila Fellow at Stanford University in 1993-94. At Princeton, he was the recipient of the Charles Ira Young Memorial Tablet and Medal, and the POEM Newport Award of Excellence in 1993. His research interest includes multimedia networking, peer-to-peer technologies and streaming, and wireless communication networks.