# On Tile Assignment for Region-of-Interest Video Streaming in a Wireless LAN

Guntur Ravindra and Wei Tsang Ooi
Department of Computer Science
National University of Singapore
ravindra@comp.nus.edu.sg, ooiwt@comp.nus.edu.sg

## ABSTRACT

We consider the following problem in this paper: A video is encoded as a set of tiles $T$ and is streamed to multiple users via a one-hop wireless LAN. Each user selects a region-of-interest (RoI), represented as a subset of $T$, in the video to watch. The RoI selected by the users may overlap. Each tile may be multicast or unicast. We define the tile assignment problem as: which subset of tiles should be multicast such that every user receives, within a transmission deadline, the subset of tiles pertaining to the RoI the user selected, while minimizing the number of unwanted tiles received by users. We present and evaluate five tile assignment methods. We show that: (i) minimizing transmission delay can lead to significant wasteful reception in the multicast group, (ii) using tile access probability to assign tiles frequently leads to assignments that violate the deadline, and (iii) a fast, greedy, heuristic works well: it performs close to the optimal method and can always find an assignment within the deadline (as long as such assignment exists).

## Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Video

## General Terms

Algorithms, Performance

## Keywords

Zoomable Video, Region-of-Interest Video Streaming

## 1. INTRODUCTION

Region-of-interest (RoI) streaming is a popular approach for streaming large format images [4] and high-resolution video [7, 8]. Users select a RoI of choice, and the server streams all the data required to render the RoI at the client end. In the case of video streaming, frames may be encoded at multiple resolutions, with the lowest resolution served to the users by default. A user may select a RoI from the lower resolution video to zoom in on and view in higher resolution. The RoI is then cropped from a higher resolution compressed video and sent to the user for display. To support such functionality, the video frames may be encoded as a grid of non-overlapping tiles [13, 5]. Another method is to use overlapping tiles [6] and encode each tile area as a separate video stream. All tiles intersecting with the RoI are cropped and streamed by the server. Representing a video as tiles reduces temporal dependency across frames and allows some degree of random access into RoI.

In this paper, we consider the problem of efficient RoI streaming in a wireless LAN, exploiting the fact that many users may select overlapping RoIs. This scenario arises in a system we are developing, where we want to stream live HD videos to many users equipped with our video player that supports RoI viewing.

While attempts have been made to improve bandwidth efficiency by encoding video based on RoI access pattern [12], reducing bandwidth alone would not suffice as there is a limit to the extent to which bandwidth may be reduced without degrading the video quality. In a one-hop wireless LAN, one could exploit an optimal combination of multicast and unicast capabilities of the access point, in order to transmit overlapping and non-overlapping regions of users' RoIs.

Figure 1 depicts a case where three users have partially overlapping RoIs. The video frame is shown as a grid of tiles with some tiles marked with the different users requiring those tiles. The two tiles required by all the three users (marked "1,2,3") could be multicast. If they are unicast, then these tiles are transmitted three times separately to each of the three users. The figure also shows that two tiles (marked "1,3") are required by only two users. If these two tiles are multicast, then User 2 also receive these two tiles although they are not part of this user's RoI. We term such an unwanted reception as *wasteful reception*. Issue of unwanted reception can be eliminated with users subscribing to advertised multicast groups and the server transmitting chosen tiles over these multicast groups. In the example of Figure 1, tiles marked "1,2,3" could be multicast on one multicast group and tiles marked "1,3" could be multicast on a different multicast group. Although such an approach would improve transmission efficiency, this approach is impractical in a setting where users can frequently change their RoI. The multicast groups would frequently change, new multicast groups would have to be created, and new groups have to be advertised. The ensuing protocol nightmare may be avoided if there is only one predefined multicast group to which all users subscribe, in addition to the unicast port over which video tiles are transmitted by the server. The onus is then on the server to assign the tiles into multicast and unicast groups, transmit all the multicast tiles on the multicast port, and unicast tiles separately to each user requiring these tiles.

The key is to address the assignment of the tiles into either multicast or unicast. The multicast transmission rate is typically lower than the unicast rate. As a result, a tile transmitted over multicast

*NOSSDAV'12,* June 7–8, 2012, Toronto, Ontario, Canada.

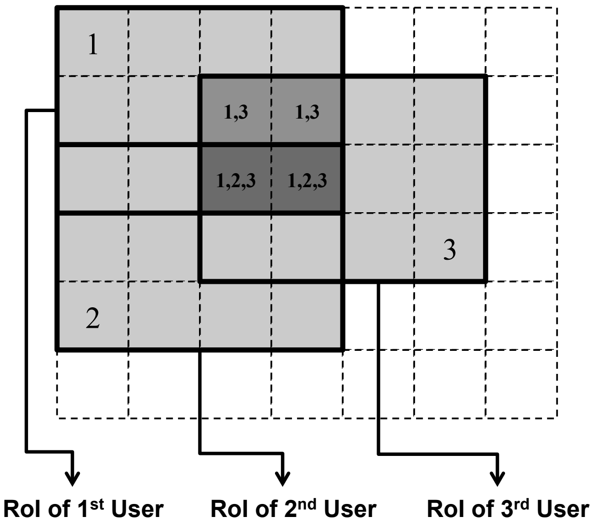


**Figure 1: A depiction of RoI overlap**

will arrive after a delay that is greater than when the tile is transmitted over unicast. Firstly, as a rule, a tile that is needed by all users in the system should be sent via multicast. If the tile is sent via unicast then the bandwidth utilized is as many times as the number of users in the system. Secondly, all the tiles in a frame should reach all the users before the tiles of the next frame can be transmitted. Hence a transmission deadline has to be met.

In this paper, we describe five methods to assign tiles into either the multicast and the unicast channel such that (i) the transmission deadline may be met, and (ii) wasteful reception of tiles is minimized.

The rest of the paper is organized as follows: in Section 2, a brief introduction to related work is mentioned. In Section 3, the problem of tile assignment is formulated. Section 4 describes five tile assignment methods. Section 5 is dedicated to an evaluation of the methods, followed by a conclusion in Section 6.

## 2. RELATED WORK

Partitioning data into multiple groups for transmission over different access channels is well studied as a channelization problem in a publish-subscribe paradigm. The solution invariably involves some form of resource allocation meeting a constraint. Adler et al. studied how to assign a published resource to different multicast groups with resource duplication [1]. The Hyper project uses a hybrid approach [14] where a multicast distribution tree may be bypassed in order to establish a unicast channel between the publisher and subscriber. Bickson et al. investigated the mapping of flows into multicast and unicast sets in a quantitative manner [2] with heuristic solutions such as, random assignment with flow merges, use of k-means clustering, and use of binary matrix decomposition. In all these approaches a generic data type is assumed. In this paper, we propose a solution for a specific video streaming case over a single hop wireless network. Although our framework fits the publish-subscribe paradigm, the need for the tiles constituting a region-of-interest reaching all the subscribers within a deadline is unique to the problem being solved. Further we explore a situation where the multicast grouping may change every second during streaming of a video. The goal that the video player should be simple in design restricts the number of multicast channels that it can subscribe to.

| | |
|---|---|
| $f$ | frame rate (frames per second) |
| $T$ | the set of tiles |
| $b$ | average tile size |
| $R_i$ | set of tiles selected by user $u_i$ |
| $T_m$ | set of tiles selected for multicast |
| $n$ | number of users |
| $n_t$ | number of users requiring a tile $t$ |
| $m_t$ | set of users requiring the tile $t$ |
| $B_m$ | multicast data rate |
| $B_u^i$ | unicast data rate for User $u_i$ |
| $p$ | probability that a multicast tile will reach all users |
| $\psi_u(t)$ | utility of a tile $t$ being unicast |
| $\psi_m(t)$ | utility of a tile $t$ being multicast |

**Table 1: Table of Notations**

The problem of assigning tiles into unicast and multicast groups can be formalized as an optimization problem involving a transmission deadline, the multicast and unicast rates, the number of users in the system, and their RoI access patterns. Hence, before we embark on strategies for tile assignment, the problem at hand is formalized.

## 3. PROBLEM DEFINITION

Let the video rate be $f$ frames per second. We assume that every second consists of one GoP of encoded video. Let each video frame be partitioned into a set $T$ of equal sized tiles, each of size $b$ bits (average tile size). Let $U = \{u_1, u_2, ..u_n\}$ be the set of users. User $u_i$ selects a RoI $R_i \subseteq T$ comprising of a subset of tiles to view. The RoIs for all the $n$ users, i.e., $\cup_{i=1}^{n} R_i$, must be delivered within $\frac{1}{f}$ seconds. Note that, as RoIs can overlap, a tile may be needed by multiple users.

In a setting described so far, it is prudent to assign tiles optimally to the multicast and unicast connections of the wireless network. As discussed in the previous section, we assume that all users subscribe to a single multicast channel (IP address and port number). The channel is known a priori, is subscribed to when the session starts, and does not change during the streaming session. Let $m_t$ be the set of users requiring tile $t$. If a tile $t$ is sent via multicast, then $t$ is also received by users who do not require the tile, as all users have subscribed to a single pre-determined multicast group. This unnecessary reception is termed as a wasteful reception. Receiving unwanted tiles would impact the client side performance in terms of CPU utilization and battery consumption as a result of processing these tiles. Hence the goal of the assignment should be to minimize wasteful reception while honoring the deadline of $\frac{1}{f}$ seconds.

To quantify the "wasteful" reception we define a utility function $\psi$. Consider a system where there are $n$ clients, and $n_t = |m_t|$ clients need a tile $t$. If $t$ is transmitted via unicast, then it has to be transmitted $n_t$ times, once to each user. Hence the utility is defined as

$$\psi_u(t) = \frac{n_t}{n}.$$

If $t$ is transmitted via multicast, then $n_t$ users that needs $t$ receive the tile, but the other $n - n_t$ users also receive the tile even though it is not required by them. The utility is now defined as

$$\psi_m(t) = \frac{n_t p - (np - n_t)}{n} = \frac{(1+p)n_t - np}{n},$$

where $p$ is the probability that a multicast packet reaches all the users. The term $\frac{np-n_t}{n}$ is a penalty term for the wasteful reception.

Further, define the supported multicast data rate as $B_m$ and the unicast data rate for user $u_i$ be $B_u^i$. Hence a tile $t$ when transmitted over the multicast channel would complete transmission in $\frac{b}{B_m}$ seconds, while it takes $\frac{b}{B_u^i}$ seconds on the unicast channel for user $u_i$.

Now, let $T_m$ be the set of tiles being multicast, and $R_i - T_m$ be the set of tiles being unicast to user $u_i$ in the system, The transmission deadline can be expressed as

$$\frac{|T_m|b}{B_m} + \sum_{i=1}^{n} \left( \frac{|R_i - T_m|b}{B_u^i} \right) \leq \frac{1}{f}. \tag{1}$$

A tile assignment is said to be *valid* if it satisfies the Inequality (1). Our goal is to find the valid tile assignment such that the total utility in the system (over all tiles) is maximized. Note that a valid tile assignment may not exist.

# 4. TILE ASSIGNMENT ALGORITHMS

We now describe several possible algorithms to assign the tiles, starting with two simple algorithms that serves as the baseline for comparisons.

## 4.1 Minimizing Transmission Delay

We begin with a naive tile assignment algorithm that compares the time to send a tile $t$ over the unicast channel and the multicast channel. If $t$ has a shorter transmission delay over the multicast channel, we transmit $t$ via multicast. Otherwise, we send $t$ via unicast. This algorithm leads to the minimum total delay for sending the tiles. We call this algorithm MIN-DELAY.

If the minimum delay produced by MIN-DELAY is larger than $1/f$, then there is no valid assignment. Note that MIN-DELAY does not consider utility when deciding the tile assignment. It, however, can support the most number of users while keeping within the deadline.

## 4.2 Maximizing Utility

The next baseline algorithm is one that finds the valid assignment that maximizes the utility. We can formulate the problem of finding maximum utility recursively as follows.

Let $T_k = \{t_1, t_2, ..., t_k\}$ be the subset of tiles (with $T_0 = \{\}$). Define $\Psi(T_k, D)$ as the maximum utility over all possible tile assignments for $T_k$ that satisfy the deadline $D$. We therefore are interested in finding $\Psi(T, 1/f)$.

We can define $\Psi(T_k, D)$ recursively by considering whether to multicast $t_k$ (for $k \geq 1$). If $t_k$ is sent via multicast, then deadline $D$ reduces by $b/B_m$. The total utility is

$$\Psi_k^m = \Psi\left(T_{k-1}, D - \frac{b}{B_m}\right) + \psi_m(t_k).$$

Otherwise, if $t_k$ is sent via unicast, $D$ reduces by $\sum_{i \in m_{t_k}} b/B_u^i$, and

$$\Psi_k^u = \Psi\left(T_{k-1}, D - \sum_{i \in m_{t_k}} \frac{b}{B_u^i}\right) + \psi_u(t_k).$$

The maximum utility is then:

$$\Psi(T_k, D) = \begin{cases} \max\{\Psi_k^u, \Psi_k^m\} & \text{if } k \geq 1 \text{ and } D \geq 0 \\ 0 & \text{if } k = 0 \text{ and } D \geq 0 \\ -\infty & \text{if } D < 0 \end{cases}$$

Implementing the above recursion leads to an exponential time search algorithm. To keep the running time tractable, we quantize the deadline $D$ and use dynamic programming, which caches the intermediate results. The resulting algorithm is pseudo polynomial. We denote this algorithm as MAX-UTIL. While MAX-UTIL has a tractable running time (in the order of minutes for our implementation), it is still too slow for our deployment where the running time is required to be within a few milliseconds. We therefore explore other methods below.

## 4.3 Threshold Algorithm

The first heuristic is called THRESHOLD, and decides the tile assignment based on utility alone. It works as follows: For each tile $t$, if $\psi_m(t) > 0$ then $t$ is multicast, otherwise $t$ is unicast. THRESHOLD simply ensures that the utility is always positive.

Since THRESHOLD does not consider delay when assigning tiles, the resulting assignment from THRESHOLD is not always valid. It is, however, a fast, linear time, algorithm.

## 4.4 Greedy Algorithm

We now discuss a greedy heuristic (GREEDY) to assign the tiles. The greedy algorithm begins with the solution from MIN-DELAY and repeatedly converts a multicast tile to unicast until the deadline is violated.

Consider what happens when we convert a multicast tile $t$ to unicast. Tile $t$ will be transmitted via unicast to every user in $m_t$. Let $T'_m = T_m - \{t\}$ be the new set of multicast tiles. Then Inequality (1) is changed to:

$$\begin{aligned} |T'_m|\frac{b}{B_m} + \sum_{i=1}^{n} |R_i - T'_m|\frac{b}{B_u^i} &\leq \frac{1}{f} \\ (|T_m| - 1)\frac{b}{B_m} + \sum_{i=1}^{n} |R_i - T_m|\frac{b}{B_u^i} + \sum_{l \in m_t} \frac{b}{B_u^l} &\leq \frac{1}{f}. \end{aligned} \tag{2}$$

The time to transmit all the tiles changes by

$$\Delta T = \left( \sum_{l \in m_t} \frac{b}{B_u^l} \right) - \frac{b}{B_m}. \tag{3}$$

If $\Delta T$ is negative (when $B_m \ll B_u^l$ and $n_t$ is small), then we have reduced the time to transmit the tiles. On the other hand, the total utility changes by

$$\begin{aligned} \Delta\psi &= \frac{n_t}{n} - \frac{n_t p - (np - n_t)}{n} \\ &= (n - n_t)\frac{p}{n}. \end{aligned} \tag{4}$$

Since $\Delta\psi$ is always non-negative, the total utility never decreases by reassigning a multicast tile $t$ to unicast.

Equations (3) and (4) lead to the following greedy algorithm: repeatedly find a tile $t \in T_m$ with smallest $n_t$ (to maximize the increase in utility) to unicast, until such operation would have violated the deadline constraint.

Note that if the unicast bandwidth $B_u^i$ is the same for all users, the selected tile will also give the smallest $\Delta T$. GREEDY is therefore optimal in this case.

## 4.5 Expectation

The four methods presented so far are run-time algorithms that rely on the current tile access pattern. We have also considered an offline algorithm that relies only on the probability of access to a tile in the frame. We call this offline algorithm EXPECTATION.

In many videos, the RoI has a locality of reference that may be predicted [11] and exploited [10, 12]. The probability that users would select a region is dependent on the content in the video and changes for every GoP. EXPECTATION uses the tile access probability to assign tiles into unicast and multicast groups, and does not depend on the actual access pattern. This algorithm can therefore be run offline and the tile assignment can simply be looked up during run time.

The tile assignment computed by EXPECTATION relies on the number of users $n$. The algorithm therefore would need to precompute the tile assignment for different values $n$.

Let $P(t)$ be the probability of accessing a tile $t$. The unicast utility can be written as $\psi_u(t) = P(t)$ and the multicast utility is $\psi_m(t) = (1+p)P(t) - p$ (where $pP(t)$ is the reward and $p - P(t)$ is the penalty).

The optimal tile assignment in this case can be computed recursively in a manner similar to MAX-UTIL. We replace $n_t$ in MAX-UTIL with $nP(t)$, and $\sum_{i \in m_t} \frac{b}{B_u^i}$ with $\frac{nb}{B_u}$, where $B_u$ is the average unicast bandwidth to a user, and obtain the following:

$$\Psi_k^m = \Psi\left(T_{k-1}, D - \frac{b}{B_m}\right) + (1+p)P(t_k) - p$$

$$\Psi_k^u = \Psi\left(T_{k-1}, D - \frac{nbP(t_k)}{B_u}\right) + P(t_k)$$

Since EXPECTATION is an offline algorithm, we can afford the extensive computation time needed to run the algorithm.

# 5. EVALUATION

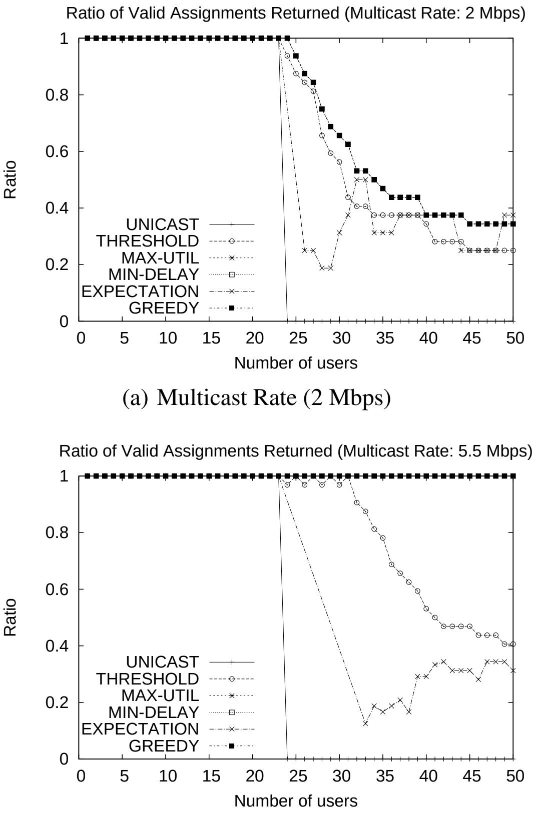
In the previous section, five methods to manage tile assignment over the multicast and unicast channels was detailed. In this section, we compare the methods in terms of their ability to meet the transmission deadline and the resulting utility of the system.

The experiment is simulation based. We set the multicast channel rate $B_m$ to 2 Mbps, and the unicast rate $B_u^i$ varies between 18 to 30 Mbps for the users. The video has a resolution of 1920×1088, with tile size of 32×32, at 25 frames per second ($f = 25$). The decoder plays the video by fetching all tiles of the RoI, before decoding for playback. The video rate is assumed to be 0.99 Mbps at 352×288 and the RoI dimension is 352×288. There are 99 tiles in each RoI. We assume that the RoI has an average data rate of 0.99 Mbps irrespective of which frame dimension it is cropped from. The bitrate of each tile $b$ is thus 0.01 Mbps.

To compute $\psi_m(t)$, we set the probability that a multicast transmission reaches all the users, $p$, to 1 and 0.6386, for multicast rate of 2 Mbps and 5.5 Mbps respectively. These valeus are based on the throughput rate as measured by Khayam et al. [9].

We use the RoI traces of 70 users who accessed a zoomable video system described in Carlier et al. [3], and use these traces to compute tile access probabilities for each of the four video resolutions. The zoomable video system relies on multiple resolution videos with compressed domain RoI cropping in order to create a virtual zoom and pan effect. Hence traces from such a system are ideal for our evaluation.

We use four different RoI patterns to evaluate the assignment methods and present the average results from these patterns below. In the case of EXPECTATION, we need a training set and a testing set. The training set is the four RoI patterns mentioned earlier, from which the probability $P(t)$ is obtained. Once the tile positions have been marked for unicast/multicast, we use the test set of RoI patterns to validate the usefulness of EXPECTATION. The test set represents the run-time pattern that needs to be handled by the server and is generated randomly following the same distribution as the training set.



(a) Multicast Rate (2 Mbps)

(b) Multicast Rate (5.5 Mbps)

**Figure 2: Ratio of Times a Given Algorithm Yields a Valid Assignment**

With the parameters and inputs as described above, we simulate the different methods for different number of users, $n$, ranging from 1 to 50. For each $n$, we simulate 32 times (8 times for each pattern) and average the results.

## 5.1 Results

**Number of Users Supported.** Figure 2 shows the percentage of instances where a given method yields a valid assignment that meets the deadline. The line labeled UNICAST is a naive method where every tile is sent via the unicast channel. This method cannot meet the target deadline when $n$ is larger than 23.

The figure shows that GREEDY, MIN-DELAY, and MAX-UTIL give a valid assignment in more instances than the other algorithms (the three lines coincide with each other). At multicast rate of 5.5 Mbps, these three methods give a valid assignment in all cases simulated. Recall that GREEDY reassigns tiles based on the assignment from MIN-DELAY, while keeping the assignment within the deadline. Therefore, GREEDY yields a valid assignment if and only if MIN-DELAY has an assignment. MAX-UTIL is also able to yield a valid assignment if one exists, since it exhaustively searches for all possible assignments.

Figure 2 also shows that EXPECTATION and THRESHOLD produce more assignments that are not valid. The result for THRESHOLD is expected, since it does not consider deadline in assigning the tiles. The result for EXPECTATION, however, is unexpectedly low. Recall that EXPECTATION maximizes for utility offline using access probability and average unicast bandwidth as input. As a result, it tends to yield an assignment whose *expected* total transmission time is close to the deadline. Variations in the actual access pattern and unicast bandwidth, however, would lead to violation of deadline, invalidating the chosen assignment. Due to the low success rate of EXPECTATION, we will not discuss it further in the following paragraphs.

**Transmission Time.** Figure 3 shows the average transmission time for the tile assignment computed by the different methods. Only valid assignments are included in the averages. The horizontal line at 0.04s indicates the deadline $1/f$.

The GREEDY and MAX-UTIL methods have the highest transmission time among the existing methods, while keeping the transmission time within the deadline. This result is expected due to the following reasons: GREEDY repeatedly reassigns multicast tiles to unicast until the deadline is violated, and therefore bumps the transmission time to as close to the deadline as possible; MAX-UTIL explores all possible solutions within the deadline, and naturally finds an assignment close to the deadline.

The figure also indicates that THRESHOLD has a lower transmission time than GREEDY and MAX-UTIL for larger $n$, but a higher transmission time for small $n$. The condition for multicasting a tile can be expressed as $n_t > np/(1+p)$. Thus, in the case of 2 Mbps multicast rate, only tiles that are needed by more than half of the users are multicast, resulting in potentially large number of unicast tiles, and therefore a larger transmission time than the other methods for small $n$. As $n$ increases, many of the inputs result in an invalid assignment when we use THRESHOLD to compute the assignment. The inputs that still yield a valid assignment are those where the RoIs are highly concentrated (high $n_t$) and benefit from multicast.

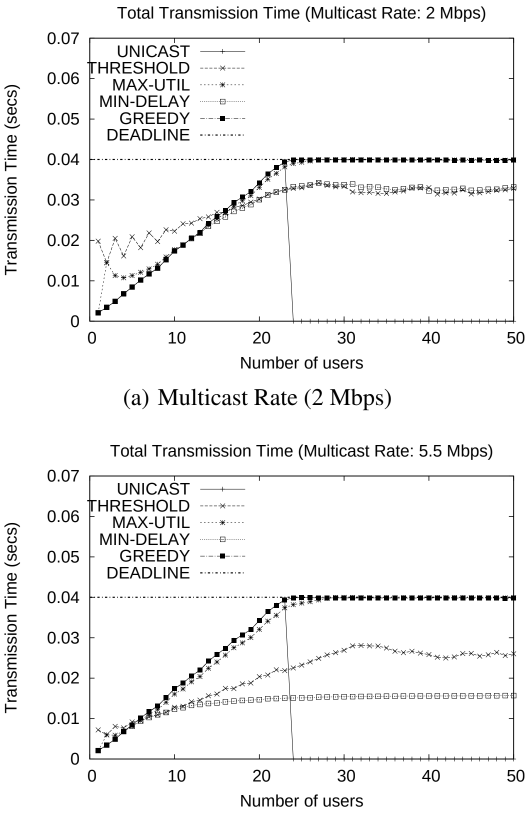
Figure 3(b) shows that the total transmission time of MIN-DELAY is significantly lower at 5.5 Mbps than all the other methods as it now has the opportunity to assign tiles to the multicast channel that has a lower transmission time (higher bandwidth). It also supports as many users as GREEDY or MAX-UTIL but at a lower transmission time (Figure 2). This result clearly indicates that many more users can be supported in the system (but at a higher channel error rate).

Note that in the case of a multicast rate of 2 Mbps, for some values of $n$, the average transmission time for THRESHOLD is even lower than MIN-DELAY. This lower value is due to some instances where THRESHOLD fails to give a valid assignment even when one exists. Since we only show the average results for valid assignments, the average value for THRESHOLD becomes lower than MIN-DELAY.

**Utility.** Figure 4 shows the total utility of all tiles (average over all instances that give a valid assignment) for different methods. For $n \leq 23$, there is a trivial solution that gives the highest utility (unicast every tile). The THRESHOLD method, however, chooses to multicast the popular tiles and pays the penalty for delivering unwanted tiles to some users, resulting in lower utility.

For larger $n$, THRESHOLD results in largest total utility (even larger than MAX-UTIL). This result is, again, due to failure of THRESHOLD to return a valid assignment. Since MAX-UTIL always finds a valid assignment as long as one exists, it sometimes returns an assignment that gives a low utility value, leading to lower average utility overall.

The figure also indicates that GREEDY is doing as good as MAX-UTIL in terms of utility in our experiments. GREEDY always converts a multicast tile to unicast in a way that maximizes utility (lowest $n_t$), and if this operation also always leads to smallest increase in transmission time, then GREEDY is also optimal. We found that this condition holds in our simulation.



(a) Multicast Rate (2 Mbps)

(b) Multicast Rate (5.5 Mbps)

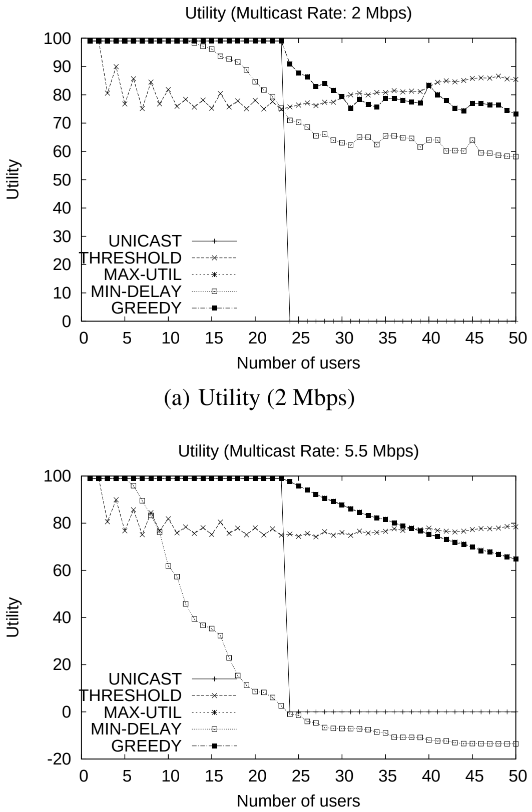
**Figure 3: Total Transmission Time**

Finally, we note that MIN-DELAY gives low utility than GREEDY and MAX-UTIL for most cases. This is expected, since it does not optimize for utility. The impact is more severe at multicast rate of 5.5 Mbps as MIN-DELAY assigns more tiles to the multicast channel in order to reduce the total transmission time, leading to very low utility. For large $n$, the utility becomes negative for MIN-DELAY, indicating that more useless tiles are received than useful tiles. Hence, even though MIN-DELAY can support more users than other methods we studied, it is not suitable in scenarios with many users.

**Discussion.** Our simulation results show that GREEDY is a reasonable method to use. It results in a valid assignment that meets the deadline as long as one exists, and is optimal (in terms of utility) under some conditions. It is fast: requiring an $O(|T|)$ step to find the minimum delay assignment, an $O(|T| \log |T|)$ step to sort the multicast tiles in increasing order of $n_t$, and finally another $O(|T|)$ step to convert the multicast tiles to unicast tiles.

## 6. CONCLUSION

We present five methods to efficiently transmit tiled regions of a video supporting region-of-interest (RoI) streaming in a wireless LAN. Each method decides to multicast or unicast a set of tiles based on how many users require these tiles and the supported data rates. A utility function that quantifies wasteful reception of



(a) Utility (2 Mbps)

(b) Utility (5.5 Mbps)

**Figure 4: Total Utility**

tiles as a result of multicasting was suggested. We showed that the GREEDY method always meets the transmission deadline (as long as a solution exists) and leads to larger utility than other methods.

In this paper, only a simple, abstract, model of the problem is considered. We plan to extend our model to include more realistic settings, such as varying *b* based on estimated encoding rate of each tile. We also plan to implement the assignment methods into our system and measure their performance in practice.

## Acknowledgement

This research is conducted under the NExT Search Center, supported by the Singapore National Research Foundation and the Interactive Digital Media R&D Program Office of Media Development Authority under research grant WBS:R-252-300-001-490.

## 7. REFERENCES

[1] M. Adler, Z. Ge, J. F. Kurose, D. Towsley, and S. Zabele. Channelization problem in large scale data dissemination. In *Proceedings of the Ninth International Conference on Network Protocols*, ICNP '01, pages 100–109, Washington, DC, USA, 2001.

[2] D. Bickson, E. N. Hoch, N. Naaman, and Y. Tock. A hybrid multicast-unicast infrastructure for efficient publish-subscribe in enterprise networks. In *Proceedings of the 3rd Annual Haifa Experimental Systems Conference*, SYSTOR '10, pages 1–7, Haifa, Israel, 2010.

[3] A. Carlier, R. Guntur, V. Charvillat, and W. T. Ooi. Combining content-based analysis and crowdsourcing to improve user interaction with zoomable video. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 43–52, Scottsdale, AZ, USA, 2011. ACM.

[4] S. Deshpande and W. Zeng. Scalable streaming of JPEG2000 images using hypertext transfer protocol. In *Proceedings of the 9th ACM International Conference on Multimedia*, MM '01, pages 372–381, Ottawa, Canada, 2001.

[5] W. Feng, T. Dang, J. Kassebaum, and T. Bauman. Supporting region-of-interest cropping through constrained compression. In *Proceedings of the 16th ACM International Conference on Multimedia*, MM '08, pages 745–748, Vancouver, Canada, 2008.

[6] S. Halawa, D. Pang, N.-M. Cheung, and B. Girod. ClassX: an open source interactive lecture streaming system. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 719–722, Scottsdale, AZ, USA, 2011.

[7] S. Heymann, A. Smolic, K. Mueller, Y. Guo, J. Rurainsky, P. Eisert, and T.Wiegand. Representation, coding and interactive rendering of high-resolution panoramic images and video using MPEG-4. In *Proceedings of the Panoramic Photogrammetry Workshop*, PPW '05, 2005.

[8] M. Inoue, H. Kimata, K. Fukazawa, and N. Matsuura. Interactive panoramic video streaming system over restricted bandwidth network. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1191–1194, Firenze, Italy, 2010.

[9] S. Khayam, S. Karande, H. Radha, and D. Loguinov. Performance analysis and modeling of errors and losses over 802.11b LANs for high-bit-rate real-time multimedia. *Signal Processing: Image Communication*, 18:575–595, 2003.

[10] A. Mavlankar and B. Girod. Pre-fetching based on video analysis for interactive region-of-interest streaming of soccer sequences. In *Proceedings of the 16th IEEE International Conference on Image Processing*, ICIP '09, pages 3061–3064, Cairo, Egypt, 2009.

[11] A. Mavlankar, D. Varodayan, and B. Girod. Region-of-Interest prediction for interactively streaming regions of high resolution video. In *Proceedings of International Packet Video Workshop*, PV '07, pages 68–77, Lausanne, Switzerland, 2007.

[12] K. Q. M. Ngo, R. Guntur, and W. T. Ooi. Adaptive encoding of zoomable video streams based on user access pattern. In *Proceedings of the 2nd ACM Multimedia Systems Conference*, MMSYS '11, pages 211–222, San Jose, CA, USA, 2011.

[13] K. Q. M. Ngo, R. Guntur, A. Carlier, and W. T. Ooi. Supporting zoomable video streams with dynamic region-of-interest cropping. In *Proceedings of the 1st ACM Multimedia Systems Conference*, MMSYS '10, pages 259–270, Phoenix, AZ, USA, 2010.

[14] R. Zhang and Y. C. Hu. Hyper: A hybrid approach to efficient content-based publish/subscribe. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, ICDCS '05, pages 427–436, Washington, DC, USA, 2005.