BOPPER: WIRELESS VIDEO BROADCASTING WITH PEER-TO-PEER ERROR RECOVERY

Shenjie Li *S.-H. Gary Chan*

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology, Hong Kong
{lisj, gchan}@cse.ust.hk

ABSTRACT

Wireless digital video broadcasting has experienced much success in recent years with some industrial systems deployed. We study in this paper how to recover lost packets in video broadcasting by means of a free broadcast-based secondary channel (such as IEEE-802.11 or bluetooth). In our scheme, termed BOPPER (Broadcasting with Peer-to-Peer Error Recovery), mobile devices collaboratively help each other to recover packet lost by broadcasting to their neighborhood. BOPPER achieves high scalability, low recovery delay, independence of a backward channel and low bandwidth cost. We study how to minimize the total number of retransmissions while achieving a certain level of recovery capability. We first consider the case with global knowledge (in terms of peer connectivity, loss status and pairwise loss rates of all peers) and formulate the problem as a linear program which can be solved efficiently. The result also serves as the optimum for our distributed protocol. We then propose and study a scalable, adaptive and distributed protocol which makes use of local information and message exchange to achieve efficient packet loss recovery. Simulation results show that our distributed scheme achieves close to the optimum with fast convergence time.

1. INTRODUCTION

With the advance of processing, memory and battery capabilities of mobile devices, it is now possible to stream video to the handhelds [1]. In recent years, there has been wide interest in wireless digital video broadcasting. Industrial standards, such as Digital Video Broadcasting (DVB) and Digital Multimedia Broadcasting (DMB), have been developed with systems deployed [2, 3]. In these systems, video streams are broadcast through a radio channel for playback at mobile devices.

In video broadcasting, packet loss is inevitable due to random bit errors, burst errors or transient outages in shadow regions. These losses have to be recovered as much as possible so as not to seriously affect video quality. To address this, the peer with packet loss (so-called *error peer*) may ask the server for retransmission via a separate backward wireless channel. By observing that nowadays mobile devices often come with a free secondary broadcast channel (such as IEEE 802.11 and bluetooth), we propose and study in this paper a more scalable and cost-effective peer-to-peer (p2p) error recovery scheme for digital video broadcasting. Our scheme, termed BOPPER (Broadcasting with Peer-to-Peer Error Recovery), makes use of peer-to-peer (p2p) broadcasting to recover the errors in one's neighborhood.

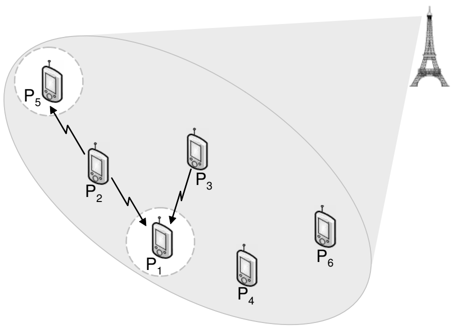


Fig. 1. P2P Error Recover for Wireless Video Broadcasting.

Figure 1 illustrates our idea. A server broadcasts video streams to wireless peers through a primary channel. Suppose peers P_1 and P_5 lose a packet while all the others successfully receive the packet. The peers P_2 and P_3 may rebroadcast (in a one-hop manner) their received packet by means of the secondary channel; as a result, the lost packet at P_1 and P_5 can be recovered. Note that P_1 may need to discard its duplicated packets. As compared with direct recovery from a server, there are several advantages in BOPPER:

- *Higher user scalability*: As the number of peers in the network increases, so is the number of losses. If each peer requests retransmission from the server, the server and network may be overwhelmed. In BOPPER, the retransmission load is distributed along all the peers. Furthermore, our approach is broadcast-based, and hence each retransmission may reach many peers simultaneously. This makes our system more scalable.

This work was supported, in part, by Direct Allocation Grant (DAG05/06.EG10), and the Innovation and Technology Commission of the Hong Kong Special Administrative Region, China (GHP/045/05).

1-4244-1017-7/07/$25.00 ©2007 IEEE

- *Lower recovery delay*: In BOPPER, lost packets are recovered directly by neighboring peers instead of by the server through the network infrastructure. Therefore, the recovery delay is lower.
- *Independence of a backward channel*: In some situations, due to the unavailability of a backward channel, peers may not be able to connect to the server for recovery. In such cases, neighbor recovery is a viable way to repair one's lost packets.
- *Lower bandwidth cost*: In general, a charge is incurred each time the backward channel is used for retransmission. Our approach achieves lower bandwidth cost for error recovery by making use of a free secondary channel.

Clearly, BOPPER would not be effective for sparse network (where there is no peer in neighborhood) or for a network where all neighbors share the same loss (such as the case of an extended regional outage). In these cases, the only option to recover lost packets is through a backward channel to the server. Therefore, BOPPER is a complementary strategy, rather than replacement of server recovery to achieve better performance.

To reduce flooding in the network, a peer with correctly received packet (so-called *successful peer*) retransmits its packet via broadcasting with a certain probability depending on the packet loss in its neighborhood. Clearly, the lower the loss in the neighborhood is, the lower is such retransmission probability. Given that energy consumption used for retransmission is an important consideration, we study in this paper how to minimize the number of retransmissions while meeting a certain level of recovery requirement (in terms of the average number of duplicate copies received at an error peer).

In this work, we first formulate the problem based on complete knowledge on peer connectivity, loss status and pairwise loss rates in the network. Our formulation is a linear program which can be solved efficiently. This serves as an optimum (lower bound on the number of retransmissions) in our study. We then propose a distributed protocol which makes use of only local information (such as the number of duplicate copies received) and local message exchanges to dynamically adjust one's retransmission probability. Our protocol effectively reduces the number of retransmissions in the system while meeting the recovery requirement. Simulation results show that the distributed protocol achieves close to the optimal performance.

There has been much work on error recovery for wireless video. Non-Causal Error Control (NCEC) is an error control scheme for wireless video multicast [4]. In NCEC, a secondary backward channel is used to transmit error control messages from peers to the server. We do not have such feedback in our system, and hence is more scalable. Collaborative Streaming among Mobiles (COSMOS), a multimedia content distribution protocol, delivers multimedia streams over wireless network by stream sharing [5]. The major issue of COSMOS is how peers take turns to achieve cost fairness and stream continuity, while the major issue of BOPPER is who should retransmit packets to meet a certain recovery objective. A distributed wireless error recovery approach using a secondary p2p network is proposed in [6]. In this approach, a non-adaptive retransmission suppression scheme is used; therefore, the residual loss rate increases with packet loss rate. Our system, on the other hand, is more adaptive in the sense that peers adjust their retransmission probability according to the channel condition so as to meet the recovery requirement. In [7], the authors propose an error recovery system for a satellite network which uses a terrestrial network to recover losses. This system requires a highly reliable secondary (terrestrial) channel, while our system works even when the loss rate in the secondary channel is high.

The rest of this paper is organized as follows. In Section 2, we present our formulation of the system. In Section 3, we propose the distributed protocol which achieves close to the optimum. In Section 4, we compare the numerical solution of our formulation (based on complete knowledge) with simulation. We conclude in Section 5.

2. PROBLEM FORMULATION

In this section, we formulate our optimization problem for p2p error recovery. We assume that complete network information is known, and hence our optimum (i.e., the minimum number of retransmissions) serves as a lower bound for the distributed protocol. We model the network as a graph $G(V, E)$, where V is the set of all peers in the network and E denotes the transmission links, that is, $(i, j) \in E$ if and only if peer j is in transmission range of peer i (and we call peer i a "neighbor" of peer j). Let $l_{i,j}$ be the packet loss rate between peers i and j in the secondary channel $\forall (i, j) \in E$. For a certain packet, let V_S be the set of all successful peers and V_E be the set of all error peers. Clearly, $V_S \cap V_E = \emptyset$ and $V_S \cup V_E = V$.

Let p_i be the probability of packet retransmission at peer $i, \forall i \in V_S$. The expected total number of packet retransmissions is obviously

$$\sum_{i \in V_S} p_i. \tag{1}$$

Note that the above is also an indication of energy consumption for error recovery in the p2p network. For an error peer j and its successful neighbor i, the probability that peer j correctly receives a recovery packet from the peer i is $(1-l_{i,j})p_i$, as the retransmission probability p_i and the channel loss probability $l_{i,j}$ are independent. The average number of copies that peer j receives is hence given by $\sum_{i \in n(j)}(1 - l_{i,j})p_i$, where $n(j) = \{i|(i, j) \in E, i \in V_S\}$ is the set of all successful neighbors of peer j. To meet a certain level of recovery capability, the number of copies has to be no less than a cer-

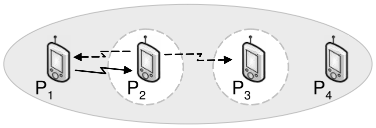


Fig. 2. NAK Suppression: dashed arrows denote NAKs; solid arrow denotes retransmission.

tain threshold θ_j, i.e.,

$$\sum_{i \in n(j)} (1 - l_{i,j}) p_i \geq \theta_j, \forall j \in V_E. \qquad (2)$$

Our problem is hence to minimize Eq.(1) subject to Eq.(2) and $0 \leq p_i \leq 1, \forall i \in V_S$, which is a linear program (LP) and can be solved efficiently. Note that in order to guarantee a solution, in our LP formulation, θ_j can not be larger than $\sum_{i \in n(j)} (1 - l_{i,j})$ (see Eq.(2)). In our LP simulation, therefore, if θ_j violates the condition (i.e., too few neighbors to help recovery), we adapt the best effort approach, i.e., set θ_j as large as possible equal to $\sum_{i \in n(j)} (1 - l_{i,j})$. However, in our simulation with the distributed protocol, θ_j does not need to be reset this way as our system can adaptively adjust p_i so as to meet θ_j as close as possible.

3. DISTRIBUTED PROTOCOL

In this section, we describe our distributed protocol which tries to minimize the average number of retransmissions in a network without the need of complete network knowledge. A peer with a lost packet broadcasts a NAK packet to all its neighbors (the term "broadcast" here means one-hop broadcast using the secondary channel). If an error peer hears a NAK from another peer, it suppresses its own NAK. Without suppression, the number of NAKs increases with the number of peers in neighborhood, which may overwhelm the channel. We introduce NAK suppression into BOPPER in order to reduce the number of NAKs.

This advantage comes with the cost that some lost packet may not be recovered. For example, in Figure 2, both peers P_2 and P_3 lose packet. Suppose P_2 sends its NAK first. This suppresses P_3. Due to power range, P_4 does not receive the NAK from P_2. Since the recovery packet broadcast from P_1 cannot reach P_3 while P_4, due to NAK suppression of P_3, is not aware of the loss at P_3, P_3 cannot recover its lost packet.

A peer j has a target protection level θ_j, and continuously calculate the average number of copies it receives $\bar{r}_j$. We use the exponential smoothing to calculate $\bar{r}_j$, i.e., after receiving r_j copies, $\bar{r}_j$ is updated according to $\alpha \bar{r}_j + (1 - \alpha) r_j$, where α is the smoothing factor between 0 and 1. In order to adaptively adjust the retransmission probability p_i of peer i depending on the loss environment, an error peer j piggybacks its NAK with a parameter $f_j = \theta_j / \bar{r}_j$. Clearly, if $f_j < 1$, the peer is "overprotected" with many copies, while $f_j > 1$ means that peer j is "underprotected." Peer i, after receiving f_j from its neighbor, adjusts its retransmission probability p_i by multiplying it by f_j. Our distributed protocol is detailed as follows:

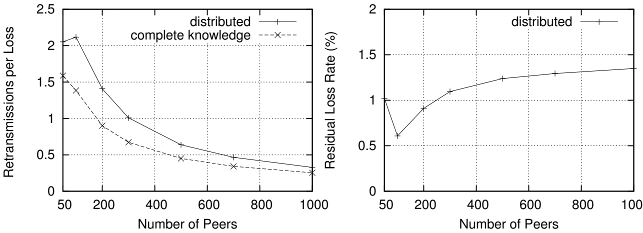
- *Error peer:* In order to meet the recovery requirement, we want an error peer with less protection (i.e., larger f_j) to send its NAK first so as not to be suppressed. Hence in our protocol, an error peer j, before sending its NAK, waits for a period that is shorter for larger f_j and vice versa (i.e., a decreasing function of f_j). Therefore, given the maximum waiting time W_{max}, peer j waits a random amount of time uniformly distributed between 0 and $\frac{1}{f_j+1} W_{max}$ (a decreasing function of f_j in the range between 0 and W_{max}, since $f_j > 0$). If no NAK is received from its neighbors in the waiting period, the peer broadcasts a NAK with f_j; otherwise, its NAK is suppressed.
- *Successful peer:* Our protocol favors less protected peers. Therefore a successful peer i receiving one or more NAKs uses the maximum f_j received to adjust its retransmission probability p_i according to $\min(f_j p_i, 1)$. Then the peer retransmits (i.e. broadcasts) the packet with probability of the adjusted p_i.Thus, if $f_j > 1$ (implying that peer j is "underprotected"), p_i increases so as to increase the expected number of copies received at peer j, and vice versa.

4. SIMULATION RESULTS

In this section, we present the numerical results of our LP solution and the simulation results of the distributed protocol. The peers are uniformly distributed in a rectangular area of 1500m × 1500m. The transmission range of peers is 250m. We assume that each packet is lost in the primary channel with a certain probability L_P independent of the other peers)though our study is by no means restricted to that). For simplicity, we consider that $l_{i,j} = L_S, \forall (i,j) \in E$, though our study can be extended for heterogeneous $l_{i,j}$. Unless otherwise stated, we use the following baseline parameters: $\theta_j = 2, \alpha = 0.5, |V| = 500, L_P = 0.1$, and $L_S = 0.1$.

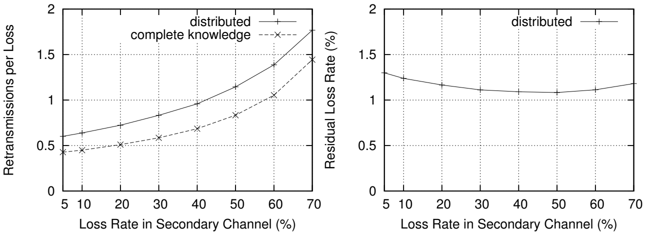
We broadcast packets to all peers through the primary channel given L_P. The LP formulation and distributed protocol are then applied on the same network topology and loss profile. We take statistics at steady state. The following performance metrics are obtained and averaged: (1) Retransmissions per loss, defined as the total number of retransmissions in the network divided by the total number of losses (i.e., the number of error peers); (2) Residual loss rate, defined as the loss rate of a peer after recovery.

In Figure 3a, we show the retransmissions per loss with respect to $|V|$. Clearly, as $|V|$ increases, the retransmissions per loss decreases, because a recovery packet may reach, and hence be shared with, more error peers. The LP solution (the



(a) Retransmissions per Loss. (b) Residual Loss Rate.

Fig. 3. Effects of Number of Peers $|V|$.



(a) Retransmissions per Loss. (b) Residual Loss Rate.

Fig. 4. Effects of Loss Rate in the Secondary Channel L_S.

line of complete knowledge) achieves the lowest number of retransmissions. Our distributed protocol achieves close to the optimum (by less than 50% penalty). The difference between two lines reflect the cost of a distributed protocol without complete knowledge. For sparse network ($|V|$ is low), there are some error peers without any recovery neighbor. This explains the relative low retransmissions per loss for low $|V|$ (e.g. $|V| = 50$). Figure 3b shows the residual loss rate versus $|V|$. In sparse network, error peers likely do not have recovery neighbors, and hence the residual loss rate is high.

Figure 4a shows the number of retransmissions per loss versus L_S. As L_S increases, more retransmissions are needed in order to meet the recovery requirement. Figure 4b shows the effects of L_S on the residual loss rate. Clearly, the residual loss rate does not change much with L_S. These two figures demonstrate the adaptability of BOPPER. When L_S increases, to ratain the average number of copies (and hence the residual loss rate), our distributed protocol re-transmits more recovery packets so as to meet a certain level of recovery capability.

Figure 5 illustrates the adaptability of our system by plotting the retransmissions per loss along time (in terms of the sequence number of packet). Starting with an arbitrary initial $p_i \in (0, 1], \forall i \in V$, our protocol quickly settles to a steady state. Our protocol favors less protected peers, hence "underprotected" peers quickly meet their recovery requirements. When all peers meet their recovery requirements, our protocol settles to a steady state close to the optimum. This shows that our protocol achieves fast convergence is achieved.

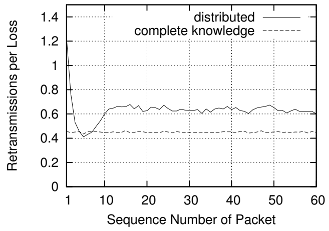


Fig. 5. Illustration of Evolution.

5. CONCLUSION

In this paper, we have presented BOPPER, which recovers lost packets in digital video broadcasting by using a free broadcast-based secondary wireless channel. BOPPER is scalable and cost-effective, and minimizes the expected number of retransmissions while meeting a certain level of recovery capability.

Given complete knowledge of the network, we have formulated our problem as a linear program which can be solved efficiently and serves as the optimum in a network without such complete knowledge. We then propose a scalable, adaptive and distributed protocol which uses local information and message exchanges to achieveclose to optimum performance. Simulation results show that our protocol achieves low residual loss rate, little retransmission overhead, and converges fast to a steady state.

6. REFERENCES

[1] S. Mohapatra, R. Cornea, N. Dutt, A. Nicolau, and N. Venkatasubramanian, "Integrated power management for video streaming to mobile handheld devices," in *Proc. 11th ACM ICMM*, Nov. 2003, pp. 582–591.

[2] "Digital video broadcasting," http://www.dvb.org/.

[3] "World DMB," http://www.worlddab.org/.

[4] I. V. Bajic, "Efficient error control for wireless video multicast," in *Proc. IEEE WMSP*, Oct. 2006, pp. 306–309.

[5] M.-F. Leung, S.-H. G. Chan, and O. C. Au, "COSMOS: Peer-to-peer collaborative streaming among mobiles," in *Proc. IEEE ICME*, July 2006, pp. 865–868.

[6] P. Sanigepalli, H. Kalva, and B. Furht, "Using p2p networks for error recovery in MBMS applications," in *Proc. IEEE ICME*, July 2006, pp. 1685–1688.

[7] E. Weigle, M. Hiltunen, R. Schlichting, V. A. Vaishampayan, and A. A. Chien, "Peer-to-peer error recovery for hybrid satellite-terrestrial networks," in *Proc. 6th IEEE P2P*, Sept. 2006, pp. 153–160.