Journal of Mobile Multimedia, Vol. 5, No. 1 (2009) 045–063

DATA TRANSMISSION PROCEDURES FOR A MULTI-SOURCE STREAMING MODEL IN MOBILE PEER-TO-PEER (p2p) OVERLAY NETWORKS

ALIREZA GOUDARZI NEMATI
Seikei University, Dept. of Computers and Information Science
3-3-1 Kichijoji-kitamachi, Musashino, Tokyo 180-8633, Japan
alireza.gn@computer.org

MAKOTO TAKIZAWA
Seikei University, Dept. of Computers and Information Science
3-3-1 Kichijoji-kitamachi, Musashino, Tokyo 180-8633, Japan
makoto.takizawa@st.seikei.ac.jp

Received June 2, 2008
Revised September 1, 2008

In peer-to-peer (P2P) overlay networks, multimedia contents are in nature distributed to peers by downloading and caching. Here, a peer which transmits a multimedia content and a peer which receives the multimedia content are referred to as source and receiver peers, respectively. A peer is realized in a process of a computer and there are mobile and fixed types of computers. A peer on a mobile computer moves in the network. Furthermore, a peer maybe realized as a mobile agent. Thus, not only receiver peers but also source peers might move in the network. In this paper, we would like to discuss how source peers deliver multimedia contents to receiver peers in a streaming model so that enough quality of service (QoS) required is supported in change of QoS of network and peer, possibly according to the movements of the peers. In this paper, we discuss a multi-source streaming (MSS) protocol where a receiver peer can receive packets of a multimedia content from multiple source peers which can support enough QoS. If a current source peer is expected to support lower QoS than required, another source peer takes over the source peer and starts sending packets of the multimedia content. The receiver peer is required to receive packets of the multimedia content with enough QoS, e.g. no packet loss even if the source peer is being switched with a new source peer. We discuss how to switch source peers so as to support enough QoS to the moving receiver peer. We evaluate the MSS protocol in terms of the fault ratio, i.e. how frequently the receiver peer fails to receive packets with enough QoS and show the MSS protocol can reduce the fault ratio.

Keywords: Mobile multimedia streaming, QoS, P2P-model, acquaintance

1 Introduction

Information systems are now shifted to the peer-to-peer (P2P) model [23, 24] from the traditional client-server model. A system is composed of peer processes (abbreviated *peers*) interconnected in P2P overlay networks. Here, peers cannot only obtain multimedia service from other peers but also peers holding multimedia contents can support other peers with some types of objects like a part of the object. For example, a peer receives a part of a multimedia movie from a peer holding the movie. Another peer obtains a degraded monochromatic version of a fully-colored movie. Thus, objects are in nature distributed to peers with various ways like downloading and caching.

Peers are realized in types of computers including mobile devices like mobile phones and car navigation systems [12, 22]. If a peer is on a mobile computer, the peer is moving in the network. In addition, peers may be realized in mobile agents [13, 16]. Here, a peer itself moves from a compute to another computer. Thus, some peer is moving in a network. QoS (quality of service) supported by a peer and communication link among peers is dynamically changing due to not only congestions and faults but also movements of peers.

Suppose there is a receiver peer p_r which would like to listen to a music content c. A receiver peer p_r finds a source peer p_1 which can support with the music content c in a network by using types of discovery algorithms like types of flooding algorithms [17, 20] and DHT (distributed hash table) algorithms [11, 15, 21]. The receiver peer p_r starts receiving packets of the music c from the detected source peer p_1. The receiver peer p_r or the source peer p_1 may be moving in the network. This means QoS of a multimedia content c which the receiver peer p_r receives from the source peer p_1 is changing. If the receiver peer p_r finds QoS supported by the source peer p_1 to be degraded, another source peer p_2 is detected before QoS obtained from the source peer p_1 gets less qualified than required. The new source peer p_2 starts transmitting packets of the multimedia content c. The receiver peer p_r has to receive packets of the movie content with higher QoS than required even in a transition state where the current source peer is being switched. The new source peer p_2 has to be synchronized with the current source peer p_1 on what packets of the movie content c to send to the receiver peer p_r. In this paper, we discuss a protocol for multiple source peers to support a receiver peer p_r with a multimedia streaming service of enough QoS in change of QoS of network and source peer and in presence of movements of the peers. A mobile receiver peer p_r has to obtain enough QoS anywhere in a network. Suppose a receiver peer p_r is moving to some location. The receiver peer p_r has to find a source peer which can support enough QoS in the location before arriving the location is difficult, may be impossible for a receiver peer p_r to collect enough information on where source peers which can support a multimedia content of enough QoS are located in the network. In this paper, we take an acquaintance approach. Each peer p_r has acquaintance peers with which the peer p_r can directly communicate. A receiver peer p_r asks its acquaintance peers near to the location to find a source peer. If a source peer p_i is detected, an acquaintance peer informs the receiver peer p_r of the source peer p_i. The receiver peer p_r asks the source peer p_i to start sending packets of the multimedia content when the receiver peer arrives at the location.

Another type of the multi-source steaming model is discussed in the paper [2]. A multimedia content is divided into segments and each of the source peers sends a segment to a receiver peer p_r. A set of multiple source peers in parallel send segments to the receiver peer p_r. In addition, data in a multimedia content is redundantly distributed to segments so that the receiver peer p_r can receive every data in the multimedia content even if some number of packets are lost and some number of the source peers are faulty. In this paper, we discuss a model where one source peer sends packets of the multimedia content to a receiver peer at a time but another source peer takes over the source peer if the source peer cannot provide the receiver peer with enough QoS.

In the multi-source streaming model, a receiver peer p_r detects another source peer and negotiates with the source peer before QoS supported by the current source peer is too degraded. We evaluate the multi-source streaming model in terms of the fault ratio, i.e. how frequently a moving receiver peer p_r receives QoS which is lower than required. We show the fault ratio can be reduced in the simulation.

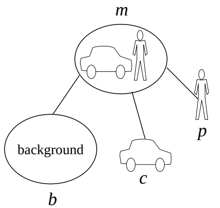
In section 2, we discuss the system model. In section 3, we discuss a multi-source streaming model. In section 4, we present how a mobile receiver peer finds a source peer with cooperation of the acquaintance peers. In section 5, we discuss a protocol for supporting a receiver peer with multimedia streaming service in presence of mobile peers. In section 6, we evaluate the multi-source streaming model in terms of fault ratio compared with a single-source streaming model.

2 System Model

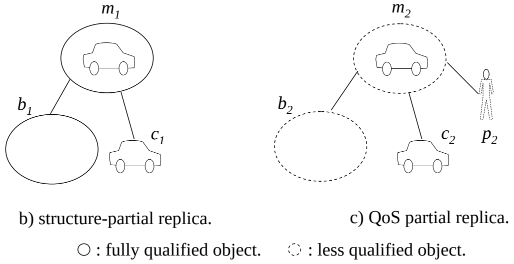
2.1 Multimedia contents

In distributed multimedia services, a receiver peer p_r just receives a multimedia content from a source peer. A source peer sends a multimedia content c to a receiver peer p_r if the source peer could provide the receiver peer p_r with the multimedia content c. There are *downloading* [5] and *streaming* [6, 7, 10] types of multimedia services to deliver multimedia contents to users. In the streaming service, it is critical for source peers to continuously deliver a multimedia content to receiver peers in real time manner. The streaming service is now more significant and economical than the downloading service.

Multimedia contents are distributed to peers with various ways. For example, suppose a peer p_i holds a fully-colored movie content c. Another peer p_j obtains the movie content c by downloading from the peer p_i. Then, the peer p_j obtains a full replica c_j of the multimedia content c as well as the peer p_i. Another peer p_k downloads the movie replica c_j from the peer p_j by degrading QoS of the movie content c to the monochromatic version c_k. The other peer p_h obtains only some scenes of the movie content c as the replica c_h. A multimedia content is hierarchically composed of objects as discussed in MPEG4 [1]. A multimedia content is characterized in terms of *part-of* relations of multimedia objects and QoS of each object as shown in Figure 1. If a replica c_i has the same object structure and QoS as an original multimedia content c, the replica c_i is referred to as a *full* replica of the multimedia content c. Otherwise, c_i is a *partial* replica of the multimedia content c. There are further types of partial replicas of a movie content. If a replica c_i has only a part of the multimedia content c like some scenes of a movie, c_i is referred to as an *structure-partial replica* of the multimedia content c. If the replica c_i supports lower QoS than the multimedia content c, c_i is a *QoS-partial replica* of the content c. For example, a monochromatic replica c_j is a QoS-partial replica of a fully-colored movie content c. The replica c_j is less qualified than the multimedia content c although the replica c_j has the same structure as the multimedia content c. Suppose a movie content m is composed of a background object b, *car* object c, and *person* object p as shown in Figure 1 a). Figure 1 b) and c) show structure-partial and *QoS*-partial replicas m_1 and m_2 of the movie content m, respectively. The structure-partial replica m_1 contains the background of object b_1 and car object c_1 but no person object p. On the other hand, the QoS-partial replica m_2 includes all the objects b_2, c_2, and p_2 but QoS of the background object b_2 is degraded and QoS of the person object p_2 is also degraded.



a) Content *m*.



b) structure-partial replica. c) QoS partial replica.

○ : fully qualified object. ◌ : less qualified object.

Fig. 1. Types of replicas.

2.2 Movement of peer

A system is composed of two hierarchal layers, *physical* and *overlay* layers. The physical layer shows an underlying network which is composed of nodes interconnected in wired and wireless links. A node is a computer which is at a fixed location or is mobile. A mobile node communicates with another node in a wireless network [8]. We assume every node can communicate with every other node by using communication protocols like mobile IP [8] even if nodes are moving. However, the communication link between a pair of nodes may not necessarily support enough and invariant QoS due to congestions and faults. The overlay layer indicates an overlay network on the physical layer, which is logically composed of peer processes (*peers*) interconnected in logical communication links. A peer p_i is realized as a process in a node D_s. The node D_s is referred to as *supporting* node of the peer p_i.

If the supporting node of a peer is moving in the underlying network, the peer is referred to as *move*. In another point, a peer itself is moving around nodes in a network if the peer is realized in a mobile agent [13, 16]. Thus, there are two types of mobile peers; *indirectly mobile* peer which is supporting node moves and *directly mobile* peer which itself moves around nodes. A mobile peer can manipulate a multimedia content in a supporting node. QoS supported by a mobile peer depends on the supporting node. We assume that every pair of peers can communicate with one another in types of networks. Here, even a directly mobile peer can make a communication link with any peer anytime and anywhere. For example, if a supporting node of a peer p_i is moving and is getting farther from the access point in mobile network, the peer may lose some packets due to noise. Thus, QoS obtained by the peer may change due to not only congestions and faults in the underlying network but also movements of peers. In this paper, we model the *movement* of a peer to mean the change of QoS.

2.3 Quality of service (QoS)

It is critical to discuss how much qualified multimedia content a receiver peer can obtain from a source peer in a network. QoS in the underlying network is shown by parameters,

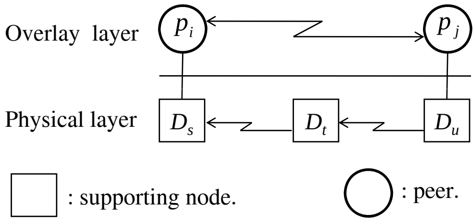


Fig. 2. Overlay and physical layers.

bandwidth, delay, and packet loss ratio [4, 3]. At the overlay layer, QoS means application level QoS of the multimedia content whose parameters are frame rate, resolution, number of colors, quality of sound, and so on depending on types of multimedia contents. A *QoS instance* is thus denoted by application level QoS parameters $\langle q_1, \ldots, q_m \rangle$. Thus, each QoS instance is in a multi-dimensional form. Hence, QoS instances are partially ordered. In this paper, we assume that for every pair of QoS instances Q_1 and Q_2, we can decide which one is higher than or equal to the other for simplicity. Here, "$Q_1 > Q_2$" means that Q_1 is higher than Q_2 and "$Q_1 = Q_2$" indicates that Q_1 and Q_2 are equivalent. $Q_1 \geq Q_2$ means $Q_1 > Q_2$ or $Q_1 = Q_2$.

Let QC_{ic} be *quality of content* (QoC), which is QoS of a multimedia content c which a source peer p_i holds. Suppose a peer p_i holds a multimedia content c with QoS QC_{ic}. Even if the multimedia content c itself supports higher QoS, enough QoS cannot be obtained from the peer p_i if the peer p_i is overloaded. Let QP_i indicate the *peer QoS* of a peer p_i. Even if a peer p_i holds a high quality multimedia content c, another peer cannot take enough QoS if the peer p_i itself is overloaded. Let QPC_{ic} be QoS with which the multimedia content c can be obtained at a peer p_i. Here, QPC_{ic} is a minimum one of QC_{ic} and QP_i $QPC_{ic} = min(QC_{ic}, QP_i)$. Let QN_{ij} show QoS of a network link between a pair of peers p_i and p_j. If the link is congested, QN_{ij} is degraded. Even if QPC_{ic} is enough QoS, another peer p_j cannot obtain enough QoS if the network does not support enough QoS. The peer p_j finally obtains the *service QoS* Q_{ijc} of the content c from the peer p_i through the network. The server QoS Q_{ijc} is given to be $min(QN_{ij}, QPC_{ic})$.

In this paper, we assume the peer QoS QP_i of a peer p_i and the multimedia content QoS QC_{ic} of a multimedia content c are invariant for simplicity even if the supporting node of the peer p_i is moving in the underlaying network. Here, only the network QoS QN_{ij} among a pair of peers p_i and p_j changes. On the other hand, the peer QoS QP_i of a directly mobile source peer p_i depends on which node supports the peer p_i. Here, not only QN_{ij} but also QP_i and QC_{ic} are changing.

3 A Multi-Source Streaming (MSS) Model

Let p_r be a receiver peer and p_i^c be a source peer which holds a replica c_i of a multimedia content c $(i = 1, \ldots, m_r, m_r \geq 1)$. Some content replica c_i may be a full replica of the multimedia content c. Another replica c_j may be only a part of the content c, i.e. structure-partial replica. The other replica c_k may include all the objects of the multimedia content c

but may be less qualified, i.e. QoS-partial replica. Thus, a source peer p_i^c may have a type of replica of a multimedia content c.

The receiver peer p_r first finds source peers of a target multimedia content c from which the receiver peer p_r can obtain enough QoS. Let RQ_{rc} be the minimum QoS of a multimedia content c required by the receiver peer p_r. Let SP_{rc} be a set of source peers which support a multimedia content c of higher service QoS than RQ_{rc}, i.e. $SP_{rc} = \{p_i^c \mid Q_{irc} \geq RQ_{rc}\}$. Here, let SP_{rc} be a set $\{p_1^c, \cdots, p_{m_r}^c\}$ $(m_r \geq 1)$ of source peers as shown in Figure 3. Suppose the receiver peer p_r takes a source peer p_i^c in the source peers set SP_{rc} and receives the multimedia content c from the source peer p_i^c. Here, the source peer p_i^c is referred to as *current* source peer of the receiver peer p_r, which is now supporting the content replica c_i to the receiver peer p_r. The current source peer p_i^c starts transmitting packets of the multimedia content c to the receiver peer p_r. If a source peer p_i^c may move in a network as well as a receiver peer p_r, the content QoS Q_{irc} which the receiver peer p_r obtains from the source peer p_i^c changes depending on QoC of the supporting node. If $Q_{irc} < RQ_{rc}$, the receiver peer p_r cannot obtain enough QoS from the source peer p_i^c even if the receiver peer p_r receives the multimedia content c. Hence, the current source peer p_i^c has to be switched with another source peer p_j^c which can support enough QoS to the receiver peer p_r, i.e. $Q_{rjc} \geq RQ_{rc}$. Thus, one of source peers $p_1^c, \cdots, p_{m_r}^c$ $(m \geq 1)$ necessarily supports the receiver peer p_r with the streaming service of the multimedia content c of enough QoS while current source peers are switched.

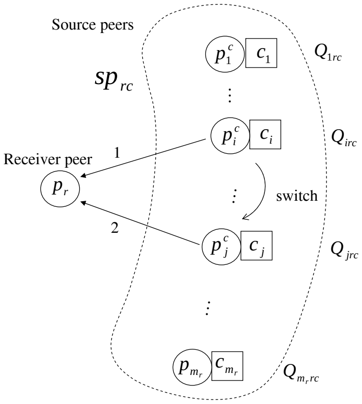


Fig. 3. Multi-source streaming model.

A multimedia content c sent by a source peer p_i^c at the overlay layer is transmitted in a sequence of bits which are in fact carried by packets in a network at the physical layer. Let BS_{ic} show a bit sequence of a replica c_i which a source peer p_i^c transmits to the receiver p_r. We assume that every source peer p_i^c translates a multimedia content c to a bit sequence BS_{ic} in the same algorithm. Here, if a pair of content replicas c_i and c_j of the multimedia content c are the same, a pair of the source peers p_i^c and p_j^c send the same bit sequences, $BS_{ic} = BS_{jc}$ to the receiver peer p_r in the network. Otherwise, a pair of the source peers p_i^c and p_j^c send different bit sequences, $BS_{ic} \neq BS_{jc}$.

Let $|BS_{ic}|$ be the total bit length of the bit sequence BS_{ic}. BT_{ic} indicates the total time to play the multimedia content c obtained from the bit sequence BS_{ic}. Here, $BT_{ic} = BT_{jc}$ if a pair of the content replicas c_i and c_j are in the same structure and support the same QoS. Let BR_{ic} show the bit rate of the content replica c_i, $BR_{ic} = |BS_{ic}|/BT_{ic}$. Let $BS_{ic}|_{\tau_2}^{\tau_1}$ be a subsequence of the bit sequences BS_{ic} to be played from time τ_1 to τ_2 as shown in Figure 4. In our steaming model, a current source peer p_i^c is taken over by another source peer p_j^c if the current source peer p_i^c could not support the receiver peer p_r with enough QoS, i.e. $Q_{irc} < RQ_{rc}$. Suppose a source peer p_i^c starts transmitting a bit sequence BS_{ic} at time τ_0. At time τ_1, another source peer p_j^c takes over the current source peer p_i^c. Let BS_{ic}^1 be a subsequence $BS_{ic}|_{\tau_1}^{\tau_0}$ of the bit sequence BS_{ic}. The other source peer p_k^c takes over the source peer p_j^c at time τ_2 as shown in Figure 5. Let BS_{jc}^2 and BS_{kc}^2 show the bit sequences $BS_{jc}|_{\tau_2}^{\tau_1}$ and $BS_{kc}|_{\tau_3}^{\tau_2}$, respectively. Suppose the source peers p_i^c, p_j^c, and p_k^c holds QoS-partial content replicas c_i, c_j and c_k with different QoS instances Q_{irc}, Q_{jrc}, and Q_{krc} to the receiver peer p_r, respectively. Suppose the network bandwidth QN_{ir} between a pair of the peers p_r and p_i^c and QN_{kr} between a pair of the peers p_r and p_k^c are so high that a fully colored movie can be transmitted to the receiver peer p_r. However, the network QoS QN_{jr} between a pair of peers p_r and p_j^c is too slow from the source peer p_j^c to transmit the fully colored movie to the receiver peer p_r. The source peer p_j^c can transmit only a monochromatic version c_1 of the fully colored movie content c. At the receiver peer p_r, the movie content c should be played without losing frames even if the QoS-partial replica c_1 is not fully colored.

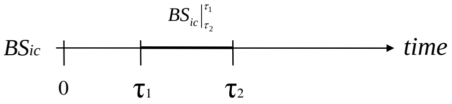


Fig. 4. Bit sequence of a content.

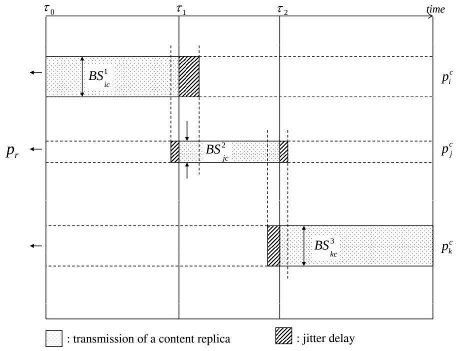


Fig. 5. Multi-source streaming.

As shown in Figure 5, the receiver peer p_r receives a bit subsequence BS_{ic}^1 from one source peer p_i^c. Then, the receiver peer p_r receives the bit sequence BS_{jc}^2 from another source peer p_j^c. The bit sequences BS_{ic}^1 and BS_{jc}^2 have to be received by the receiver peer p_r in a seamless manner. We have to synchronize the bit sequence BS_{ic}^1 with the bit sequence BS_{jc}^2. Since it takes time to synchronize the bit sequences, i.e. jitter delay, the source peer p_i^c sends the bit sequence BS_{ic} for a longer time than $\tau_1 - \tau_0$, i.e. $BS_{ic}|_{\tau_1+\delta}^{\tau_0}$. The source peer p_j^c starts sending the bit sequence BS_{jc} earlier than time τ_1. That is, the source peer p^c_j sends a bit subsequence $BS_{jc}|^{\tau_1-\delta}_{\tau_2+\delta}$ from the time $\tau_1 - \delta$ to $\tau_2 + \delta$. The receiver peer p_r receives redundantly bits of the multimedia content c in the bit subsequences $BS_{ic}|^{\tau_1-\delta}_{\tau_1}$ and $BS_{jc}|^{\tau_1}_{\tau_1+\delta}$ from the both the source peers p^c_i and p^c_j. The duration from time $\tau_1 - \delta$ to time $\tau_1 + \delta$ is referred to as *overlapping time* of the source peers p^c_i and p^c_j. In the overlapping time, the source peer p^c_j has to synchronize the transmission of the multimedia content c with the current source peer p^c_i so that the receiver peer p_r can receive the content c with required QoS.

4 An Acquaintance-based Model for a Mobile Receiver Peer

4.1 Acquaintance peers

Suppose a receiver peer p_r is moving in the network while receiving packets of a multimedia content c from the current source peer p^c_i. It takes time for the receiver peer p_r to detect another source peer p^c_j and to start receiving packets of the content c from the source peer p^c_j. The receiver peer p_r has acquaintance peers [18, 19] with which the receiver peer p_r can directly communicate. First, the receiver peer p_r asks acquaintance peers which know about the service QoS of a target source peer. The acquaintance peers are nearer to a target node to which the receiver peer p_r is planing to move. Let A_i be a set of acquaintance peers of a peer p_i.

For each acquaintance peer p_j in the acquaintance set A_i, the peer p_i holds acquaintance information $\langle p_j,\ QP_j,\ QN_{ij},\ QC_{jc},\ time\rangle$ where QP_j, QN_{ij}, and QC_{jc} indicate the peer QoS of p_j, network QoS between p_i and p_j, and content QoS of c_{jc}, respectively and *time* shows when the acquaintance peer p_j obtains the information. Here, QC_{jc} = "-" if the acquaintance peer p_j does not hold a multimedia content c. A pair of peers exchange its acquaintance information with one another. A peer p_i sends an inquiry message to the acquaintance peer p_j. On receipt of the inquiry message, the acquaintance peer p_j sends an acquaintance information $\langle p_k, QP_k, QN_k, QC_{kc}, t\rangle$ of its acquaintance peer p_k to the peer p_i. On receipt of the acquaintance information of p_k from the peer p_j, the peer p_i stores the information $\langle p_k, QP_k, QN_{ik}, QC_{kc}, t'\rangle$ in its database DB_i where $t' = max(t$, current time of $p_i)$ and $QN_{ik} = min(QN_{ij}{+}QN_k$, current $QN_{ik})$. Here, the peer p_j gets an acquaintance of the peer p_i. If the peer p_k is already an acquaintance of the peer p_i, the peer p_i updates the information of the peer p_k with the acquaintance information $\langle p_k, QP_k, QN_{ik}, QC_{kc}, t'\rangle$ if $t > t'$ where t' is current time. Since the memory size of the database DB_i is limited, obsolete acquaintance information have to be thrown away to make a space to store new acquaintance information. Here, the oldest acquaintance information is removed in the database DB_i. Let DB_{ci} be a set of acquaintance information of a multimedia content c in the database DB_i.

Each peer p_i obtains the network QoS QN_{ij} with an acquaintance peer p_j, for example, by periodically sending PING messages [9] to the acquaintance peer p_j. Each peer p_i is assumed to be able to obtain information on the peer QoS QP_i.

The receiver peer p_r selects a source peer p^c_i by using the acquaintance information in the database DB_i as follows:

[**Source peer selection**]

1. Source peers of the multimedia content c whose service QoS is greater than the requirement QoS RQ_{rc} are found in the database DB_{ci}, i.e. $SP_{rc} = \{p^c_i \mid \langle p^c_i,\ QP_i,\ QN_{ri},\ QC_{ic},$

$t_i\rangle$ in the database DB_i where $QC_{ic} \neq$ "-", and $Q_{irc} = min(QP_i,\ QN_{rc},\ QC_{ic}) \geq RQ_{rc}\}$.

2. If $|SP_{rc}| \geq 2$, a source peer p_i^c where time t_i is the largest in SP_{rc} is taken as a current source peer.

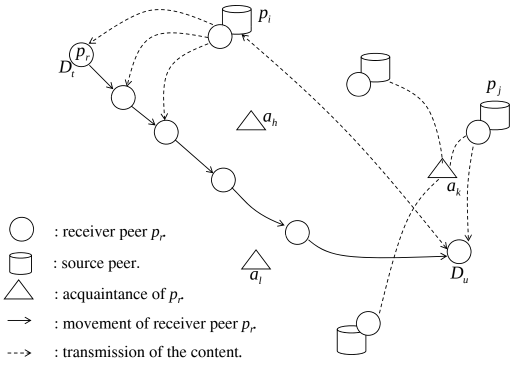


Fig. 6. Acquaintance peers.

In Figure 6, a moving receiver peer p_r on a node D_t is receiving packets of a multimedia content c from a source peer p_i^c. Suppose the receiver peer p_r would like to move to a node D_u. The receiver peer p_r finds an acquaintance peer a_k in the acquaintance set A_k which is nearer to the destination node D_u. The receiver peer p_r asks the acquaintance peer a_k to find a source peer p_j^c which can support the streaming service to the receiver peer p_r which satisfies the required QoS RQ_{rc}. Then, the receiver peer p_r asks the source peer p_j^c to send packets of the multimedia content c to the node D_u when the receiver peer p_r arrives at the node D_u. As written in the example, a moving receiver peer has to receive the multimedia content c with enough QoS anywhere in the network. If the receiver peer p_r knows a location where to go, source peers which can support the multimedia content c with enough QoS at the location have to be found.

Suppose a receiver peer p_r is moving and has a route RT_r by which the receiver peer p_r moves in the network. The route RT_r is realized in a sequence of wireless communication cells.

[Source peer detection]

1. The receiver peer p_r sends the route RT_r to its acquaintance peers in the acquaintance set A_r.

2. On receipt of the route RT_r from the receiver peer p_r, an acquaintance peer a_t of the receiver peer p_r looks for cells in the route RT_r.

3. If the acquaintance peer a_t is in a cell d of the route RT_r, the acquaintance peer a_t finds a source peer p_i^c by using the source peer selection algorithm presented here. The source peer p_i^c can support a receiver peer with the multimedia content c of enough QoS if the receiver peer is in the cell d.

4. The acquaintance peer a_t sends a source peer information $\langle d,\ p_i^c,\ QP_i,\ QN_{it},\ QC_{ic},\ \tau_i\rangle$ to the receiver peer p_r. The receiver peer p_r collects the information $\langle d,\ p_i^c,\ QP_i,\ QN_{it},\ QC_{ic},\ \tau_i\rangle$ for each cell in the route RT_r. If there are multiple information $\langle d,\ p_i^c,\ QP_i,\ QN_{it},\ QC_{ic},\ \tau_i\rangle$ and $\langle d,\ p_j^c,\ QP_j,\ QN_{iu},\ QC_{jc},\ \tau_j\rangle$ for a cell d, the source peer p_j^c is taken if $Q_{tic} > Q_{ujc}$.

5. The receiver peer p_r estimates time t_a when the receiver peer p_r will arrive at the cell d. The receiver peer p_r sends a content request to the source peer p_i^c in some time units before the estimated arrival time st_d. The receiver peer p_r sends a content request to the source peer p_i^c before moving to the cell d.

6. On receipt of the content request from the receiver peer p_r, the source peer p_i^c sends an ACK message to the receiver peer p_r if the source peer p_i^c could support the receiver peer p_r with the multimedia content c. Otherwise, the source peer p_i^c sends a NAK message to the receiver peer p_r.

5 Data Transmission Procedures

5.1 Receiver-source communications

A receiver peer p_r finds a source peer p_i^c which can support the receiver peer p_r with the streaming service of a multimedia content c with enough QoS, i.e. $Q_{irc} \geq RQ_{rc}$. While receiving packets of the multimedia content c from the current source peer p_i^c, the receiver peer p_r monitors the current QoS Q_{ic} and the network QoS QN_{ir} supported by the source peer p_i^c and the network, respectively. Then, the receiver peer p_r obtains service QoS $Q_{irc} = min(Q_{ic}, QN_{ir})$ of the source peer p_i^c. If the service QoS Q_{irc} is expected to get degraded to be lower than the required QoS RQ_{rc}, the receiver peer p_r finds another source peer p_j^c which can support enough service QoS Q_{jrc}, i.e. $Q_{jrc} \geq RQ_{rc}$.

We discuss how to coordinate a receiver peer p_r and multiple source peers $SP_r = \{p_1^c, \ldots, p_{m_r}^c\}$ $(m_r \geq 1)$ so that the receiver peer p_r can receive packets of a multimedia content c from one of the source peers. First, the receiver peer p_r finds source peers p_1^c , …, $p_{m_r}^c$ which support the multimedia content c in a P2P overlay network by using some P2P discovery algorithm [14, 20]. The receiver peer p_r selects one source peer, say p_i^c whose service QoS Q_{irc} satisfies the application requirement QoS RQ_{rc}, $Q_{irc} \geq RQ_{rc}$. Then, the receiver peer p_r sends a *content* request *Creq* to the source peer p_i^c. The content request *Creq* includes parameters $\langle \tau, RQC_{ic}, RQ_{rc}\rangle$ where RQC_{ic} is the required QoC of the multimedia content c and RQ_{rc} is the service QoS which the receiver peer p_r would like to receive from the source peer p_i^c. τ shows from what bit in the bit sequence BS_{ic} the source peer p_i^c should send. Here, $\tau = 0$, i.e. the source peer p_i^c sends bits from the first bit of BS_{ic}. If the source peer p_i^c agrees on serving the multimedia content c to the receiver peer p_r, the source peer p_i^c sends an ACK message with the peer QoS QP_i and the content QoS QC_{ic} to the receiver peer p_r. Then, the receiver peer p_r calculates the service QoS Q_{irc} from QP_i, QC_{ic}, and QN_{ir} and sends a *Start* message with start time τ_s and end time τ_e to the source peer p_i^c. Then, the source peer p_i^c starts transmitting packets of the multimedia content c. τ_s is grate $\tau - \delta$ as discussed in the preceding section. Here, τ_s is 0 and τ_e is time from the beginning to the end of the multimedia content. The receiver peer p_r starts receiving packets from the *current* source peer p_i^c. If the source peer p_i^c sends every packet of the multimedia content c required, i.e.

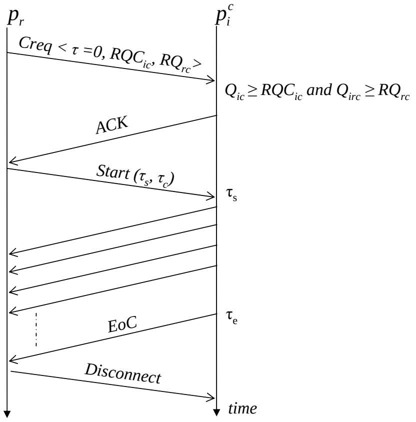


Fig. 7. Receiver source communication.

from time τ_s to τ_e, the source peer p_i^c sends an *end-of-content (EoC)* message to the receiver peer p_r. Then, the receiver peer p_r sends a *disconnect* request to the source peer p_i^c.

During the transmission of packets from a source peer p_i^c, the service QoS Q_{irc} might be degraded, e.g. due to the movement of the receiver peer p_r. First, suppose the receiver peer p_r finds the degradation of the service QoS from the source peer p_i^c. The receiver peer p_r calculates the service QoS Q_{irc} each time the receiver peer p_r receives a packet from the source peer p_i^c. Even if $Q_{irc} \geq RQ_{rc}$, a source peer p_i^c is *suspicious* if $Q_{irc} \leq TRQ_{ri}$. $TRQ_{ri} = \alpha * RQ_{rc}$ where $\alpha(\geq 1)$ is a constant. If the source peer p_i^c is suspicious for some time units, the source peer p_i^c is *dangerous*. If $Q_{irc} < RQ_{rc}$, the source peer p_i^c is *faulty*. If a current source peer p_i^c gets dangerous, the receiver peer p_r finds another source peer p_j^c. It takes time T to switch a source peer since the receiver peer has to find another source peer p_j^c and do negotiation with the source peer p_j^c. At each time t, the receiver peer p_r estimates *breaking time* bt_t when the receiver peer p_r could not receive enough QoS from the current source peer p_i^c if the QoS Q_{irc} supported from the source peer p_i^c is being degraded. Hence, the condition $|bt_t - t| > T$ is required to be satisfied. Otherwise, the current source peer is switched with another source peer without obtaining enough QoS. Here, let $Q_{irc}(t)$ show QoS Q_{irc} which is supported by a source peer p_i^c at time t. The threshold value TRQ_{ri} is obtained as follows:

$$TRQ_{ri} = Q_{irc}(t_1),\ t_1 > t,\ \text{and}\ (bt_t - t_1) \geq T.$$

Next, if the source peer p_i^c could not support the receiver peer p_r with enough QoS, the source peer p_i^c sends a *retire* message to the receiver peer p_r. On receipt of the *retire* message, the receiver peer p_r tries to find another source peer. If the receiver peer p_r finds another peer p_j^c and the source peer p_j^c starts sending packets, the receiver peer p_r sends a *disconnect* request to the source peer p_i^c.

5.2 Receiver peers

Suppose that a receiver peer p_r is receiving packets of a multimedia content c from a current source peer p_i^c. Both the receiver peer p_r and the source peer p_i^c monitor the service QoS

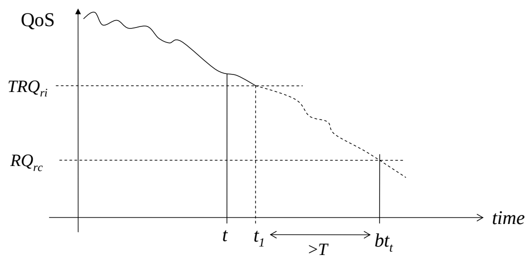


Fig. 8. Breaking time.

Q_{irc} obtained from the source peer p_i^c through the network and QoS QPC_{ic} which the source peer p_i^c is supporting, respectively. Suppose the receiver peer p_r detects that QoS supported by the source peer p_i^c is expected to be under the required QoS ($Q_{irc} < RQ_{rc}$) some time units τ later. Let δ_{ri} be delay time between the receiver peer p_r and the source peer p_i^c. τ has to be longer than $2 * \delta_{ri}$. In fact, if the some QoS Q_{irc} gets smaller than some QoS value TQ_r, the receiver peer p_r considers the source peer p_i^c to be suspicious. If the source peer p_i^c is suspicious for λ time units, the receiver peer p_r starts finding another source peer p_k^c. Here, the receiver peer p_r finds another source peer p_k^c of the multimedia content c and sends a *change* request to the source peer p_k^c. On receipt of the *change* request, the source peer p_k^c sends an ACK message to the receiver peer p_r if the source peer p_k^c could send the multimedia content c where $RQ_{rc} \leq Q_{kc}$. Then, the receiver peer p_r sends a *Start* request to the source peer p_k^c. Here we assume that every source peer p_k^c has a full replica c_k of a content c for simplicity. That is, $BS_{ic} = BS_{jc}$ for every pair of source peers p_i^c and p_k^c.

A multimedia content c is realized as a sequence $P(c)$ of packets, $pk_1, \cdots, pk_q (q \geq 1)$ to be transmitted in the network. Each packet carries a subsequence of the bit sequence BS. Each packet pk is identified in the packet sequence number $pk.seq$. Here, $pk_x.seq = x$. In the receiver peer p_r, a variable REQ shows the sequence number seq of a packet of the multimedia content c which the receiver peer p_r has most recently received from a source peer p_i^c. The receiver peer p_r sends a *Start* request with sequence number REQ to the source peer p_k^c. On receipt of the *Start* request, the source peer p_k^c calculates the delay time δ_{rk} to the receiver peer p_r and the minimum transmission rate $minTR_{rk}$ which is decided by QoS of the multimedia content c. Then, the sequence number seq of a packet which the source peer p_k^c starts transmitting is calculated as $seq = REQ + minTR_{rk} * \delta_{rk}$. Then, the source peer p_k^c starts transmitting packets from a packet pk_{seq} as shown in Figure 9. The receiver peer p_r receives packets from the source peer p_k^c while receiving packets from the current source peer p_i^c:

```
On entering a suspicious state, {
    $count = 0$;
    $SREQ = REQ$;
    $p_k^c$ = Find-New-Source ($c$); /* find another source peer */
    $SEQ = REQ + minTR_{rk} * \delta_{rk}$; /* sequence number of packets which starts transmitting to $p_k^c$ */
    send start to $p_k^c$;
```

```
    }
On receipt of a packet pk from $p_k^c$, {
    if current $\neq p_k^c$, {
      if $REQ > pk.seq$,
        discard pk; /* packet pk is discard */
      else if $REQ = pk.seq$, {
        receive pk;
        $REQ = REQ + 1$;
        $count = count + 1$;
        /* change the source peer to $p_k^c$ and stop $p_i^c$; */
        if $REQ - SREQ > minPN$ and count/$(REQ - SREQ) \geq 0.9$ {
          send disconnect to current;
          current = $p_k^c$;
          }
        }
        else discard pk; /* $REQ \neq pk.seq$ */
      }
    }
    else /* current = $p_k^c$ */
      if $REQ = pk.seq$, {
        receive pk;
        $REQ = REQ + 1$; }
      else if $REQ < pk.seq$ {
        $REQ = pk.seq + 1$;
        discard pk; }
    else /* $REQ > pk.seq$ */
      discard pk;
    }
```

Here, *minPN* shows the number of packets. After receiving the number *minPN* of packets since sending a *Start* message to the source peer p_k^c if the receiver peer p_r receives packets in order from the source peer p_k^c, the receiver peer p_r changes the current source peer with the source peer p_k^c.

5.3 Source peers

If the receiver peer p_r decides on changing the current source peer, the receiver peer p_r sends a *Disconnect* request to the current source peer p_i^c. On receipt of the *Disconnect* request from the receiver peer p_r, the source peer p_i^c stops transmitting packets to the receiver peer p_r in the following procedures:

```
[Receiver peer $p_r$]
    $p_r$ finds another source peer $p_i^c$;
    $p_r$ sends a Content request Creq with $QC_{ic}$ and $RQ_{rc}$ to $p_i^c$;
    On receipt of an ACK message from $p_i^c$, {
      current = $p_i^c$; /* current server is $p_i^c$ */
      $REQ = 1$;
```

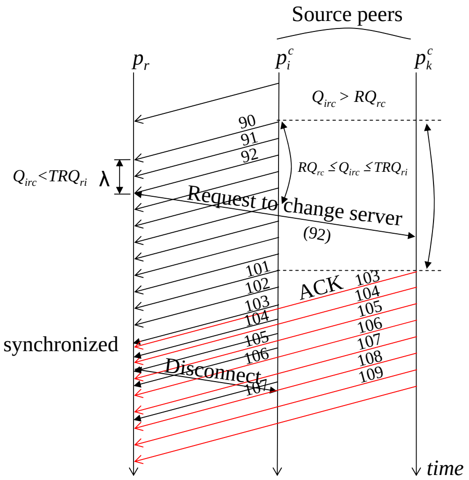


Fig. 9. Switch of source peer.

```
    send a Start message to $p_i^c$;
}
On receipt of a NACK message from $p_i^c$, {
    find another source peer;
    go to the beginning of this procedure;
}
```

A source peer p_i^c behaves on receipt of request from the receiver peer p_r as follows:

```
[Source peer $p_i^c$ ]
    On receipt of a Content request Creq with QoS information $QC_{ic}$ and $RQ_{rc}$ from a
receiver peer $p_r$, {
            if $QPC_{ic} > RQ_{rc}$, {
                send ACK to $p_r$;
                $SEQ = 0$;}
            else send NACK to $p_r$;
    }
    On receipt of a Start request from $p_r$, {
            $stop$ = false;
            while (~ stop) {
                $SEQ = SEQ + 1$;
                send a packet $pkt_{SEQ}$ to $p_r$;
            }
    }
    On receipt of a Disconnect request from $p_r$, {
        stop = true; /* stop sending packets to $p_r$; */
```

}

The current source peer p_i^c detects that p_i^c is expected not to support the receiver peer p_r with enough QoS $Q_{ic}(\geq RQ_{rc})$. Here, the source peer p_i^c finds another source peer p_k^c which can support the receiver peer p_r with the multimedia content c of enough QoS. Then, the source peer p_i^c sends a *content* request of the multimedia content c to the source peer p_k^c. If the source peer p_k^c agrees on sending the multimedia content c, the current source peer p_i^c sends a *source-change* request to the receiver peer p_r. On receipt of the *source-change* request from the source peer p_i^c, the receiver peer p_r sends a *Start* request with the sequence number *seq* to the source peer p_k^c as discussed here. Thus, not only the receiver peer p_r but also the current source peer p_i^c monitor QoS of the content c. If the service QoS Q_{irc} is expected to be too degraded to satisfy the requested QoS RQ_{rc}, the receiver peer p_r or the source peer p_i^c tries to find another source peer p_k^c. Finally, the receiver peer p_r sends a *Start* request to a source peer p_k^c which takes over the current source peer p_i^c.

By using the protocol, a receiver peer p_r can receive a sequence of packets of the multimedia content c with required QoS even if the receiver peer p_r or source peer is moving. The receiver peer p_r changes a source peer p_i^c with another source peer p_k^c which can send packets of the multimedia content c with required QoS if the source peer p_i^c could not support enough QoS.

6 Evaluation

In the evaluation, there is an area A where there are a moving receiver peer p_r and fixed source peers $p_1^c, \cdots, p_n^c$ $(n \geq 1)$ of a multimedia content c which are interconnected in wireless channels. The source peers $p_1^c, \cdots, p_n^c$ are uniformly distributed in an area A where the receiver peer p_r arbitrarily moves. Here, we make the following assumptions on the evaluation:

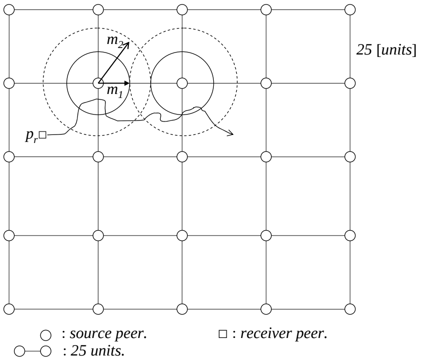


Fig. 10. Mesh.

1. Totally 25 source peers are uniformly deployed in a 100×100 [units] mesh area as shown in Figure 10. The mesh interval is 25 [unit].

2. The receiver peer p_r moves with the velocity v_r [unit/sec] in one direction to a location $\langle x,\ y\rangle$ for one time unit and then randomly changes the direction to a location $\langle x + v_r sin\theta,\ y + v_r cos\theta\rangle$, where the angle θ is randomly calculated. One unit time shows 10[msec.] in the evaluation.

3. The bandwidth between the receiver peer p_r and a source peer p_i^c is considered as the service QoS Q_{irc}. The bandwidth Q_{irc} depends on the distance d_{ir} between the receiver peer p_r and the source peer p_i^c. The maximum bandwidth is assumed to be 1.0 which the receiver peer p_r can obtain if p_r exists on the same location as a source peer p_i^c. The minimum bandwidth is 0.
4. The required bandwidth RBW is 0.56. $TRQ_{ri} = 0.6 + 0.4 \cdot RBW = 0.824$ and $SRQ_{ri} = 0.3 + 0.7 \cdot RBW = 0.692$. $BW(d)$ shows the bandwidth which a receiver peer p_r can receive at distance d from a source peer. $BW(d) = 1$ for $0 \leq d \leq m_1$. m_2 is randomly selected from m_1 to $2m_1$. For $m_1 < d \leq m_2$, $BW(d)$ is $\frac{-0.95}{(m_1 - 2m_1)^2} m_2^2 + \frac{1.9 m_1}{(m_1 - 2m_1)^2} m_2 - \frac{0.95 m_1^2}{(m_1 - 2m_1)^2} + 1$. $BW(d)$ is approaching to 0 from 0.05 for $d > m_2$.

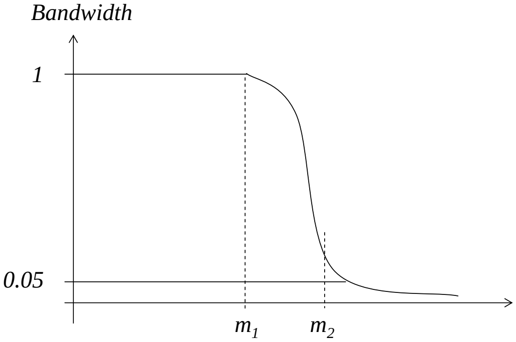


Fig. 11. Graph of bandwidth.

The larger m_1 is, with the stronger radio a source peer sends messages to the receiver peer p_r. The assumptions mean that one unit shows 10[m]. If a receiver peer shows a person who is walking with 4[km/h], the velocity $v_r = 0.011$[units/sec.]. The velocity v_r is 0.055 for a bicycle with velocity 20 [km/h]. $v_r = 0.222$ for a car with velocity 80 [km/h] and 0.277 for 100 [km/h], respectively. In our multi-source streaming (MSS) model, a current source peer p_i^c transits to a suspicious state in a receiver peer p_r if the bandwidth Q_{irc} decreases. Here, the receiver peer p_r finds another source peer which can support enough QoS by using some detection algorithms. If a source peer p_j^c is found, the receiver peer p_r negotiates with the source peer p_j^c. Then, the source peer p_j^c starts sending packets of the multimedia content c to the receiver peer p_r if the current source peer p_i^c would not support enough QoS. On the other hand, in the traditional (T) model, the receiver peer p_r starts finding another source peer if QoS supported by the current source peer is degraded. Since it takes time to find another source peer and negotiate with the source peer, the receiver peer p_r may not receive enough QoS. If a receiver peer p_r does not receive enough QoS, the receiver peer p_r is referred to as *faulty*. Figure 12 shows the average *fault* ratios of the multi-source streaming (MSS) model and the traditional (T) model that the receiver peer p_r is in the faulty state for m_1 for $v_r = 0.011$. As shown in Figure 12, the *MSS* model implies less fault ratio than the *T* model. For example, the fault ratio of the *MSS* model is 40.21% while 72.28% in the *T* model for $m_1 = 9$. The fault ratio of the *MSS* model is only 55% of the *T* model. Figure 13 shows the fault ratio for the velocity $v_r = 0.138$, i.e. a car velocity of 50 [km/h].

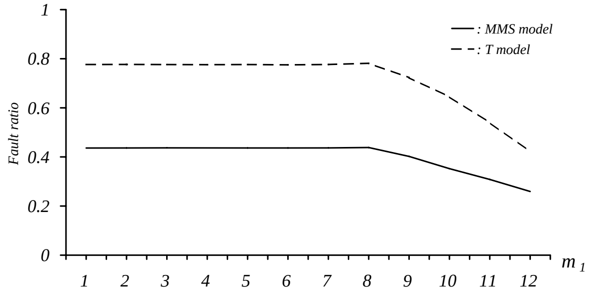


Fig. 12. Fault ratio ($v_r = 0.011$).

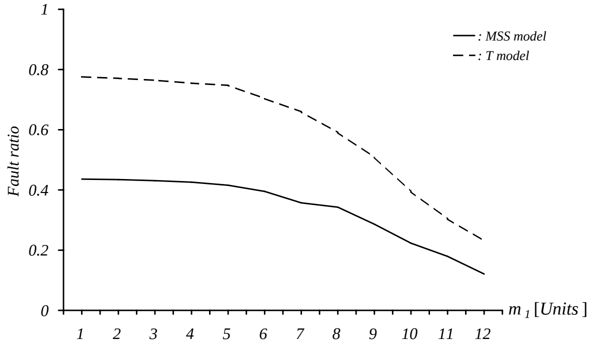


Fig. 13. Fault ratio ($v_r = 0.138$).

Figures 14 and 15 show how the fault ratio changes with the change of the velocity v_r for the MSS and T models. In the T model, the faster the receiver peer p_r is, the more greatly the fault ratio is increased. The MSS model supports more reliable communication, i.e. smaller fault ratio than the T model.

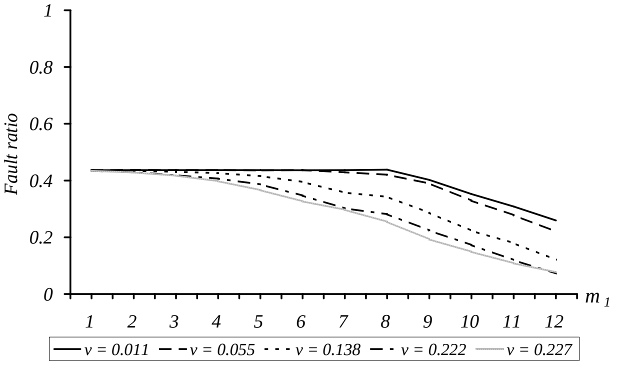


Fig. 14. Fault ratio in the MSS model.

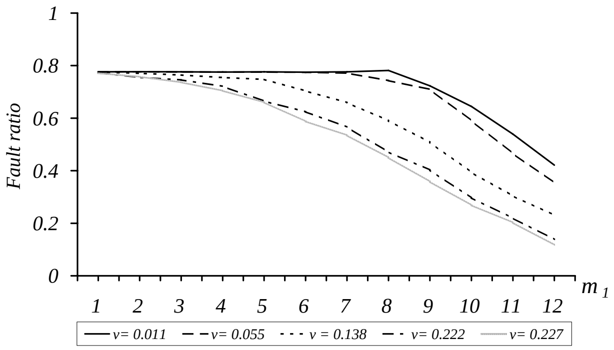


Fig. 15. Fault ratio in the T model.

7 Concluding Remarks

In this paper, we discussed how to support a mobile receiver peer with enough QoS of the multimedia steaming service by cooperation of multiple source peers. We proposed a type of multi-source streaming (MSS) model where a receiver peer can receive packets of a multimedia content c with enough QoS from one of multiple source peers. Not only a receiver peer but also source peer is moving in a network. Here, QoS supported by the source peer is changing according to the movement of the receiver peer and source peer. In this paper, the receiver peer switches a source peer with another source peer if the source peer might not support enough QoS. In the MSS model, the receiver peer detects another source peer which can support the multimedia content c of enough QoS with help of the acquaintance peers. In Furthermore, the receiver peer negotiates with the detected source peer before the current source peer will not be able to support the receiver peer with the multimedia content c of enough QoS.

We evaluated the MSS model compared with the traditional (T) model in terms of the fault ratio, i.e. how often a receiver peer receives packets with lower QoS than required. In the evaluation, we showed the MSS model implies about 55% smaller fault ratio than the T model.

References

1. M. Bober and J. Kittler. Video coding for mobile communications-mpeg4 perspective. In *Proc. IEE Colloquium on the Future of Mobile Multimedia Communications (Digest No. 1996/248)*, pages 3/1–3/9, 1996.
2. S. Itaya, N. Hayashibara, T. Enokido, M. Takizawa. Distributed coordination protocols to realize scalable multimedia streaming in peer-to-peer overlay networks. In: *Proc. of ICPP2006 (2006)*, pages 569–576, 2006.
3. J.-I. Jung. Quality of service in telecommunications. i. proposition of a qos framework and its application to b-isdn. *Communications Magazine, IEEE*, 34(8):108–111, Aug 1996.
4. J.-I. Jung. Quality of service in telecommunications .ii. translation of qos parameters into atm performance parameters in b-isdn. *Communications Magazine, IEEE*, 34(8):112–117, Aug 1996.
5. L. Michael, M. Mihaljevic, S. Haruyama, and R. Kohno. A framework for secure download for software-defined radio. *Communications Magazine, IEEE*, 40(7):88–96, Jul 2002.

6. H. Montes, G. Gomez, and D. Fernandez. An end-to-end qos framework for multimedia streaming services in 3g networks. In *Proc. 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 4, pages 1904–1908, 2002.
7. S. Mungee, N. Surendran, and D. Schmidt. The design and performance of a corba audio/video streaming service. In *Proc. 32nd Annual Hawaii International Conference on HICSS-32 System Sciences*, page 14pp., Volume Track8 1999.
8. C. Perkins. Mobile ip. *Communications Magazine, IEEE*, 35:84–99, 1997.
9. J. Postel. Internet Control Message Protocol. RFC 792 (Standard), 1981. Updated by RFCs 950, 4884.
10. R. Ramanujan, J. Newhouse, M. Kaddoura, A. Ahamad, E. Chartier, and K. Thurber. Adaptive streaming of mpeg video over ip networks. In *Proc. nd Annual Conference on Local Computer Networks*, pages 398–409, 2–5 Nov. 1997.
11. A. Rowstron and P. Druschel. Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.
12. H. Shimoura and K. Tenmoku. Development of elemental algorithms for future dynamic route guidance system. In *Proc. Vehicle Navigation and Information Systems Conference*, pages 321–326, 31 Aug.–2 Sept. 1994.
13. S. Shiraishi, T. Enokido, and M. Takizawa. Fault-tolerant mobile agent in distributed objects systems. In *Proc. of the 9th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2003)*, pages 145–151, 2003.
14. S. Spangler and J. Kreulen. Knowledge base maintenance using knowledge gap analysis. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–466, New York, NY, USA, 2001. ACM.
15. I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications. *IEEE/ACM TON*, 11(1):17–32, 2003.
16. Y. Tanaka, N. Hayashibara, T. Enokido, and M. Takizawa. Fault-Tolerant Transactional Agent Model for Distributed Objects. *In Proc. of 9th IEEE International Symposium on Object and component-oriented Real-time Distributed Computing (ISORC 2006)*, pages 279–286, 2006.
17. K. Watanabe, T. Enokido, M. Takizawa, and K. Kim. Charge-based Flooding Algorithm for Detecting Multimedia Objects in Peer-to-Peer Overlay Networks. *Proc. of IEEE 19th Conference on Advanced Information Networking and Applications(AINA-2005)*, 1:165–170, 2005.
18. K. Watanabe, T. Enokido, M. Takizawa, and K. Kim. Acquaintance-based Protocol for Detecting Multimedia Objects in Peer-to-Peer Overlay Networks. *Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05)*,pages 245–252, 2005.
19. Watanabe, K. and Nakajima, Y., Enokido, T., Takizawa, M. Ranking Factors in Peer-to-Peer Overlay Networks. *ACM TASS*, 2(3):11:1–11:26, 2007.
20. P. Wright. Knowledge discovery in databases: tools and techniques. *Crossroads*, 5(2):23–26, 1998.
21. B. Y. Zhao, J. Kubiatowicz, and A. D. Joseph. Tapestry: An Infrastructure for Fault-resilient Wide-area Location and Routing. Technical Report UCB/CSD-01-1141, University of California, Berkeley, 2001.
22. F. Zijderhand and J. Biesterbos. Functions and applications of socrates: a dynamic in-car navigation system with cellular-radio based bi-directional communication facility. In *Proc. Vehicle Navigation and Information Systems Conference*, pages 543–546, 1994.
23. Napster. http://www.napster.com.
24. Wikipedia. http://en.wikipedia.org/wiki/Peer-to-peer/.