Journal of Mobile Multimedia, Vol. 1, No.3 (2005) 198-210

A NEW VOD TECHNIQUE TO SUPPORT CLIENT MOBILITY

KATSUHIKO SATO

Japan Radio Co., Ltd.
sato@lab.jrc.co.jp

MICHIAKI KATSUMOTO

National Institute of Information and Communications Technology
katumoto@nict.go.jp

TETSUYA MIKI

The University of Electro-Communications
miki@ice.uec.ac.jp

Received May 12, 2005
Revised Aug 13, 2005

This paper introduces fragmented patching, a new video on-demand technique that enables mobile clients to receive a video stream while moving freely. Patching techniques that significantly reduce the required network bandwidth through multicasting have shown potential for on-demand video distribution. However, patch-flow techniques based on unicast data are unsuitable for providing services to mobile clients because an intricate form of mobile routing is needed for each unicast flow to enable it to individually follow a moving client. Conversely, in fragmented patching, patch flows are sent via broadcasting. The patch flows are divided into segments to avoid increasing traffic due to broadcasting; each of the segments is aggregated to be shared with as many clients as possible. In addition, we have considered broadcasting shared flows also to eliminate any overhead arising from multicast tree construction. This paper analyzes the network bandwidth required for fragmented patching for two cases: when the patch flow is broadcast and the shared flow is multicast, and when both the patch and shared flows are broadcast. Numerical analysis based on the traffic intensity (Erlang) has revealed that the aggregation effect caused by segmenting patch flows counteracts the increase in traffic caused by broadcasting. It also showed that fragmented patching reduces the required bandwidth by a greater extent than other patching techniques even when both the patch and shared flows are broadcast.

Key words: VOD, mobility, patching, multicast, broadcast

1 Introduction

Broadband wireless access and mobile network technologies enable the distribution of rich content and the provision of a wide range of services to mobile users. This paper presents a new video-on-demand (VOD) technique, called fragmented patching, which supports client mobility, enabling clients to move around freely even while receiving a video stream. Current VOD services require that clients have a fixed connection to a network at least while receiving video. Patching techniques, developed to reduce the required network bandwidth through multicasting, can potentially be applied for on-demand video distribution. With these techniques, video content is sent by multicasting (i.e., through a shared flow) and shared by clients who submit requests at about the same time. The initial part of the

multicast content data, which is unavailable to clients who submit later requests, is individually delivered through unicasting (i.e., through a patch flow).

Individually unicasting patch flows to clients, though, is unsuitable for providing services to mobile clients because of the mobile routing overhead involved in individually routing each unicast flow to variously changing destinations. If the unicast routing is based on location-dependent addressing such as IP network, the overhead becomes particularly large. Techniques using a Care-of Address (CoA) (e.g., Mobile IP) must take route optimization into consideration to prevent unnecessary traffic. As the period of time involved in binding the CoA process and creating optimized route becomes longer, it becomes harder to guarantee glitch-free playback of streaming video since data received through a patch flow is immediately played back. This mobile routing overhead increases the burden on networks as both the request rate and the frequency of client movement increase (the latter basically depends on the projected cell size and each client's moving speed).

We first attempted to send patch flows via broadcasting, which enabled moving clients to immediately receive patch flows anywhere. Once the physical or logical distribution tree is constructed, broadcasting does not require any further routing procedures. This is considerably simpler than even multicasting where the distribution tree must be dynamically grafted and pruned. However, broadcasting leads to unnecessary traffic on links that have no clients on them and thus increases the required network bandwidth. We then proposed breaking down each patch flow into segments that could be aggregated so that as many clients as possible could share them. The length of a segment was set to the reciprocal of the arrival rate of request (i.e., the request interval) to provide the maximum aggregation effect.

Meanwhile, the shared flow, which is multicast to multiple clients, is relatively applicable to supporting client mobility since multicast routing has a routing mechanism based on location-independent addressing; i.e., it only branches the flow to the client's new destination. Furthermore, branching the flow is unnecessary if there are already other clients with which the moving client can share the flow at the new destination. However, there still remains the overhead of dynamically grafting and pruning branches to maintain an optimal multicast tree. The frequency of constructing multicast tree increases as the request rate rises, and increased client movement accelerates the frequency increase. Therefore, we next attempted to send the shared flows via broadcasting as was done with the patch flows.

This paper analyzes the network bandwidth required for fragmented patching for two cases: when the patch flow is broadcast and the shared flow is multicast, and when both the patch and shared flows are broadcast. The two cases are then compared with patching techniques through mathematical models that determine the average usage of the link bandwidth with respect to the traffic intensity (Erlang). Through this analysis, we show that fragmented patching lessens traffic, although it adds to traffic on branch links when the request rate is low. For example, the technique with patch flow broadcast and shared flow multicast reduces traffic intensity by 44% on the trunk link, but increases it by 11 and 68% on branch links (which branch to four and eight links, respectively) when the request rate is 20 and the content length is 2. However, it reduces traffic intensity by 67% on the trunk link and by 36 and 6% on the same branch links when the request rate is 100. The technique with both patch and shared flow broadcast reduces traffic intensity by 44% on the trunk link, but increases it by 13 and 76% on branch links (which branch to four and eight links, respectively) when the request rate is 20. However, it reduces traffic intensity by 67% on the trunk link and by 36 and 6% on the same branch links when the request rate is 100.

2 Patching Techniques

A number of effective VOD techniques using multicasting have been reported. Batching [1], piggybacking [2], and the block-transfer based techniques of Woo and Kim [3] and Kalva and Fuhrt [4] are early examples of techniques developed prior to patching. Carter and Long [7], Hua and Cai [8, 9], and Gao and Towsley [10] reported patching techniques based on streaming-transfer techniques. These techniques require less bandwidth availability at the client's network interface and fewer multicast groups, while also providing immediate delivery. Gao and Towsley used a mathematical approach to show the required server bandwidth and optimal generation rate of multicast flows using patching techniques. When a patching technique is used, media content sent via multicasting is called a shared flow as it is shared between clients who make requests at about the same time. The initial content data not available to clients that make later requests is individually delivered to these clients through unicasting – which is called a patch flow. As shown in Fig. 1, the first client (Req. 1) receives a shared flow only, and clients arriving later (Req. 2, 3, 4, and 5) receive both shared and patch flows. The patch flow for the Req. 2 client, for example, provides the initial portion of the data from the beginning of the content to the 0.5 content position, which corresponds to the arrival time of Req. 2. The shared data are not immediately played back, but are buffered until the patch flow data have been completely played back. To minimize network traffic, the technique dynamically determines the generation rate of shared flows according to the current request rate.

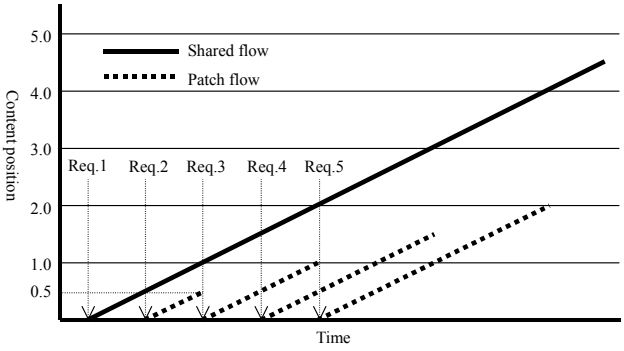

Figure 1 Patching technique.

3 Fragmented Patching to Support Mobility

To support client mobility, we have refined the patching technique. As mentioned, the patch flow is divided into segments that are broadcast and shared with as many clients as possible.

Figure 2 shows how the patch flows are shared. Each segment is expressed as $s1$, $s2$, … in order from the beginning of the content. Assuming that the request rate is now λ, every segment has the same length: $1/\lambda$. Consider a client corresponding to request 6, for example. This request arrives $5/\lambda$ after request 1, and the client then needs a patch flow that includes five segments: $s1$, $s2$, $s3$, $s4$, and $s5$. In this case, $s1$ and $s5$ must be newly sent, but $s2$, $s3$, and $s4$ can be shared with the previous clients corresponding to requests 5, 4, and 5, respectively. (In the figure, the segments that can be shared by a previous client are shown in gray.) Figure 2 shows one important rule: $s1$ is newly sent for every request, $s2$ is newly sent for every second request and $s3$ is newly sent for every third request. That is, sn is newly sent for every n_{th} request.

To help clarify the basic concept of fragmented patching, the above explanation is premised on every request interval being the same. In reality, though, intervals are always unsettled. Figure 3 shows an example of random request arrivals. Requests 4 and 5 arrive, respectively, $0.5/\lambda$ and $1/\lambda$ earlier than the normal time. Requests 6 and 8 arrive, respectively, $1/\lambda$ and $0.5/\lambda$ later than the normal time. In this situation, patch flows segments are shared as follows.

The premature request 5 allows the client to receive *s*2, which is shared with the previous client of request 3. The delayed request 6 means that for the client to receive *s*2, *s*3, and *s*6, the server must newly send these segments. However, this allows the next client to receive *s*1, *s*2, *s*3, and *s*6 as a flow shared with the client of request 6. The delayed request 8 means that for the client to receive *s*2 and *s*4, the server must newly send these segments.

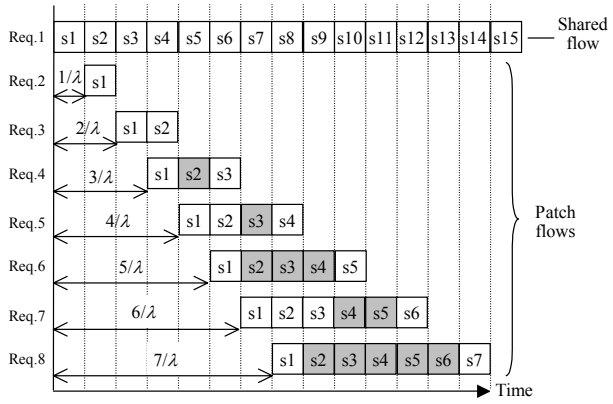

Figure 2 Fragmented patching technique: segments shown in gray are shared with those needed for a previous request; i.e., they are not actually sent.

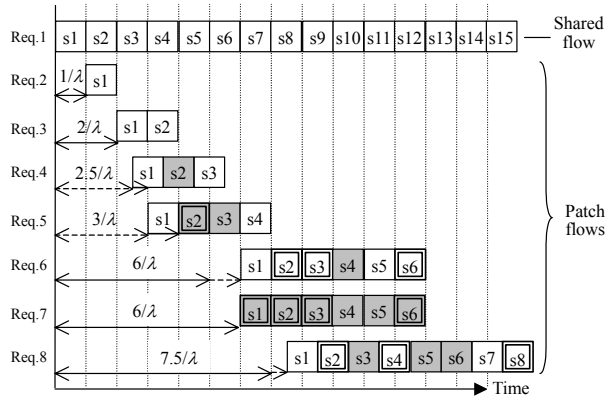

Figure 3 Fragmented patching technique: requests arrive at random. The segments with a double box are those that differ from the segments in Fig. 2.

As can be seen from Figs. 2 and 3, the number of segments that the server actually sends to clients remains approximately the same whether the request arrival is random or uniform. A request prior to the normal time allows the client to receive more segments that are to be dispatched to the previous clients. Meanwhile, the delayed request to the normal time allows more subsequent clients to share the segments to be dispatched to the client. That is, the required bandwidth should not be affected by requests arriving at random.

To implement fragmented patching, the server performs the following simple process.

Upon receiving a request, the server
- *checks the lapsed time from the start time of the last shared flow*
- *determines which segments the new client needs to receive*
- *sees if any of those segments have already been scheduled to be sent*
- *sees if any of the scheduled segments can be received by the new client*
- *determines which segments must be newly sent*
- *records the segment numbers and time to be sent in the schedule table*
- *informs the client of the schedule*
- *sends the segments according to the schedule table*

In addition, the server calculates the average request rate every time it receives a request. The server changes the segment length if it detects a significant change in the request rate. Meanwhile, clients select the segments they want to receive according to the received schedule table.

4 Mathematical Models

This section analyzes the required network bandwidth of both trunk and branch links for both patching and fragmented patching. For the latter, we consider two cases: when the patch flow is broadcast and the shared flow is multicast, and when both the patch and shared flows are broadcast. The required network bandwidth can be expressed as the traffic intensity (Erlang), which is the product of the average request rate, the average flow length, and the average flow bandwidth.

This paper lets h, λ, and τ denote, respectively, the content length, the request rate, and the generation rate of shared flows. We assume that the content is transmitted at a constant bit rate with bandwidth 1, except for the merge flow in hierarchical merging. Figure 4 shows the network model and the link definitions. The network has a balanced tree topology. The trunk link is a single link that directly connects to the server. The branch links are those that are branched by nodes capable of multicasting. The number of branch links is expressed as m. The reason for using a symmetric tree is that we can assume the request rate on m branched links is equal to the request rate on the trunk link divided by m, if we also assume that the same number of clients connect to each of the links and requests occur evenly; i.e., there is no deviation in the request rate among links. (Note that a real network does not always form a symmetric tree. The network model can be transformed into an asymmetric tree. In such a case, m is determined as the ratio of the downstream request rate to the total request rate.) Each flow is actually delivered at varied intervals, although equal intervals are shown in Figs. 5 and 6. We assume that requests arrive randomly within a short time span; i.e., each request occurs independently without any correlation with other requests.

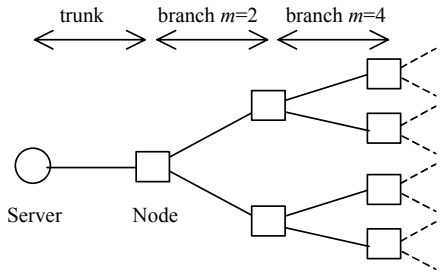

Figure 4 Network Model.

4.1 Patching Technique

4.1.1 Traffic Intensity on the Trunk Link

We first consider the traffic intensity of shared flows. The bandwidth $b_{\text{p\_shared\_t}}$, the rate $r_{\text{p\_shared\_t}}$, and the length $l_{\text{p\_shared\_t}}$ of a shared flow are, respectively,

$$b_{\text{p\_shared\_t}} = 1,\ r_{\text{p\_shared\_t}} = \tau,\ l_{\text{p\_shared\_t}} = h.$$

The traffic intensity of shared flows $\rho_{\text{p\_shared\_t}}$ is then

$$\rho_{\text{p\_shared\_t}} = b_{\text{p\_shared\_t}} \times r_{\text{p\_shared\_t}} \times l_{\text{p\_shared\_t}} = \tau h .$$

Next, we consider the traffic intensity of patch flows. The bandwidth $b_{\text{p\_patch\_t}}$ and the rate $r_{\text{p\_patch\_t}}$ of a patch flow are, respectively,

$$b_{\text{p\_patch\_t}} = 1,\ r_{\text{p\_patch\_t}} = \lambda - \tau .$$

The number of patch flows between two shared flows is $\lambda/\tau - 1$. As Fig. 5 shows, the patch flow lengths are $1/\lambda$, $2/\lambda$, $3/\lambda$, …, so the average length is

$$l_{\text{p\_patch\_t}} = \frac{1}{\lambda/\tau - 1} \sum_{k=1}^{\lambda/\tau - 1} \frac{k}{\lambda} = \frac{1}{2\tau} .$$

The traffic intensity of patch flows $\rho_{\text{p\_patch\_t}}$ is then

$$\rho_{\text{p\_patch\_t}} = b_{\text{p\_patch\_t}} \times r_{\text{p\_patch\_t}} \times l_{\text{p\_patch\_t}} = 1 \times (\lambda - \tau) \times \frac{1}{2\tau} = \frac{\lambda - \tau}{2\tau} .$$

Therefore, with the patching technique the total traffic intensity on the trunk link $\rho_{\text{p\_t}}$ is

$$\rho_{\text{p\_t}} = \rho_{\text{p\_shared\_t}} + \rho_{\text{p\_patch\_t}} = \tau h + \frac{\lambda - \tau}{2\tau} \quad (1)$$

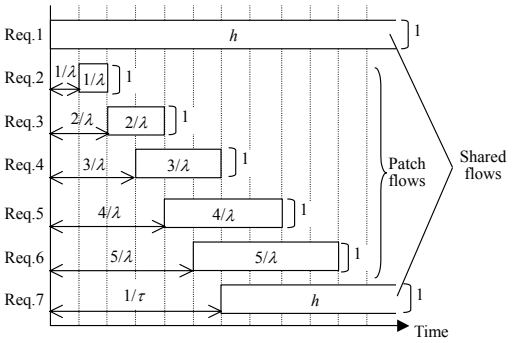

Figure 5 Trunk link flows with the patching technique.

4.1.2 Traffic Intensity on the Branch Links

Starting with the traffic intensity of shared flows, the bandwidth $b_{\text{p\_shared\_b}}$ of a shared flow is

$$b_{\text{p\_shared\_b}} = 1.$$

The number of requests between two shared flows is expressed as N ($N = \lambda/\tau$). The possibility that a shared flow will not occur on a branch link is $((m-1)/m)^N$. The expected rate of shared flow occurrence on the branch link, $r_{\text{p\_shared\_b}}$, is

$$r_{\text{p\_shared\_b}} = \left(\frac{m-1}{m}\right)^N \times 0 + \left(1 - \left(\frac{m-1}{m}\right)^N\right) \times \tau = \left(1 - \left(\frac{m-1}{m}\right)^{\frac{\lambda}{\tau}}\right) \times \tau .$$

The shared flow on a branch link will have one of two lengths. When a request triggering the dispatch of a shared flow occurs on the branch link, the length of the shared flow is h (case 1 in Fig. 6). Otherwise, when a request that does not trigger the dispatch of a shared flow occurs on the branch link, the length of the shared flow is $h - 1/(\lambda/m)$ (case 2 in Fig. 6). Here, $1/(\lambda/m)$ is the time until the first patch flow occurs on the branch link. The possibility of case 1 is $1/N$ (i.e., as τ approaches λ – as the proportion of the shared flow becomes large – the frequency of case 1 increases). Therefore, the expected value for the length of the shared flow, $l_{\text{p\_shared\_b}}$, is

$$l_{\text{p\_shared\_b}} = h \times \frac{1}{N} + \left(h - \frac{1}{\lambda/m}\right)\left(1 - \frac{1}{N}\right) = h + \frac{m\tau}{\lambda^2} - \frac{m}{\lambda} .$$

The traffic intensity of shared flows $\rho_{\text{p\_shared\_b}}$ is then

$$\rho_{\text{p\_shared\_b}} = b_{\text{p\_shared\_b}} \times r_{\text{p\_shared\_b}} \times l_{\text{p\_shared\_b}} = 1 \times \left(1 - \left(\frac{m-1}{m}\right)^{\frac{\lambda}{\tau}}\right) \times \tau \times \left(h + \frac{m\tau}{\lambda^2} - \frac{m}{\lambda}\right) .$$

Regarding the traffic intensity for patch flows, the bandwidth $b_{\text{p\_patch\_b}}$ is

$$b_{\text{p\_patch\_b}} = 1.$$

When the number of branch links is m, the rate $r_{\text{p\_patch\_b}}$ is

$$r_{\text{p\_patch\_b}} = \frac{\lambda - \tau}{m} .$$

The average length of patch flows on a branch link is the same as on a trunk link:

$$l_{\text{p\_patch\_b}} = \frac{1}{2\tau} .$$

The traffic intensity of patch flows $\rho_{\text{p\_patch\_b}}$ is then

$$\rho_{\text{p\_patch\_b}} = b_{\text{p\_patch\_b}} \times r_{\text{p\_patch\_b}} \times l_{\text{p\_patch\_b}} = 1 \times \frac{\lambda - \tau}{m} \times \frac{1}{2\tau} = \frac{\lambda - \tau}{2m\tau} .$$

Therefore, with the patching technique the total traffic intensity on a branch link, $\rho_{\text{p\_b}}$, is

$$\rho_{\text{p\_b}} = \rho_{\text{p\_shared\_b}} + \rho_{\text{p\_patch\_b}} = \left(1 - \left(\frac{m-1}{m}\right)^{\frac{\lambda}{\tau}}\right) \times \tau \times \left(h + \frac{m\tau}{\lambda^2} - \frac{m}{\lambda}\right) + \frac{\lambda - \tau}{2m\tau} \quad (2)$$

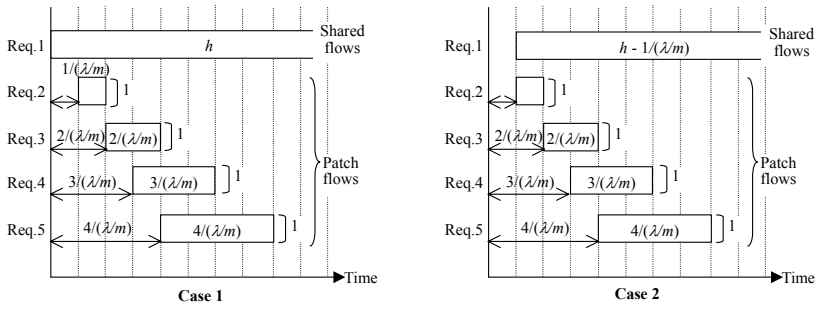

Figure 6 Branch link flows with the patching technique.

4.2 Fragmented Patching Technique

We now consider the two cases: when the patch flow is broadcast and the shared flow is multicast, and when both the patch and shared flows are broadcast. In the former case, the traffic on the branch links is lighter than that on the trunk link because of the multicast effect. In the latter case, the traffic is the same on both the trunk and branch links.

4.2.1 Traffic Intensity on the Trunk Link

The traffic intensity on the trunk is the same for both cases. It does not matter whether the shared flow is multicast or broadcast on the trunk link. We consider the traffic intensity of shared flows first. The bandwidth $b_{\text{f\_shared\_t}}$, the rate $r_{\text{f\_shared\_t}}$, and the length $l_{\text{f\_shared\_t}}$ of a shared flow are, respectively,

$$b_{\text{f\_shared\_t}} = 1,\ r_{\text{f\_shared\_t}} = \tau,\ l_{\text{f\_shared\_t}} = h.$$

The traffic intensity of shared flows $\rho_{\text{f\_shared\_t}}$ is then

$$\rho_{\text{f\_shared\_t}} = b_{\text{f\_shared\_t}} \times r_{\text{f\_shared\_t}} \times l_{\text{f\_shared\_t}} = \tau h\ .$$

This is the same as with the patching techniques.

Now we consider the traffic intensity of the fragmented patch flow. The bandwidth $b_{\text{f\_fragment\_t}}$ and the rate $r_{\text{f\_fragment\_t}}$ are

$$b_{\text{f\_fragment\_t}} = 1,\ r_{\text{f\_fragment\_t}} = \lambda - \tau.$$

As shown in Fig. 2, the total length of fragmented patch flows can be expressed as

$$n \times \frac{1}{\lambda} + \left\lceil \frac{n}{2} \right\rceil \times \frac{1}{\lambda} + ,\ldots, + \left\lceil \frac{n}{n-1} \right\rceil \times \frac{1}{\lambda} = \frac{1}{\lambda} \sum_{k=1}^{n-1} \left\lceil \frac{n}{k} \right\rceil \quad n = \lambda / \tau - 1\,,$$

where n is the number of requests between two shared flows. The average $l_{\text{f\_fragment\_t}}$ is

$$l_{\text{f\_fragment\_t}} = \frac{1}{n} \times \frac{1}{\lambda} \sum_{k=1}^{n-1} \left\lceil \frac{n}{k} \right\rceil = \frac{1}{\lambda(\lambda/\tau - 1)} \sum_{k=1}^{\lambda/\tau - 2} \left\lceil \frac{\lambda/\tau - 1}{k} \right\rceil.$$

The traffic intensity of a fragmented patch flow $\rho_{\text{f\_fragment\_t}}$ is then

$$\rho_{\text{f\_fragment\_t}} = b_{\text{f\_fragment\_t}} \times r_{\text{f\_fragment\_t}} \times l_{\text{f\_fragment\_t}} = \frac{\lambda - \tau}{\lambda(\lambda/\tau - 1)} \sum_{k=1}^{\lambda/\tau - 2} \left\lceil \frac{\lambda/\tau - 1}{k} \right\rceil.$$

Therefore, the total traffic intensity of the fragmented patching technique on the trunk link, $\rho_{\text{f\_t}}$, is

$$\rho_{\text{f\_t}} = \rho_{\text{f\_shared\_t}} + \rho_{\text{f\_fragment\_t}} = \tau h + \frac{\lambda - \tau}{\lambda(\lambda/\tau - 1)} \sum_{k=1}^{\lambda/\tau - 2} \left\lceil \frac{\lambda/\tau - 1}{k} \right\rceil \quad (3)$$

4.2.2 Traffic Intensity on the Branch Links

The traffic intensity on the branch links decreases as the number of branches m increases when the shared flow is multicast. We consider the traffic intensity of shared flows first. The bandwidth $b_{\text{f\_shared\_b}}$ of a shared flow is

$$b_{\text{f\_shared\_b}} = 1.$$

As with the patching technique, the possibility that a shared flow will not occur on a branch link is $((m-1)/m)^N (N = \lambda/\tau)$. The expected rate of shared flow occurrence on the branch link, $r_{\text{f\_shared\_b}}$, is

$$r_{\text{f\_shared\_b}} = \left(1 - \left(\frac{m-1}{m}\right)^{\frac{\lambda}{\tau}}\right) \times \tau .$$

Also, as in the patching technique, the shared flow has one of two lengths: h or $h - 1/(\lambda/m)$. The expected value for the length of the shared flow, $l_{\text{f\_shared\_b}}$, is

$$l_{\text{f\_shared\_b}} = h + \frac{m\tau}{\lambda^2} - \frac{m}{\lambda}.$$

The traffic intensity of shared flows $\rho_{\text{f\_shared\_b}}$ is then

$$\rho_{\text{f\_shared\_b}} = b_{\text{f\_shared\_b}} \times r_{\text{f\_shared\_b}} \times l_{\text{f\_shared\_b}} = 1 \times \left(1 - \left(\frac{m-1}{m}\right)^{\frac{\lambda}{\tau}}\right) \times \tau \times \left(h + \frac{m\tau}{\lambda^2} - \frac{m}{\lambda}\right).$$

Next, we consider the traffic intensity of fragmented patch flows. The bandwidth $b_{\text{f\_fragment\_b}}$, the rate $r_{\text{f\_fragment\_b}}$, and the length $l_{\text{f\_fragment\_b}}$ are the same as on the trunk link, so

$$b_{\text{f\_fragment\_b}} = 1,\ r_{\text{f\_fragment\_b}} = \lambda - \tau,\quad l_{\text{f\_fragment\_b}} = \frac{1}{\lambda(\lambda/\tau - 1)} \sum_{k=1}^{\lambda/\tau - 2} \left\lceil \frac{\lambda/\tau - 1}{k} \right\rceil.$$

The traffic intensity of a fragmented patch flow $\rho_{\text{f\_fragment\_b}}$ is then

$$\rho_{\text{f\_fragment\_b}} = b_{\text{f\_fragment\_b}} \times r_{\text{f\_fragment\_b}} \times l_{\text{f\_fragment\_b}} = \frac{\lambda - \tau}{\lambda(\lambda/\tau - 1)} \sum_{k=1}^{\lambda/\tau - 2} \left\lceil \frac{\lambda/\tau - 1}{k} \right\rceil.$$

Therefore, with the fragmented patching technique (when the shared flow is multicast) the total traffic intensity on the branch link, $\rho_{\text{f\_b}}$, is

$$\rho_{\text{f\_b}} = \rho_{\text{f\_shared\_b}} + \rho_{\text{f\_fragment\_b}} = \left(1 - \left(\frac{m-1}{m}\right)^{\frac{\lambda}{\tau}}\right) \times \tau \times \left(h + \frac{m\tau}{\lambda^2} - \frac{m}{\lambda}\right) + \frac{\lambda - \tau}{\lambda(\lambda/\tau - 1)} \sum_{k=1}^{\lambda/\tau - 2} \left\lceil \frac{\lambda/\tau - 1}{k} \right\rceil \quad (4)$$

5 Analysis and Considerations

This section compares the two techniques using the mathematical models.

Figure 7 depicts the traffic intensity on both the trunk and branch links with the patching technique. The curves are functions of τ, $\rho_{p\_t} = f(\tau)$, and $\rho_{p\_b} = f(\tau)$ from Eqs. 1 and 2, where the request rate λ and the content length h are set to 100 and 2, respectively. Figure 7 shows that the functions are downward convex curves, which consist of a linear function for shared flow traffic and a fractional function for patch flow traffic, and each ρ takes its minimum value when τ is a particular value, τ_{min}. The traffic intensity at any τ decreases as the number of branches m increases. This is because the request rate on the link simply falls as m rises. As τ approaches λ (as a greater proportion of the flows are transmitted as shared flows), ρ approaches the same traffic intensity as with simple unicast distribution; i.e., λh on the trunk and $\lambda h/m$ on the branches. The traffic intensity can be constantly maintained at a minimum by dynamically updating the generation rate of shared flows, τ_{min}, from the observed request rate.

Figure 8 depicts the traffic intensity with the fragmented patching technique when the shared flows are multicast and the patch flows are broadcast. The curves are the functions $\rho_{f\_t} = f(\tau)$ and $\rho_{f\_b} = f(\tau)$ from Eqs. 3 and 4, where λ and h are set to 100 and 2, respectively. Since the patch flows are broadcast, the patch flow traffic is the same among the trunk and branch links. The traffic for patch flows is relatively small, so the traffic for shared flows dominates. The curves therefore increase monotonically. We can thus assume that each ρ takes its minimum value at the smallest τ, namely, $\tau_{min} = 1/h$. Substituting τ_{min} into Eqs. 3 and 4,

$$\rho_{f\_t} = 1 + \frac{\lambda - 1/h}{\lambda(\lambda h - 1)} \sum_{k=1}^{\lambda h - 2} \left\lceil \frac{\lambda h - 1}{k} \right\rceil \quad (5)$$

$$\rho_{f\_b} = \left(1 - \left(\frac{m-1}{m}\right)^{\lambda h}\right) \times \left(1 + \frac{m}{\lambda^2 h^2} - \frac{m}{\lambda h}\right) + \frac{\lambda - 1/h}{\lambda(\lambda h - 1)} \sum_{k=1}^{\lambda h - 2} \left\lceil \frac{\lambda h - 1}{k} \right\rceil \quad (6)$$

The traffic intensity decreases as the number of branches m increases while τ is not small (i.e., the proportion of patch flow traffic is low). As τ approaches λ, ρ approaches the same traffic intensity as with simple unicast distribution. Note that the traffic intensity with fragmented patching with both the shared and patch flow broadcast is the same on both the trunk and branch links.

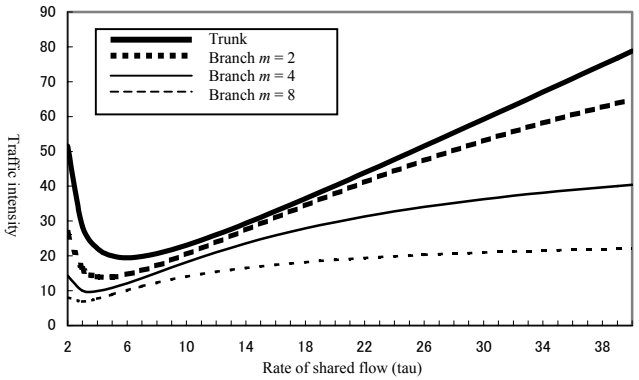

Figure 7 Traffic intensity on each link with patching.

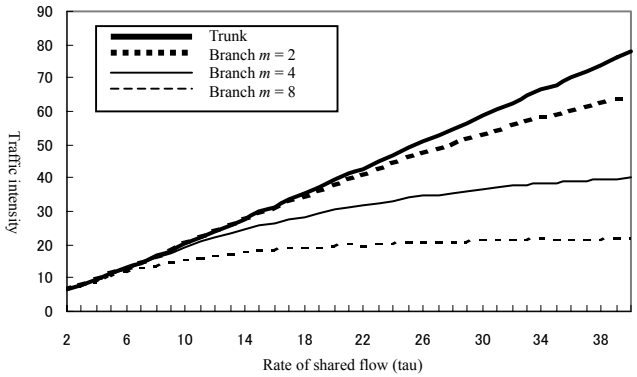

Figure 8 Traffic intensity on each link with fragmented patching.

Figures 9 and 10 compare the minimized traffic with the two techniques on the trunk and branch links (m = 4 and 8), respectively. The curves in Fig. 9 are the functions $\rho_{p\_t} = f(\lambda)$ from Eq. 1 and $\rho_{f\_t} = f(\lambda)$ from Eq. 5. The curves in Fig. 10 are the functions $\rho_{p\_b} = f(\lambda)$ from Eq. 2 and $\rho_{f\_b} = f(\lambda)$ from Eq. 6. The content length h is 2. For Eqs. 1 and 2, τ_{min} is computed for each λ. For Eqs. 5 and 6, τ_{min} is constant at $1/h$.

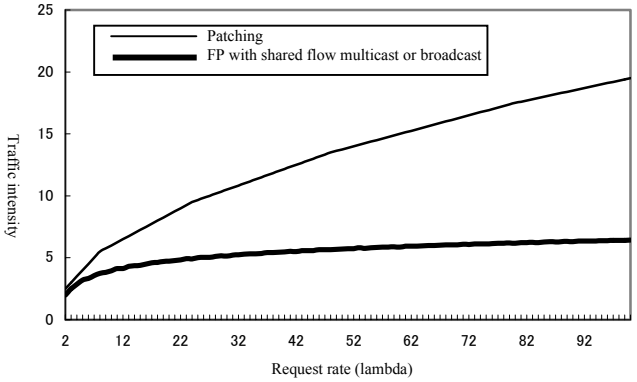

Figure 9 Comparison of minimized traffic intensity on the trunk link.

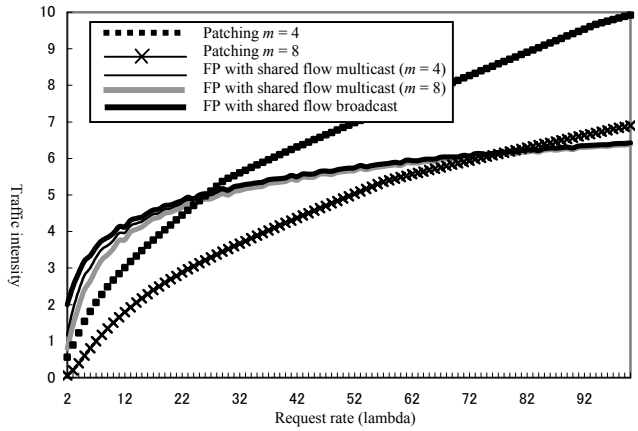

Figure 10 Comparison of minimized traffic intensity on the branch links (m = 4 and 8).

Both fragmented patching techniques (with the shared flow multicast or broadcast) greatly reduce traffic on the trunk link compared to the traffic with patching technique, but they increase traffic on the branch links because they use broadcasting for the patch flow or both the patch and shared flows (which delivers data to all links including those with no clients wanting to receive them). The increase in the rate of traffic intensity (the gradients of the curves) is much less with fragmented patching, though, than with patching. This is because the traffic for shared flows with fragment patching is constant for any request rate (i.e., the rate of shared flows is $1/h$ at any time and the traffic intensity is then always 1). In other words, with fragmented patching, the traffic increase is due to only to a rise in patch-flow traffic, as opposed to a rise in both shared and patch flow traffic with patching techniques. Therefore, at a higher request rate, the traffic intensity with fragmented patching is less than that with patching.

The difference in traffic between broadcasting and multicasting the shared flow in fragmented patching decreases as the request rate increases. This is because the shared flow traffic, whose transmission rate τ is very small (almost $1/h$), is shared with many clients and there is traffic on almost all the links at high request rate. The advantage of multicasting diminishes.

As a result, compared with patching techniques, fragmented patching with patch flow broadcast and shared flow multicast reduces traffic by 44% on the trunk link but increases it by 11 and 68% on branch links (m = 4 and 8, respectively) when the request rate λ = 20, and reduces traffic by 67% on the trunk and 36 and 6% on the branches (m = 4 and 8, respectively) when λ = 100. Meanwhile, fragmented patching with both patch and shared flow broadcast reduces traffic by 44% on the trunk link but increases it by 13 and 76% on branch links (m = 4 and 8, respectively) when the request rate λ = 20, and reduces traffic by 67% on the trunk and 36 and 6% on the branches (m = 4 and 8, respectively) when λ = 100.

Note that the broadcast of shared flow and/or segments allows the effectiveness of reducing traffic to decline as the number of branch links increases. This implies the proposed technique is suitable for small-scale networks where the request rate is relatively high (i.e., clients are densely distributed), but unsuitable for large-scale networks where the request rate is relatively low (i.e., clients are sparsely distributed).

6 Conclusion

In this paper, we have proposed a mobile VOD technique called fragmented patching, which is based on existing patching techniques. This technique breaks down patch flows into segments which are sent via broadcasting, enabling mobile clients to receive patch flows anywhere. As a result of aggregating segments to be shared among multiple clients, fragmented patching counteracts the increase in traffic caused by broadcasting.

Through numerical analysis based on the traffic intensity, we have shown that fragmented patching reduces traffic overall, although it adds to traffic on branch links at a low request rate. In addition, we showed that fragmented patching was almost equally effective as patching at reducing the traffic even though the shared flow is also broadcast. For example, fragmented patching with patch flow broadcast and shared flow multicast reduced traffic intensity by 44% on the trunk link, but increased traffic by 11 and 68% on branch links (which branched to four and eight links, respectively) when the request rate was 20 and the content length was 2. In contrast, it reduced the traffic intensity by 67% on the trunk link and by 36 and 6% on the same branch links when the request rate was 100. Fragmented patching with both shared and patch flow broadcast reduced traffic intensity by 44% on

the trunk link, but increased traffic by 13 and 76% on branch links (which branched to four and eight links, respectively) when the request rate was 20. In contrast, it reduced the traffic intensity by 67% on the trunk link and by 36 and 6% on the same branch links when the request rate was 100.

Fragmented patching is promising for high-demand video distribution within a particular area. We plan to explore the possibility of its application to the distribution of popular TV programs or local news programs over public wireless access services, or to multimedia learning systems (e.g., lectures in universities and training in companies) over private wireless access networks.

References

1. Dan, A., Sitaram, D., and Shahabuddin, P.: Scheduling Policies for an On-Demand Video Server with Batching, Proc. 2nd ACM Int'l. Multimedia Conference, (San Francisco, CA, Oct. 1994).
2. Golubchik, L., Lui, J., and Muntz, R.: Adaptive Piggy-back: A Novel Technique for Data Sharing in Video-On-Demand Storage Servers, ACM Multimedia System Journal, Vol.4, No.3, 1996.
3. Woo, H. and Kim, C.K.: Multicast scheduling for VOD services, Multimedia Tools and Applications 2(2), Mar.1996.
4. Kalva, H. and Fuhrt, B.: Techniques for improving the capacity of video-on-demand systems, Proc. 29th Annual Hawaii Int'l. Conf. on System Sciences, Wailea, HI, USA, IEEE Computer Society Press, Jan. 1996.
5. Uno, S., Tode, H., and Murakami, K.: Simple and Efficient Video-on-Demand Scheme with Segment Transmission over High Speed Network, IEICE Trans. Communications, Vol. E84-B, No.1, Jan. 2001.
6. Xie, Z., Uno, S., Tode, H., and Murakami, K.: The Scheduling of Contents Delivery with Multicast and Burst Transfer, IEICE Technical Report, NS2002-58, Jun. 2002.
7. Carter, S.W. and Long, D.E.: Improving Video-on-demand Server Efficiency Through Streaming Tapping, Proc. Int'l. Conf. on Computer Communication and Networks, (Las Vegas, Sep.1997).
8. Hua, K.A., Cai, Y., and Sheu, S.: Patching: A Multicast Technique for True Video-On-Demand Services, Proc. ACM Multimedia '98, (Bristol, U.K., Sept. 1998).
9. Cai, Y., Hua, K.A., and Vu, K.: Optimizing Patching Performance, Proc. Multimedia Computing and Networking '99, (San Jose, CA, Jan. 1999).
10. Gao, L. and Towsley, D.: Supplying Instantaneous Video-on-Demand Services Using Controlled Multicast, Proc. IEEE Int'l. Conf. on Multimedia Computing and Systems '99, (Florence, Italy, Jun. 1999).
11. Eager, D.L., Vernon, M.K., and Zahorjan, J.: Minimizing Bandwidth Requirements for On-Demand Data Delivery, Proc. 5th Int'l. Workshop on Multimedia Information System, (Indian Wells, CA, Jan. 1999).
12. Eager, D.L., Vernon, M.K., and Zahorjan, J.: Optimal and Efficient Merging Schedule for Video-on-Demand Servers, Proc. 7th ACM Int'l. Multimedia Conference, (Orlando, FL, Nov. 1999).
13. Eager, D.L., Vernon, M.K., and Zahorjan, J.: Bandwidth Skimming: A Technique for Cost-Effective Video-on-Demand, Proc. Multimedia Computing and Networking '00, (San Jose, CA, Jan. 2000).
14. Zhao, Y., Eager, D.L., and Vernon, M.K.: Network Bandwidth Requirement for Scalable On-Demand Streaming, Proc. 21st Annual Joint Conf. IEEE INFOCOM, (New York, NY, Jun. 2002).
15. Sato, K. and Katsumoto, M.: A Proposal of Multicast for Personalized Media Stream Delivery, Proc.16th Int'l. Conf. on Information Networking Vol. 2 4D-4, (Cheju Island, Korea, Jan. 2002).
16. Sato, K., Katsumoto, M. and Miki, T.: Asynchronous Media Casting Network: An Optimal Network Scheme for On-demand Video Distribution, Proc.17th Int'l. Conf. on Advanced Information Networking and Application, (Xi'an, China, Mar. 2003).