Hindawi Publishing Corporation
Advances in Multimedia
Volume 2008, Article ID 273187, 9 pages
doi:10.1155/2008/273187

Research Article

A Collaborative Wireless Access to On-Demand Services

Zohar Naor

Department of Mathematics Physics and Computer Science, Faculty of Science and Science Education, University of Haifa at Oranim Academic College of Education, 31905 Haifa, Israel

Correspondence should be addressed to zohar Naor, zohar@math.haifa.ac.il

Received 28 April 2007; Revised 1 September 2007; Accepted 12 November 2007

Recommended by Jianwei Huang

A collaborative access scheme that exploits the broadcast nature of the wireless communication in order to achieve multicast content delivery is presented in this paper. The key idea is that individual clients requesting for the same content can collaborate and share the same data channel. As opposed to broadcasting, this method enables the clients to determine online the delivered content, and thus supports on-demand services. On the other hand, a multicast content delivery is much more efficient than a unicast content distribution, which must use a dedicated data channel per each and every client. This method is particularly suitable for sessions having a long-time duration, for applications in which clients can subscribe to ahead of time, and for applications in which the clients receive the same information simultaneously. A multicast content distribution increases the network service throughput in terms of the expected number of clients served simultaneously, and therefore it offers a reduced waiting time for content delivery at highly loaded time periods. It is shown that the problem of maximizing the efficiency of distributing a content in a wireless network is NP-hard. An approximation algorithm is therefore used, that for any $0 < \varepsilon < 1$ finds an approximation solution with a relative accuracy ε. The proposed method does not require any hardware modification on the network equipment. Thus, it can be easily implemented.

1. INTRODUCTION

The concept of collaborative wireless networking is emerging as a promising technology to enhance system performance by sharing wireless resources among the demanding clients. For instance, sharing antennas to enable a virtual multiple-input multiple-output (MIMO) is an example for a collaborative scheme. The focus in this study is to use an application layer collaborative method in order to establish a virtual MIMO system. Video applications, such as near video-on-demand (NVOD) service, are characterized by sessions having a relatively long-time duration, and a relatively high arrival rate of clients requesting for the same data. Consequently, even a moderate demand for video applications may potentially block the wireless channel, unless a collaborative scheme is used, that enables bandwidth sharing among different clients. The traditional content distribution methods were originally designed for wired networks. As such, they do not consider the wireless channel conditions. For existing wireless networks, even a moderate demand for video streaming can easily exceed the system available bandwidth. For this reason, the approach adopted by manufactures of wireless communication equipment for advanced content distribution over wireless networks is to build dedicated broadcast networks for what is known as "mobile TV." These networks are expected to use a new infrastructure, based on either *DVB-H* [1] or *MediaFlo* [2] standards. Unfortunately, these schemes do not support on-demand services. Whereas unicast multimedia streaming is used for mobile TV today, broadcast extensions to mobile networks like 3GPP *MBMS* and 3GPP2 *BCMCS* are under standardization. The contribution of this paper is a formulation of the content distribution problem as an optimization problem that is proved to be NP-hard, and the development of an efficient, yet reliable scheme that enables an on-demand content distribution.

Traditionally, there are three representative approaches for content distribution: media servers replication, using existing proxies for media data buffering, and constructing peer-to-peer overlay networks, dedicated to content distribution. These schemes were designed mostly for wired networks and do not consider the special characteristics of the wireless channel.

Building mirror sites is a very efficient solution. Unfortunately, since the mirror sites are core nodes, they are not aware of the conditions on the wireless channel. Moreover, multicast-based content delivery is very hard to be implemented by core nodes. For this reason, no large-scale commercial service supports multicast content distribution at the Internet backbone.

The proxy-based approach for content distribution is to use the memory of existing proxies for caching media data. The main drawback of this approach is that due to the huge memory size of a typical video file, the proxy memory cannot support a scalable service, and there is a need to use special machines, as it is done by cable TV companies. In order to overcome this problem, segment-based proxy caching strategies were proposed [3], to cache partial segments of the media objects instead of their entirety. Yet, the memory limitation still restricts the service scalability, unless a dedicated spacial machine is used. Most importantly, the proxy is not aware of the variable conditions of the wireless channel.

The idea to construct a peer-to-peer overlay network, dedicated to content distribution, is very popular across the Internet [4]. The key idea of this method is that each node in the overlay network forwards the data it is receiving, and serves other nodes as a server. The usage of an overlay peer-to-peer network that uses a proxy system supported by peer-to-peer networks was proposed in [5]. This approach is very effective when applied in wired networks. Its main drawback is that when nodes are leaving the group, the nodes that remain in the overlay network may suffer from service disconnection. Recovery algorithms [6] were proposed to recover from this situation. Unfortunately, the dynamics of group membership in a wireless network is much greater than that experienced in a wired network, due to the client mobility. Moreover, a wireless channel is severely vulnerable to phenomenons such as multipath fading, interference, base station reselection, and so forth. The outcome of these phenomenons is that a wireless channel may suffer from high and variable round trip time, rate fluctuations, link outage, and occasional burst losses. For these reasons, the approach of peer-to-peer overlay network used in the Internet is not recommended for a wireless network, due to the mobility of the participating peers and the instability of the peer-to-peer connection for a long-time period.

The studies cited above focus on the wired network, and do not consider the special characteristics of the wireless channel. Hence, there is a need to develop a strategy that can cope with the wireless channel conditions in a more efficient way, that exploits the broadcast nature of the wireless communication. It would be desirable to use this property to utilize the scarce wireless bandwidth more efficiently.

1.1. The contribution of this work

The main contribution of this work is the usage of the access point to the wireless network as a proxy server. Thus, independent clients requesting for the same content can collaborate, and share the same data channel. Traditionally, the access point to the wireless network (e.g., a base station in a cellular network) is used only for transmission. The suggested method requires a cross-layer architecture, aimed to reduce redundant traffic and thus to increase the network capacity, in terms of number of clients served simultaneously. Such an architecture can be applied on a metropolitan WiMAX network, in which the coverage area is sufficiently large such that many clients are expected to receive the same video stream simultaneously. The key idea is to exploit the broadcast nature of the wireless communication in order to create a mechanism that enables to transmit data requested by many users residing in the same area only once. This mechanism is achieved by an addressing method that labels each request for data, using the application type, the encoding method, and the data required as the keys. Consequently, independent requests for the same data initiated from the same area are accumulated in order to form a virtual multicast group. A Virtual Multicast Group (VMG) is an ad hoc multicast group, that contains all the users residing in the same area and waiting for the same data at the same encoding method. Since the task of forming VMGs is very hard to be implemented by core nodes, it is performed at the access portion, that is, by the wireless network. For this reason, this method can be integrated with previously published methods for content distribution [3–5].

Content distribution can be potentially performed either by the content provider, or by the clients themselves [4, 5], or by an intermediate node. The first two methods were originally designed for wired networks. As such, they do not consider the wireless channel conditions. They are vulnerable to phenomenons such as high and variable round trip time, rate fluctuations, occasional burst losses, and instability of the client-network physical connection for a long-time period. Hence, these methods are not optimized for a high quality of service content distribution over wireless networks. Most importantly, whenever the bandwidth required to satisfy the content demands directed to different content providers exceeds the available system bandwidth, only the wireless network can select which content is delivered first. The VMG method is based on an application-layer module, to be installed in an intermediate node, located at the gateway between the wireless network and the wired backbone. This method is especially suitable for applications that are either delay insensitive, or that the clients receive the information simultaneously. Examples for such applications are live video streaming, NVOD service, and download requests of popular video files. Due to the long-time duration of a typical video session, and its wide bandwidth, even a moderate demand for such applications can easily exceed the available bandwidth offered by a conventional wireless system. Hence, a bandwidth sharing method must be used under realistic load conditions. The expected waiting time for a service offered by the VMG method is in the worst case (under low load conditions) the same waiting time offered by the conventional methods that use a unicast content distribution. As the demand for content increases, the superiority of the VMG method increases exponentially, in comparison to the conventional unicast-based methods. Under realistic load conditions (i.e., similar to the demand experienced for NVOD service), only a VMG method can cope with an intensive demand for the same content. Another important feature is that an implementation of the VMG scheme requires only minor software modifications, mainly on the network equipment. Only a small modification is required on the user equipment.

Recently it has come to our attention that the idea to use the access point as a proxy server has been independently proposed in [7]. However, the system considered in [7] is a WLAN, while this paper considers a wireless network such as a WiMAX network, which is used for content distribution. The basic idea proposed in [7] is to cache the recently transmitted downstream data in the AP for a possible future use. This data may be reused when a node attempts to *transmit* data to a peer node. In this paper, the collaboration is conducted when neighboring nodes attempt to *receive* the same information. Thus, the collaboration mechanism is totally different from the one presented in [7].

1.2. Paper organization

The structure of the rest of this paper is as follows. Model and problem formulations are given in Section 2. The concept of multicast content distribution is introduced and analyzed in Section 3. Performance analysis and simulation results are provided in Section 4. A summary and concluding remarks are given in Section 5.

2. MODEL AND PROBLEM FORMULATION

This work considers a metropolitan WiMAX access network, that provides on-demand multimedia services. The network is a broadband wireless access network for stationary, or relatively slow, clients sharing the same access point (AP). The information is delivered to the clients through a downlink data channel, shared among the clients residing within the AP coverage area. This channel is controlled by the system, that allocates the bandwidth to the demanding clients. The wireless bandwidth is assumed to be sufficient to support multiple requests simultaneously, and to use statistical multiplexing in order to transmit multiple variable bit rate transmissions over one broadband channel. The time slot allocated to each subscriber by the IEEE 802.16 protocol can be enlarged in order to cope with variable bit rate transmissions. The clients are assumed to be stationary, in the sense that during the data retrieval session the user location is fixed.

The amount of data, measured in bytes, required to retrieve a content depends on the content size in bytes, on the receiving device, and on the quality of the user-network connectivity. The user receiver determines the compression standards (e.g., *MPEG*-2), and the user-network connectivity affects the packet loss probability through the session. Each personal content demand is associated with a priority R, defined as the network revenue associated with this content multiplied by the user waiting time for a service, and a cost M', defined as the information amount in bytes that must be transmitted. M' depends on M, the memory size of the required content, the compression and encoding standard used, and on the packets loss during the session (i.e., on the user-network connectivity). The *normalized priority* ϕ is defined by

$$\phi = \frac{R}{M'} = \frac{tr}{M'}, \qquad (1)$$

where r is the network revenue associated with this demand and t is its waiting time. Most service providers charge a fixed price per content type (e.g., the same price for a movie). Hence, ϕ depends on R and M', which depends on the content size M, the encoding method, and the user-network connectivity.

It is assumed that the personal content demands initiated by the clients are generated according to a Poisson process with a mean λ arrival per time unit. The time duration of a session is exponentially distributed with a mean $1/\mu$ service time. The *load factor* ρ is defined as

$$\rho = \frac{\lambda}{\mu}. \qquad (2)$$

The *service throughput* is defined as the expected number of personal content demands satisfied by the network per time unit. The network goal is to maximize the service throughput, without starving demands for unpopular content. Note that new content demands arrive online. Since certain personal content demands may have a higher priority than other demands (e.g., due to a longer waiting time), the problem of maximizing the service throughput can be generalized as specified below. The *weighted service throughput* is defined as the sum of priorities over all personal content demands satisfied by the network, that is, given a set of U clients, and a set of B content demands $\mathcal{B} = \{b_1, \dots, b_B\}$, where each content demand b_i has (a) a priority P_i that depends on the number of users associated with this demand, on their waiting time, and on the required data and service $P_i = \sum_i R_i$ and (b) a weight ψ_i, that reflects the required download rate, measured in bits per time unit. The maximum *weighted service throughput* problem is defined as follows: given a set of B content demands, the goal is to maximize the achievable *weighted service throughput* Θ, under the condition that the system bandwidth capacity is bounded from above by a predefined constant C. That is, the goal is to select a subset $B' \subseteq B$ of content demands, such that the sum

$$\Theta = \sum_{j \in B'} P_j \qquad (3)$$

is maximal, under the condition that the sum ψ of the weights of the content demands satisfied by the network is bounded from above by the network bandwidth capacity C. That is,

$$\psi = \sum_{j \in B'} \psi_j \leq C, \qquad (4)$$

where Θ is defined as the *weighted service throughput*. Equation (4) is the constraint that must be satisfied by the optimal assignment defined in (3). It is assumed that a request for a content is transmitted through an up-link signaling channel, and the acknowledge message is transmitted by the local server through a down-link control channel. Clearly, the problem of maximizing the service throughput is a special case of the maximum weighted service throughput problem. From now on we therefore consider only the problem of maximum weighted service throughput.

Theorem 1. *The problem of maximum weighted service throughput is NP-hard.*

Proof. Given a group of n bandwidth demands, the goal is to maximize the weighted service throughput. We now prove that this problem is NP-hard. The reduction is from the knapsack problem which is known to be NP-hard [8]. The input to the knapsack problem is a set of n elements. Each element has a size s_i and a weight w_i, where s_i and w_i are positive integers. The goal is to maximize the sum of weights of elements, under the condition that the sum of sizes is bounded from above by C, where C is a pre-defined constant, known as the knapsack capacity. That is, to find a subset of the elements such that $\sum s_i \leq C$ and $\sum w_i$ is a maximum.

Given a knapsack problem with n elements and a knapsack with a capacity of C units, we consider the following *maximum weighted service throughput* problem: the input to the *maximum weighted service throughput* problem is a system consisting of n bandwidth demands and a bandwidth capacity C. For each element i given as an input to the Knapsack problem, with a size s_i and a weight w_i, we associate a bandwidth demand with a priority w_i and a required bandwidth s_i. Note that the term "weight" has a different meaning by each problem: the weight w_i in the original Knapsack problem is associated with the priority P_i of the equivalent bandwidth demand in the *maximum weighted service throughput* problem, while the size s_i in the original Knapsack problem is associated with the weight (i.e., the required bandwidth) of the equivalent ith bandwidth demand in the *maximum weighted service throughput* problem. The assignment strategy that maximizes the *weighted service throughput* is the optimal assignment for the original knapsack problem. Clearly, the opposite direction is also true—and any *maximum weighted service throughput* problem is essentially a 0/1 knapsack problem (i.e., each element can be either assigned or not assigned to the knapsack). Hence, given a knapsack problem, an optimal assignment can be always found by a linear reduction to the equivalent *maximum weighted service throughput* problem, and vice versa. □

3. MULTICAST CONTENT DISTRIBUTION

The key idea of the proposed method is to establish an efficient mechanism for a multicast transmission to multiple points residing in the same area. Since the problem of optimal content distribution is NP-hard, a fast approximation algorithm is used, which can achieve any desired approximation ratio. To achieve this goal, the *VMG* server handles a queue of content demands. A unique task identification (ID) is assigned to each content demand, depending on the required service type, the required data, and the expected bandwidth (different devices requesting the same content may deserve/capable for a different download rate). The VMG server handles a *task table*. Each task has a unique entrance in this table, to be determined by its ID. The list of all its associated clients is attached to each task. Upon receiving a new content demand, the VMG server computes its unique task ID and checks if this ID already exists in the *task table*. If it does not exist, the demand is forwarded *immediately* to the wired backbone for data retrieval. On the other hand, if this ID already exists, that is, a request to retrieve this content has been already sent (by another client), there is no need for another content retrieval. The new client just joins to the list of clients requesting for this data and waiting for a reply. When the data is available, the VMG server distributes the data efficiently to the clients, using a single data channel. This distribution mechanism is achieved simply by using the VMG (i.e., the task) ID as the addressee. Hence, the expected waiting time for data retrieval can be only reduced, in comparison to the conventional content distribution methods, while the bandwidth utilization can be only improved. Since the VMG server distributes the data, it must also handle the transcoding problem. This problem is handled by retrieving the same content only once, then converting the code to the required formats, using different channels for the different formats. Hence, while the same data is retrieved only once, the VMG server may assign multiple channels to the same content whenever it is required. In these cases, each channel is associated with a different VMG. That is, the term VMG considers only one (multicast) channel, shared among multiple clients. Due to the transcoding problem, the same content can be potentially transmitted through multiple channels.

The basic mechanism that enables an AP to transmit data directed to multiple points only once is based on the broadcast nature of the wireless channel, and on the bandwidth allocation strategy of the IEEE 802.16 protocol. The signaling data is transmitted through a shared downlink channel controlled by the system. In order to receive data, each subscriber must track its own unique ID, and to response only to the messages directed to him/her. The proposed distribution mechanism is that each client tracks its relevant VMG ID, in addition to its own personal ID. The AP allocates the same data channel to all the members of the same VMG. Consequently, privacy and security are provided since the signaling is conducted individually. The data is provided only to the VMG members, since the date channel is allocated only to them. On the other hand, this data channel is shared among all the VMG members. Thus, redundant traffic is avoided since the same information is transmitted only once.

The proposed method can potentially increase the utilization of both the wireless links, as well as the wired links associated with the wireless network. Since data requested by many users residing in the same area is transmitted only once, the bandwidth efficiency of the wireless links can be only increased. In addition, since many requests for the same data are accumulated by the VMG server and sent to the wired backbone as a single request, also the bandwidth efficiency of the wired portion of the network should increase. In many applications, such as NVOD and live video streaming, the data is received simultaneously by many users residing in the same area.

3.1. The definitions of priority and cost

Given the condition that n users request the same data denoted by k, the priority P^k of the VMG associated with k is given by

$$P^k = \sum_{i=1}^{n} R_i^k = \sum_{i=1}^{n} r_i^k t_i^k, \quad (5)$$

where R_i^k is the priority associated with the personal content demand of the ith user for the data k, r_i^k is the network revenue from this personal demand, and t_i^k is the waiting time of this personal demand. The *priority density* π_k of the content demand k, that reflects the network priority per bandwidth unit associated with this demand, is defined by

$$\pi_k = \frac{P^k}{\psi^k} = \frac{\sum_{i=1}^{n} R_i^k}{\psi^k}, \quad (6)$$

where ψ^k is the cost, that is, the required download rate of k, given in (10).

3.2. The content distribution algorithm

The content distribution algorithm is as follows.

(1) Assign a priority and weight, and compute, using (6), the *priority density* for each bandwidth demand.
(2) Sort the bandwidth demands in a nonincreasing order by their *priority density*. For demands having the same *priority density*, sort in a nondecreasing order by their expected time duration T_i (i.e., the shorter request is selected first, in order to reduce the expected waiting time for a service).
(3) Assign the bandwidth demands using the rule that the most valuable demand (i.e., the demand with the larger *priority density*) comes first. Whenever a bandwidth demand b_i is assigned, the system available bandwidth capacity C is reduced by the *weight (rate)* w_i for the next T_i time units, and the *service throughput* Θ, given in (3), is increased by P_i. In case of a variable rate it is assumed that $w_i(t)$ is given for $0 \leq t < T_i$. *Note that preemption is not allowed*. This process continues until either there is no available system bandwidth, or all the bandwidth demands are assigned. If either all the bandwidth demands are assigned, or all the system bandwidth is used, do not continue to the next step; the assignment algorithm is successfully completed, and the assignment is optimal. Otherwise, continue to the next step.
(4) Apply a nearly optimal assignment algorithm, with a relative precision ϵ, where ϵ is a predefined constant.

The proposed approximation algorithm is based on the fast approximation algorithm for the Knapsack problem proposed by Ibarra and Kim [9]. The key idea is to partition the bandwidth demands into small and large items. The weights of the large items are scaled, and a dynamic programming is used to find an optimal solution to the problem with scaled weights and capacity. Afterwards, the small items are added by a greedy algorithm as follows: Let S' denotes the *service throughput* assigned by the third step of the assignment algorithm. We define a normalized factor $\delta = S'(\epsilon/3)^2$. The approximation algorithm creates a table whose length is $\lfloor (3/\epsilon)^2 \rfloor$, that contains elements that the profit of each one of them is less than or equal to $S'\epsilon/3$. A dynamic programming is used to maximize the *service throughput* without violating the bandwidth capacity C, by adding elements from the table to the list of bandwidth demands already assigned. A detailed description of the algorithm of Ibarra and Kim can be found in [9]. Many theoretical and practical papers have addressed the issue of an approximation solution to the 0/1 Knapsack problem [10, 11]. Some of them have a better worst case running time and space complexity. However, these algorithms outperforms the approximation algorithm of Ibarra and Kim [9] only for a very large input. The number n of different bandwidth demands arriving to the same VMG server during a relatively short time period (the expected waiting time of a bandwidth demand) is not expected to be that large to justify a relatively complex and nontrivial approximation algorithm, as these algorithms cited above. For this reason, the performance of the approximation algorithm used in this study is expected to be superior to alternative methods, due to its simplicity and relative accuracy. For most practical cases, the number of different bandwidth demands handled by the VMG server is expected to be sufficiently small, such that an optimal solution can be easily derived by the proposed assignment algorithm.

3.3. Implementation considerations

The concept of VMGs can be implemented by a software module, referred to as the *VMG server*. A VMG server is an application layer module, to be installed at the gateway between the wireless network and the wired backbone. In order to handle privacy and security issues, this gateway is defined as a proxy server. The VMG server provides services such as live video streaming, NVOD, download of video and music files, multicast, and a video conference call. There is no need for a significant modification of the user equipment, but the capability to track the task ID (i.e., the VMG ID), in addition to its own user ID. In addition, whenever a user initiates a bandwidth demand, he/she must update its location upon every movement to another AP. The operation of reselecting an AP must lead to leave-multicast operation at one VMG, and join-multicast operation at another VMG. If either VMGs has only one client, then the VMG server should either delete the VMG (if its only client leave the group), or create a new VMG (if a new client attempts to join an empty VMG). From the user point of view, no hardware modification is required. Multiple download requests for the same file or service should be sent from the VMG server to the wired IP network as a single request, created by a virtual user (i.e., the VMG server). The task ID is used as the ID of the virtual multicast group. Once the data is retrieved, the VMG server has to distribute this data to the list of clients waiting for a reply. Using the generic multicast mechanism already provided by wireless networks, the VMG server application maps the clients to the multicast groups.

3.4. Transmission errors recovery

A wireless channel is likely to suffer from phenomenons such as high and variable round trip time, rate fluctuations, link outage, and occasional burst losses. The main result of these phenomenons is a loss of packets. Consequently, a wireless channel is in general less reliable than a wired channel. For this reason, packets loss must be considered. This section considers the recovery mechanism which handles packets loss. Since it is assumed that users can move only between sessions, but not during a session, it is assumed that the packet loss probability of a user u_i during the session is a constant p_i $(0 < p_i < 1)$. It is assumed that p_i can be estimated for a good approximation (e.g., using the signal to noise ratio, SNR, and/or historical data). The expected number of transmissions required for a successful transmission is therefore given by

$$\sum_{k=1}^{l} p_i^{k-1}(1-p_i)k < \sum_{k=1}^{\infty} p_i^{k-1}(1-p_i)k = \frac{1}{1-p_i}, \quad (7)$$

where l is the maximum number of re-transmissions allowed by the access protocol. In IEEE 802.11 a typical value of l is 7. Hence, the actual amount of data required to receive D bytes is bounded from above by $D/(1-p_i)$. The bandwidth required to receive the same information during the same time interval is therefore

$$\psi_i = \frac{\psi_0}{1-p_i}, \quad (8)$$

where ψ_0 is the bandwidth required for $p_i = 0$. Hence, the cost of satisfying the personal content demand initiated by u_i depends also on the network connectivity of u_i. Given the receiving device of u_i and the required content, the download rate ψ_0 is determined at purchase time. The actual download rate ψ_i must take into consideration the expected packets loss. Given a VMG (i.e., the group of users attached to the same content demand and having the same ideal download rate ψ_0), the multicast download rate is determined as follows: The first client who initiates the content demand set the multicast download rate to be equal to its own download rate ψ_i given in (8), and the packet loss probability plp of the *VMG* to be equal to its own packet loss probability. Every additional client u_j with an estimated packet loss probability p_j increases the *VMG* plp by $p_j(1-\text{plp})$:

$$\text{plp} = \text{plp} + (p_j - p_j\text{plp}) = \text{plp} + p_j(1-\text{plp}). \quad (9)$$

Using the same analysis that lead to (8), the *VMG* required download rate is given by

$$\psi_{\text{vmg}} = \frac{\psi_0}{1-\text{plp}}. \quad (10)$$

3.5. Motivation

The motivation for using the VMG mechanism depends on the load on the shared data channel. Assuming a Poisson arrival rate with a mean λ arrival per time unit, and an exponentially distributed with a mean $1/\mu$ service (i.e., session) time, a conventional content distribution server behaves as an $M/M/1$ system. The expected waiting time for data retrieval is given by (see, e.g., [12])

$$W = \frac{\rho/\mu}{1-\rho}. \quad (11)$$

It follows from (11) that the waiting time grows exponentially with the *load factor* $\rho = \lambda/\mu$ [12]. As either the arrival rate or the service time increases, so does the waiting time. Video applications are characterized by a relatively long session time. For example, a typical time duration of live streaming of a movie is about 90 minutes. Moreover, the arrival rate of demands for popular data (e.g., popular video files, news broadcast) is also expected to be relatively high. Hence, even a moderate demand for video services can easily jam the wireless channel, unless some sharing method is used, that enables reusability of the wireless channel. As the arrival rate for a specific content is sufficiently high, whenever the time interval between two consecutive arrivals of requests to the same content is smaller than the waiting time of the first arrival, the second request for that content can just join to the first request and they can both share the same data channel, thus reducing their average waiting time. Given a rate of λ arrivals per time unit of demands for the same content, the expected time interval between two consecutive arrivals is $1/\lambda$. Upon a new arrival, the condition under which its expected waiting time for a service is longer than the expected waiting time for a new arrival of a request for the same content is given by

$$W > \frac{1}{\lambda}. \quad (12)$$

To simplify the analysis, it is assumed that only one stream is served. Substitute W from (11), then we get that the condition under which a new demand for the same content is expected to join the first demand while it is still waiting for a service is given by

$$W = \frac{\rho/\mu}{1-\rho} > \frac{1}{\lambda}. \quad (13)$$

Multiply both sides by λ and substitute $\rho = \lambda/\mu$, then we get that

$$\frac{\rho^2}{1-\rho} > 1, \quad (14)$$

which leads to

$$\rho^2 + \rho - 1 > 0. \quad (15)$$

The value of ρ which satisfies (15) and the condition $\rho \geq 0$ is

$$\rho > \frac{\sqrt{5}-1}{2} \approx 0.618. \quad (16)$$

Equation (16) implies that whenever the *load factor* satisfies the condition $\rho > 0.618$ a batch service outperforms an individual service. That is, whenever $\rho > 0.618$, the expected time duration between two consecutive arrivals is shorter than the expected waiting time for a service. Thus, a new arrival is likely to join a previous arrival. Consequently, the expected waiting time for a service is reduced. It follows from (16) that the superiority of bandwidth sharing increases with the arrival rate for this content, and with the expected service time and waiting time. As the number of popular files/streams increases, the service rate increases to $E[s]\mu$, where $E[s]$ is the expected number of users sharing the same data channel. The expected waiting time for an FIFO service policy is therefore

$$W_F = \frac{\rho/E[s]\mu}{1-\rho/E[s]} \leq W, \tag{17}$$

where equality holds if, and only if, $E[s] = 1$. That is, the *load factor* ρ is sufficiently small such that no bandwidth sharing is used.

4. PERFORMANCE ANALYSIS AND SIMULATION RESULTS

In this section the performance of the proposed VMG method is compared to the performance of the conventional method currently used by existing wireless networks that uses a unicast content distribution. Note that it follows from (9) that the bandwidth required for multicast content distribution is always less than or equal to the bandwidth utilization of the conventional content distribution that uses unicast.

Theorem 2. *Given a constant number of content items (i.e., VMGs), the waiting time of any personal demand served by an FIFO scheduling is bounded from above by a predefined constant.*

Proof. Let n be the number of supported VMGs. Let T_{all} be the total service time of all the VMGs:

$$T_{\text{all}} = \sum_{i=1}^{n} t_i, \tag{18}$$

where t_i is the expected service time of the ith VMG. Then, under the worst case condition, a content demand arrived just after starting the download of its associated VMG, and cannot join to an ongoing session, has to wait to all the other $n-1$ VMGs to be served, as well as to its own VMG to be served. Hence, the waiting time is bounded from above by the time duration required to serve these n VMGs. This time period is bounded from above by T_{all}, and this bound is not tight whenever some of the VMGs can be served simultaneously. □

Let us consider a system in which content demands for a particular content are generated according to a Poisson process with a mean λ arrivals per time unit. Given that a new arrival has to wait Δt time units for a service, the probability of a new arrival during the waiting time of this demand, of another demand for this content is given by

$$P_{\text{batch}} = 1 - e^{-\lambda \Delta t}. \tag{19}$$

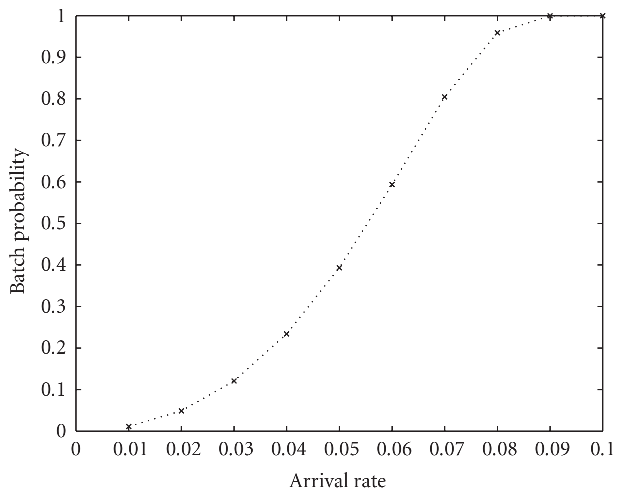

Figure 1: The batch probability of creating a VMG containing at least two collaborating clients, as a function of the expected arrival rate.

Using the individual service policy currently used, the expected waiting time is given in (11). Substitute the expected waiting time for Δt in (19), then we get the value of P_{batch} as a function of λ and μ. Figure 1 depicts the batch probability of at least two collaborating clients served simultaneously as a function of the arrival rate for the same specific content. Only this content is downloaded, and the download time is exponentially distributed with a mean of 10 time units. It is clearly demonstrated that even for a relatively low arrival rate of 0.1 arrival per time unit, the probability of accumulating at least two clients in the same VMG is practically 1. For instance, if the expected download time of a video file is 10 minutes, even for one arrival per 18 minutes the probability of having $n_i > 1$ is greater than 0.5. As the arrival rate approaches to one arrival per 10 minutes the expected waiting time of an $M/M/1$ system that uses a unicast content distribution actually explodes, and a batch service must be used. As the download time increases, so does ρ, the motivation for using multicast content distribution increases. For instance, if the expected time duration of a session is 30 minutes, a typical value for a popular TV show, or news broadcast, then even for one arrival per 54 minutes the probability of having more than one client at the same multicast group is greater than half.

4.1. Simulation results

Figure 2 depicts the expected waiting time for content delivery, for multicast content distribution and for unicast content distribution, as a function of the arrival rate to the system. The content demands are generated according to a Poisson process, and are equally distributed among 30 files. The download time of a file is exponentially distributed with a mean of 20 minutes. The system bandwidth capacity can support only 10 download requests at a time. As the arrival rate increases, the expected waiting time of a unicast content distribution increases exponentially, and the superiority

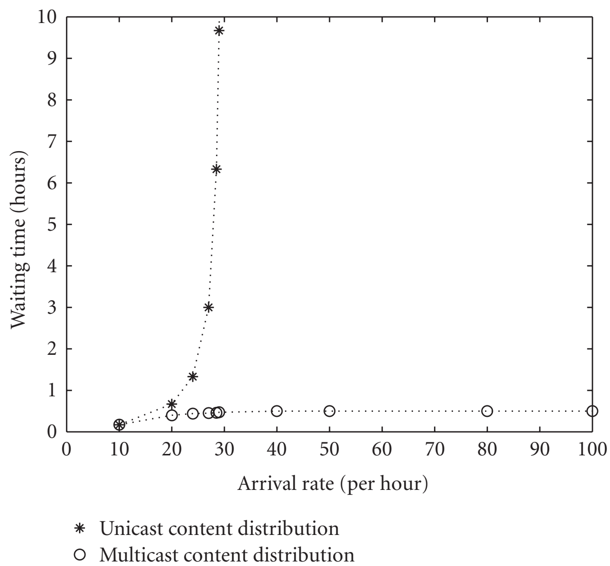

FIGURE 2: The expected waiting time offered by multicast content distribution, and by unicast content distribution methods, as a function of the arrival rate to the system, for a homogeneous load distribution.

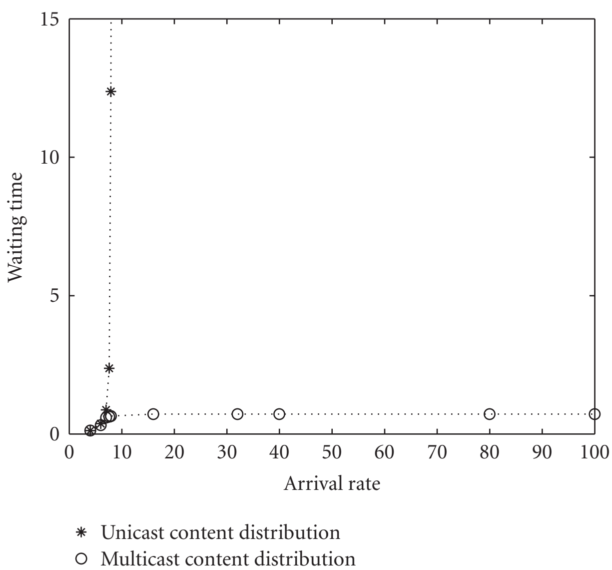

FIGURE 3: The waiting time in hours, as a function of the arrival rate to the system per hour, for a nonhomogeneous load distribution, in comparison to a unicast content distribution.

of a multicast content distribution over a unicast service is demonstrated very clearly.

Figure 2 considers content demands that are equally distributed among different video files. In reality, different video files are requested at different rates. The pattern of arrival rate to video on demand service and video rental stores [13] indicates that the majority (about 60%–80%) of the requests are for a few files. Many simulation experiments were conducted, for different load distribution patterns, and download time. Due to space limitation, two representative results are given below. Figure 3 considers a system that supports 16 different files. 68% of the new arrivals are equally distributed among 4 most popular files, other 20% of the arrivals are equally distributed among other 4 files, and the rest of 12% of the new arrivals are equally distributed among the 8 least popular files. The average download time of each file is 30 minutes, and an AP can support 4 different download requests simultaneously. New arrivals were generated according to a Poisson process, and the download time was exponentially distributed. The arrival rate ranges up to 100 arrivals per hour to an AP. Considering an AP that supports 1000 clients, this arrival rate represents a rating of 10% of the customers per hour, which is a relatively low arrival rate for cable TV and video-on-demand systems. Note that under these conditions, a unicast content distribution explodes as soon as the arrival rate to the AP approaches to 8 arrivals per hour.

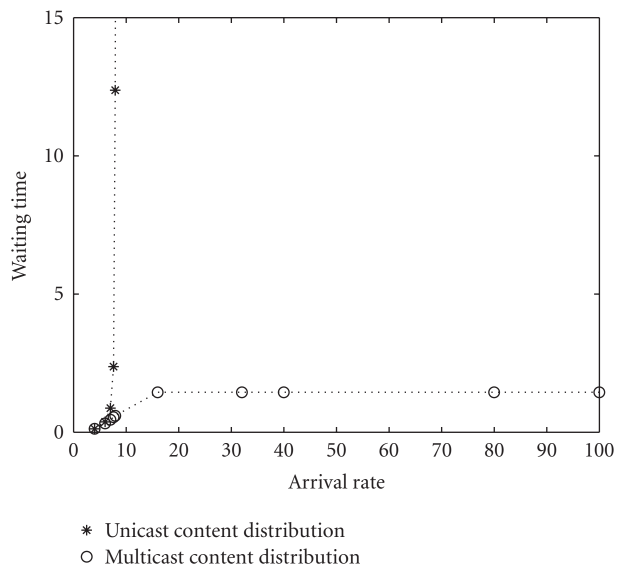

FIGURE 4: The waiting time in hours, as a function of the arrival rate per hour to the system.

Figure 4 considers a system that supports a group of 33 files. 3 popular files are requested by 20% of the clients per each file, while the other 40% of new arrivals are equally distributed among 30 less popular files. The system bandwidth capacity and the expected download time are both the same as in Figure 3. The superiority of multicast content distribution over a unicast content distribution is clearly demonstrated.

Due to the nature of the wireless fading channels, it may not be always possible to assign the same data channel to all the clients requesting for the same stream. Thus, it may not be always possible to assign all the clients requesting for the same content at the same time to a single data channel, since such an assignment may cause a significant variance in the signal-to-noise ratio experienced by different clients. However, since different clients can share the same data file, the memory utilization of the AP must be better. Thus, the expected number of clients sharing the same content at the same time can be used as a reliable measurement tool for the

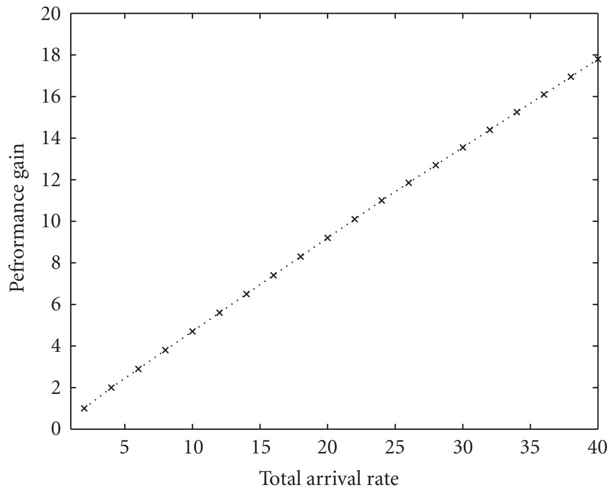

Figure 5: The performance gain, measured by the average number of clients sharing the same downloaded file, as a function of the total arrival rate per hour to a system which serves 100 clients.

performance gain achieved by the batch service. This performance gain can be reflected either by the utilization of the data channel, and consequently reducing the expected waiting time as predicted by (17), or by the AP memory utilization, or both. Figure 5 depicts the average number of clients sharing the same content $E[s]$ given in (17), as a function of the total arrival rate to the system, for an AP which serves 100 clients, and a realistic load distribution [13]. It is clearly shown that even for a moderate demand, the performance gain is significant. A group of 100 clients sharing the same AP can be expected for a WiMAX network which serves few buildings.

5. SUMMARY AND CONCLUDING REMARKS

The main advantage of multicast content distribution over the conventional unicast content distribution is its scalability. Due to the huge bandwidth consumed by video applications, and the relatively long-time duration of a typical session, even a moderate demand for video applications is sufficient to block a conventional wireless channel that uses a unicast content distribution. Hence, a collaborative scheme must be used, in order to enable bandwidth sharing. Under realistic load conditions, a multicast content distribution yields a significantly shorter waiting time than a unicast content distribution. Being implemented at the access portion of the network, the construction of the multicast groups is straightforward.

REFERENCES

[1] http://www.dvb.org.
[2] http://www.qualcomm.com/mediaflo.
[3] S. Chen, B. Shen, S. Wee, and X. Zhang, "Adaptive and lazy segmentation based proxy caching for streaming media delivery," in *Proceedings of the 13th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '03)*, pp. 22–31, Monterey, Calif, USA, June 2003.
[4] S. Ganguly, A. Saxena, S. Bhatnagar, R. Izmailov, and S. Banerjee, "Fast replication in content distribution overlays," in *Proceedings of the 24th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 4, pp. 2246–2256, Princeton, NJ, USA, March 2005.
[5] L. Guo, S. Chen, S. Ren, X. Chen, and S. Jiang, "PROP: a scalable and reliable P2P assisted proxy streaming system," in *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCS '04)*, vol. 24, pp. 778–786, Tokyo, Japan, March 2004.
[6] M. Guo and M. H. Ammar, "Scalable live video streaming to cooperative clients using time shifting and video patching," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 3, pp. 1501–1511, Hong Kong, March 2004.
[7] E. Tan, L. Guo, S. Chen, and X. Zhang, "SCAP: smart caching in wireless access points to improve P2P streaming," in *Proceedings of 24th International Conference on Distributed Computing Systems Workshops (ICDCS '07)*, Toronto, Canada, June 2007.
[8] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, NY, USA, 1979.
[9] O. H. Ibarra and C. E. Kim, "Fast approximation algorithm for the Knapsack and sum of subsets problems," *Journal of the ACM*, vol. 22, no. 4, pp. 463–468, 1975.
[10] H. Kellerer, R. Mansini, U. Pferschy, and M. G. Speranza, "An efficient fully polynomial approximation scheme for subset-sum problem," *Journal of Computer and Systems Science*, vol. 66, no. 2, pp. 349–370, 2003.
[11] M. J. Magazine and O. Oguz, "A fully polynomial approximation algorithm for the 0-1 knapsack problem," *European Journal of Operational Research*, vol. 8, no. 3, pp. 270–273, 1981.
[12] L. Kleinrock, *Queueing Systems*, vol. 1, 2, John Wiley & Sons, New York, NY, USA, 1976.
[13] C. Griwodz, M. Bär, and L. C. Wolf, "Long-term movie popularity models in video-on-demand systems or the life of an on-demand movie," in *Proceedings of the 5th ACM International Conference on Multimedia (MULTIMEDIA '97)*, pp. 349–357, Seattle, Wash, USA, November 1997.