# Vanets: Case Study of a Peer-to-peer Video Conferencing System

Tareq Hossain, Yi Cui, and Yuan Xue
Vanderbilt Advanced NETwork Systems Laboratory
Department of Electrical Engineering and Computer Science
Vanderbilt University, Nashville, TN 37212
Email: {tareq.hossain, yi.cui, yuan.xue}@vanderbilt.edu

## I. Introduction

Peer-to-peer (P2P) is a powerful platform for a variety of multimedia streaming applications over the Internet such as video-on-demand, video conferencing, live broadcasting, etc. A P2P system is extremely cost-effective since it utilizes the resources (CPU cycles, storage space, and uplink bandwidth) of peer machines.

Recently, there has been significant development in designing video conferencing systems. Video conferencing requires substantial network bandwidth in order to be able to transport high volume video data. This is particularly true for multi-party (MP) video conferencing system. One way to address the demand of high bandwidth is to use a network-based device called Multipoint Control Units (MCU). However, MCUs' high cost and maintenance complexities make it only feasible for large business enterprises. Alternatively, the low cost P2P video conferencing services allow participants to establish pair-wise video communication. However, using this technology in a MP video conferencing system greatly reduces the video quality experienced by the user.

The decentralized and shared bandwidth nature of a P2P network, therefore, makes it an effective choice for implementing video conferencing applications. In this work we present Vanets, a P2P based video conferencing system that takes advantage of transcoding to optimally allocate streaming rates for all participating peers. Our solution distinguishes between active and passive participants and enhances the video quality of the active participant.

The remainder of this proposal is organized as follows. In Sec. II, we present an overview of the Vanets interface, network model, tree construction algorithm, and video adaptation procedure. In Sec. III we present our demonstration plan.

## II. Implementation

Fig. 1 shows the proposed interface of Vanets. We identify a user as an active participant when the user is talking and allocate higher bandwidth compared to other passive participants. Inside the Vanets window, users receive streaming video from the active participant at a higher rate. In order to maintain the capacity constraint of a participant's uplink connection, we decrease the receiving rate from passive participants. Fig. 2 shows the basic network model of Vanets. We define each participant as *vanets entity*, which consists of a user, a video capture device and a video display device. Our physical topology is based on the assumption [1] that the bottleneck link of an end-to-end connection only happens at the uplink of the sending peer.


Fig. 1. Vanets interface

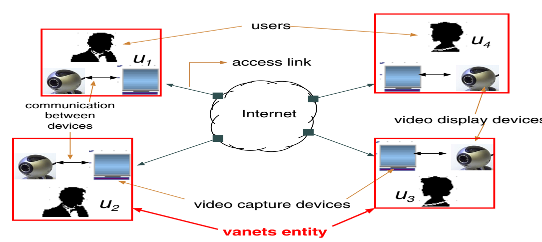


Fig. 2. Vanets network model

### A. Network Model

We consider a P2P network consisting of $H$ end hosts denoted as $\mathcal{H} = \{1, 2, ..., H\}$. For two peers $h_i$(parent) and $h_j$(child), the flow between them is defined as $f_{ij}$ and the relationship is defined as $h_i \rightarrow h_j$. We define $\mathcal{S}_i$ to be the set of outgoing flows for peer $h_i$. Each peer $h_i$, acts as a root of a unique balanced tree that contains all the participants. Peer $h_i$ uses the tree to send its active and passive stream. Having a unique tree for each peer ensures that a peer receives active/passive streams from all other peers. We also define $m$ to be the maximum number of outgoing flows for a root of a tree. In both active and passive situations, if a peer $h_i$ relays stream from its parent to its child, we collect such child peers in a set $\mathcal{T}_i$.

For an active peer $h_i$, let $a_h(h_i)$ be the maximum rate for high quality video it wishes to send to its children and $a_l(h_i)$

TABLE I
TREE CONSTRUCTION ALGORITHM

**Each end host** $h_i$
<u>Initialization</u>
$\mathbb{F}_i = \{h_j \,|\, \forall\, j,\, c_j \in \mathcal{C}_i\}$
$\mathcal{R}_i = \mathcal{C}_i$
$\mathcal{S}_i = \emptyset$
<u>Tree Construction</u>
**for** $i = 1$ **to** $m$
    pick $h_j \in \mathbb{F}_i$ such that $\forall\, c_k \in \mathcal{R}_i,\, c_j \geq c_k$
    $\mathcal{R}_i \leftarrow \mathcal{R}_i - c_j$
    $\mathbb{F}_i \leftarrow \mathbb{F} - h_j$
    $\mathcal{S}_i \leftarrow \mathcal{S}_i + h_j$
**end for**
**for** $m + 1$ **to** $H - 1$
    pick $h_j \in \mathbb{F}_i$ such that $\forall\, c_k \in \mathcal{R}_i,\, c_j \geq c_k$
    pick $h_k \in \mathcal{S}_i$ such that $\forall\, h_q \in \{\mathcal{S}_i \backslash h_k\}$
        **if** $|\mathcal{S}_k| = |\mathcal{S}_q|$ **then** $c_k \geq c_q$ **and**
        **if** $|\mathcal{S}_k| < |\mathcal{S}_q$ **then** $c_k \leq c_q$
    $\mathcal{R}_i \leftarrow \mathcal{R}_i - c_j$
    $\mathbb{F}_i \leftarrow \mathbb{F} - h_j$
    $\mathcal{T}_k \leftarrow \mathcal{T}_k + h_j$
**end for**

be the low quality video it expects to receive from its parents. Similarly, when a peer is passive, we define $p_h$ and $p_l$ to be the maximum rate allocated for receiving video streams from active and passive peers respectively. We also define $r_p$ to be the rate at which passive peer $h_i$ wishes to send its video stream to its children. Therefore, the capacity constraint for an active peer is:

$$m \cdot a_h + (H-1) \cdot a_l + t_r \cdot |\mathcal{T}| \leq c \quad (1)$$

where $c$ is the uplink bandwidth, $a_h = k \cdot a_l$ and $t_r$ is the relay rate for child $h_j \in \mathcal{T}$ that will be dynamically adopted with transcoding. Here, $k$ can be adjusted to achieve the desired $a_h$ and $a_l$. For a passive peer, the capacity constraint can be given as:

$$n \cdot p_h + (H-n-1) \cdot p_l + m \cdot r_p + t_r \cdot |\mathcal{T}| \leq c \quad (2)$$

where $n$ is the number of active participants. For a peer $h_i$, we also define $\mathcal{C}_i = \{c_j \,|\, \forall\, h_j \in \mathcal{H},\, h_i \neq h_j\}$ to be the set containing the capacity of all other peers.

### B. Tree Construction

Table I presents the tree construction algorithm. If the maximum number of hops between any two peers is $d$, then $m$ is defined as $m = \lceil \sqrt[d]{H} \rceil$. Our choice of $m$ is based on the fact that there are exponential number of shapes a tree can take. Therefore, by limiting the depth of a tree, we seek to create a balanced structure with higher capacity peers at the top of the tree. During the tree construction, the peers directly communicate with each other to receive the capacity information, relay peer information, etc.

Fig. 3 shows four possible unique tree configurations with four participants. In this example, When peer $A$ (Fig. 3(a)) is active, it sends high quality video to its children $B$ and $C$. Peer $B$ relays this video to peer $D$. Peer $A$ receives passive video from $B$ (Fig. 3(b)), which it relays (transcode if necessary to meet capacity constraint) to peer $C$. Peer $A$ also receives passive video from peer $D$ (Fig. 3(d)) and peer $C$ through peer $D$ (Fig. 3(c)). Therefore, peer $A$ uses the link capacity of peer $B$ to deliver high quality video to $D$. Since $B$ is a passive participant, it reduces its own video stream quality to meet the capacity constraint.

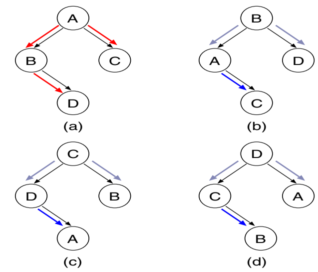


Fig. 3. Unique tree configuration for each peer

### C. Video Rate Adaptation

Each peer in the conference directly communicates with other peers to coordinate control information. Between a sending and a receiving peer, the transfer rate is determined by taking the minimum of the offered (by sender) and the requested (by receiver) rate. We use transcoding to adapt the video stream to the desired rate. In transcoding, the video signal is changed by the relaying peer to meet a lower encoding rate through either re-encoding or changing key parameters such as the quantization values. Transcoding is performed with X264 [2].

We also restrict the maximum number of hop counts between any two participants to two. This not only simplifies the design of our tree construction, but also ensures an acceptable level of latency during the conference session.

## III. DEMONSTRATION PLAN

Our demonstration will involve two participants on two laptop computers with webcam and internet connections. We will display the Vanets system with video adaptation capability. We will also display a referenced P2P conferencing system with no video adaptation where each recipient peer sends its own video stream of the same quality to all other participants.

The conferencing systems will involve a total 4 to 5 participants, i.e., the remaining participants will include actual users connected from our research lab at Vanderbilt University, and circulating video feeds from various PlanetLab [3] computers. The audience at the conference can also participate in the conferencing through the laptop computers at the demo site.

## REFERENCES

[1] M. Chen, M. Ponec, S. Sengupta, J. Li, and P. A. Chou, "Utility maximization in peer-to-peer systems," in *SIGMETRICS '08: Proceedings of the 2008 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. ACM, 2008, pp. 169–180.
[2] "Videolan." [Online]. Available: http://www.videolan.org
[3] "Planetlab." [Online]. Available: http://planet-lab.org