國　立　中　央　大　學

資 訊 工 程 研 究 所

碩　士　論　文

# Peer-to-Peer AOI Voice Chatting for Massively Multiplayer Online Games

# 巨量多人線上遊戲之同儕網路互動範圍語音交談

研 究 生：陳泓翔

指導教授：江振瑞 博士

中 華 民 國 九 十 六 年 七 月

# Peer-to-Peer AOI Voice Chatting for Massively Multiplayer Online Games

Student：Huang-Shiang Chen

Advisor：Dr. Jehn-Ruey Jiang

**A Thesis**

**Submitted to Department of Computer Science and Information Engineering**

**College of Electrical Engineering and Computer Science**

**National Central University**

**in Partial Fulfillment of the Requirements**

**for the Degree of**

**Master**

**in**

**Computer Science and Information Engineering**

**July 2007**

**Jhongli City, Taiwan, Republic of China**

# 國立中央大學圖書館
# 碩博士論文電子檔授權書

(95 年 7 月最新修正版)

本授權書所授權之論文全文電子檔(不包含紙本、詳備註 1 說明)，爲本人於國立中央大學，撰寫之碩/博士學位論文。(以下請擇一勾選)

(Ｖ)同意 (立即開放)

(　　)同意 (一年後開放)，原因是：＿＿＿＿＿＿＿＿＿＿

(　　)同意 (二年後開放)，原因是：＿＿＿＿＿＿＿＿＿＿

(　　)不同意，原因是：＿＿＿＿＿＿＿＿＿＿

以非專屬、無償授權國立中央大學圖書館與國家圖書館，基於推動「資源共享、互惠合作」之理念，於回饋社會與學術研究之目的，得不限地域、時間與次數，以紙本、微縮、光碟及其它各種方法將上列論文收錄、重製、公開陳列、與發行，或再授權他人以各種方法重製與利用，並得將數位化之上列論文與論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

研究生簽名: 陳泓翔　　學號：945202056

論文名稱: Peer-to-Peer AOI Voice Chatting for Massively Multiplayer Online Games

指導教授姓名：江振瑞

系所　：資訊工程所　□博士班　■碩士班

日期：民國 96 年 7 月 11 日

---

備註：

1. 本授權書之授權範圍僅限電子檔，紙本論文部分依著作權法第 15 條第 3 款之規定，採推定原則即預設同意圖書館得公開上架閱覽，如您有申請專利或投稿等考量，不同意紙本上架陳列，須另行加填聲明書，詳細說明與紙本聲明書請至 http://blog.lib.ncu.edu.tw/plog/ 碩博士論文專區查閱下載。
2. 本授權書請填寫並**親筆**簽名後，裝訂於各紙本論文封面後之次頁（全文電子檔內之授權書簽名，可用電腦打字代替）。
3. 請加印一份單張之授權書，填寫並親筆簽名後，於辦理離校時交圖書館（以統一代轉寄給國家圖書館）。
4. 讀者基於個人非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

論文名稱：巨量多人線上遊戲之同儕網路互動範圍語音交談

校所組別：國立中央大學 資訊工程研究所　　頁數：28

研究生：陳泓翔　　指導教授：江振瑞　博士

## 中文摘要：

近年來，隨著網路的發達，線上遊戲也越來越熱門。而為了增進線上遊戲的遊戲經驗也有許多的研究進行著，其中最明顯的便是遊戲畫面的增進。然而目前在線上遊戲中的交談方式仍然是採用輸入文字來互動交談，而不是對人類來說較自然的語音交談。由於遊戲的操控和文字的輸入都得使用鍵盤，因此玩家無法一邊操控遊戲中的人物一邊和其他玩家交談，這在許多的線上遊戲(第一人稱射擊、角色扮演)裡，是很煩人的一件事。因此，我們於這篇論文中提出適用於多人線上遊戲的互動範圍語音交談。

在線上遊戲裡同時可能有很多遊戲內容和使用者，但是一個玩家所能互動的範圍是有限的，而這個以玩家為中心圓形的範圍，我們便定義為互動範圍(Area of Interest)。那所謂的互動範圍語音交談便是讓玩家能自然的跟互動範圍內的其他玩家們用語音來交談，玩家能聽到互動範圍內其他玩家的聲音，反之亦然。這不但使玩家能更容易的和其他玩家交談，互動範圍內的交談更使線上遊戲變得更生動。在本論文裡，我們提出了 QuadCast、SectorCast 來達到互動範圍內的語音交談，我們也對這兩個方法做了許多的分析和模擬來比較他們的效能。

**關鍵字：**同儕網路、語音交談、巨量多人線上遊戲、Voronoi、互動範圍。

# Peer-to-Peer AOI Voice Chatting for Massively Multiplayer Online Games

**Student：Huang-Shiang Chen　Advisor：Prof. Jehn-Ruey Jiang**

***Department of Computer Science and Information Engineering,***

***National Central University***

***Jhongli City, Taoyuan, 320, Taiwan***

***Abstract*** ─In recent years, massively multiplayer online games (MMOGs) have become more and more popular. Many techniques are proposed to enhance the experience of MMOGs, such as realistic 3D graphics, vivid animations, and player communication tools, etc. However, in most MMOGs, communication between players is still based on text, which is unnatural and inconvenient. In this these, we propose the concept of AOI voice chatting for MMOGs, which is voice chatting of dynamic membership based on the *Area of Interest* (AOI) of players in the MMOG. The term AOI is defined to be the area around a player that s/he can sense and interact. By AOI voice chatting, an MMOG player can easily chat by voice with other players within her/his AOI. This improves the way players communicate with one another and provides a more realistic virtual environment. We propose two peer-to-peer schemes, namely QuadCast and SectorCast, to achieve efficient AOI voice chatting for MMOGs. We also perform simulation experiments for the two solutions to show their performances.

***Keywords*** ─Peer-to-Peer, Voice Chatting, MMOG, Voronoi Diagram, Area of Interest

# Contents

**1 Introduction** **3**

**2 Related work** **6**

2.1 Architecture of MMOG . . . . . 6

2.2 Immersive audio system . . . . . 6

2.3 Human conversational speech model . . . . . 8

**3 The System Model and Problems** **9**

3.1 System model . . . . . 9

3.2 The problems of a base solution . . . . . 10

**4 Proposed solutions** **13**

4.1 Quadrant-based forwarding (QuadCast) . . . . . 14

4.2 Sector-based forwarding (SectorCast) . . . . . 16

4.3 Packet aggregation . . . . . 17

4.4 Alternatives of the recipient list . . . . . 19

**5 Evaluation** **21**

5.1 Total upload bandwidth required to support AOI voice chatting . . . . 22

5.2 Simulation under bandwidth limitation . . . . . 23

5.3 Average end-to-end latency . . . . . 24

**6 Conclusion** **26**

# List of Figures


1 The model of human conversational speech . . . . . 8

2 System model . . . . . 9

3 AOI broadcast . . . . . 10

4 Burst network bandwidth usage in AOI broadcast . . . . . 11

5 Determination of the effects of end-to-end delay by E-model . . . . . 13

6 Format of a forwarding voice packet . . . . . 14

7 Quadrant-based voice package forwarding . . . . . 15

8 Player clustering in MMOG. . . . . 16

9 Sector-based voice packet forwarding. . . . . 17

10 A scenario of packet aggregation. . . . . 18

11 The concept of voice mixing by color mixing . . . . . 19

12 Total upload bandwidth required to support AOI voice chatting. . . . 22

13 Total byte sent under bandwidth limitation. . . . . 24

14 Drooping rate under bandwidth limitation. . . . . 25

15 Average end-to-end latency. . . . . 25

# 1 Introduction

In recent years, massively multiplayer online games (MMOGs) have become more and more popular. For example, World of Warcraft [5], one of the most popular MMOGs, reached a record of 8.5 million subscribed players worldwide. And according to a report by ScreenDigist [4], the MMOG market broke $1 billion mark in the west in 2006. An MMOG is a computer game which can support hundreds or thousands of players playing simultaneously in a virtual world over internet. A player in the MMOG is represented by a personalized 3D character called an avatar. By controlling the avatar, a player can navigate the virtual world, fight monsters for rewards, interacts with other players, and so on.

Many techniques are proposed to enhance the experience of MMOGs, such as realistic 3D graphics, vivid animations, and player communication tools, etc. However, in most MMOGs, communication between players is still based on text, which is unnatural and inconvenient. Players *type* and *read* the text in the chat box instead of *speaking* and *listening*. Furthermore, because the mouse and the keyboard are the major input devices of most MMOGs, one may not control the avatar and communicate with other players at the same time. When a player wants to chat with others by typing text, he loses the control of his avatar for a while. It is inconvenient for players to use, especially for those not good at typing. As a result, players begin to seek for voice chatting solutions, such as Teamspeak[2], Ventrilo[3], and Skype[1], etc.

Teamspeak and Ventrilo are two popular VoIP applications supporting group voice chatting. They are client-server based and thus need dedicated servers, which can be publicly charged or voluntarily supported. When a user of a group talks, s/he transmits voice packets to the server. This server then mixes voice streams of all group users and send back to each of them. The client only delivers player's voice packet and receive the mixed voice packet from the server, but the server needs to receive voice packets from all clients and deliver mixed packets to them during a pre-specified short period of time. Therefore, the number of players supported by a server depends on the server's

network bandwidth and computation power. Skype is a popular peer-to-peer based VoIP application. It not only supports telephony over internet, but also the group voice chatting. Skype needs no dedicate server for mixing voice packets because of its peer-to-peer architecture. However, since Skype only support group voice chatting for at most five users, it is not suitable for MMOGs, which usually have more than five players in a group. Teamspeak, Ventrilo, and Skype are not widely used in MMOGs because they have static group membership (i.e., the membership of a group is fixed or seldom changed). A user has to intentionally join a certain group to talk to someone in the group. For example, when a user sees a crowd of players and wants to figure out what they are talking about, s/he has to join their group intentionally to talk to them; s/he can not just move toward them to overhear or join the conversation. This reduces the interactivity of players in an MMOG.

In this thesis, we propose the concept of AOI voice chatting for MMOGs, which is dynamic-membership voice chatting based on the *Area of Interest* (AOI) of players in the MMOG. The term AOI is defined to be the area around a player that s/he can sense and interact [11]. By AOI voice chatting, an MMOG player can easily chat by voice with other players within her/his AOI. This improves the way players communicate with one another and provides a more realistic virtual environment. We also propose two peer-to-peer schemes, namely QuadCast and SectorCast, to achieve efficient AOI voice chatting for MMOGs. The two schemes adopt the peer-to-peer architecture to eliminate the requirement of servers and to utilize the bandwidth of all participating players.

The rest of this thesis is organized as follows: Chapter 2 introduces some background knowledge of peer-to-peer networks and voice chatting. Also, we describe a human conversational speech model in this chapter. In Chapter 3, we first give a scenario about how AOI voice chatting works in the MMOG. Then, we describe how we model the system as well as the design goal and major challenges. In Chapter 4, we propose the two solutions to support AOI-voice chatting for MMOGs. We perform simulation experiments for the two solutions: The simulation results and the comparisons are

given in Chapter 5. Finally, concluding remarks are drawn in Chapter 6.

# 2 Related work

## 2.1 Architecture of MMOG

Most MMOGs nowadays are based on the client-server architecture. In such an architecture, the virtual world of MMOG is maintained on a centralized server or server cluster, where players log in and start playing the game. By a centralized server or server cluster, the consistency of game states can be easily maintained and cheating between players can also be avoided. However, because the server is in charge of all event processing and message transmission, it becomes a performance bottleneck when the number of players are increasing, this constrains the scalability of the MMOG system.

To achieve better scalability, researchers propose peer-to-peer architectures, such as VON[7], Solipsis[9], Apolo[8], etc., for the MMOG. In the peer-to-peer architectures, every peer runs the same program in a distributed manner without a centralized server. As we mentioned before, in the MMOG, a player senses and interacts only with other players in his/her AOI. Therefore, a peer only exchanges and processes messages with limited peers within the AOI. In this way, the peer-to-peer MMOG architecture can potentially provide better scalability than the client-server architecture. However, because there is no centralized server, many problems become more complex to solve. For example, in the client-server architecture, the neighbor discover is not a problem because the server has all position information of players. But in peer-to-peer architecture, players have to discover their new neighbors by themselves through a predefined protocol.

## 2.2 Immersive audio system

In [12], the author proposed *immersive audio communication system* for MMOGs. The system allows an avatar to hear voices of all avatars in his AOI by creating a personalized audio scene for every avatar. This personalized audio scene mixes and attenuates all voice from other avatars according to the propagation distance. The

author also examined advantages and trade-offs of the peer-to-peer, the centralized server and the distributed server architectures.

In the peer-to-peer architecture, a peer connects directly to other peers in its AOI. Due to the direct connection between peers, the system provides low latency and has no single point failure. However, if a peer has many neighbors in its AOI, it cannot afford the bandwidth consumption. In the centralized server architecture, the central server gathers voice streams from all clients and mixes them with the consideration of propagation attenuation. After that, the server sends the mixed audio scene back to all clients. In this architecture, the central server becomes the bottleneck of the system and the single point of failure. In the distributed server architecture, the whole virtual world is partitioned into multiple regions, called locales, and audio streams in one or more locales can be processed by different locale servers located in different country. A peer can transmit the audio streams to one of the locale servers with the shortest latency. The audio scene is produced by the server and transmitted by the nearby locale servers. This paper focused on the distributed server architecture, and proposed a locale server relocation algorithm to minimize the transmitting latency between players.

The paper [10] proposed a peer-to-peer based immersive audio streaming system for MMOGs. This paper proposed to use the Voronoi diagram to find out the direct neighbors for a peer, and also proposed a model for mixing audio streams of direct neighbors under the consideration of peers' positions and directions. Each peer in the system gathers the voice streams of direct neighbors in each time step, mixes them according to the audio mixing model and sends back the mixed stream to its direct neighbors. The audio streams in this system is directional and different direct neighbors will receives different mixed audio streams. The system is scalable because only direct neighbors' voice streams are mixed. However, the system has the echo problem that a peer or player may hear its own voice just sent earlier.

## 2.3 Human conversational speech model

The article [13] described some characteristics and statistics of human conversational speech. The human conversation is modelled as many on-off audio signals classified into four types: single talk, double talk, pause and mutual silence (refer to Figure 1) In a two-member conversation (A and B are speakers), single talk means either A or B is talking; double talk means both A and B are talking; mutual silence means both A and B are silent. And when A stop talking, he is in the pause state.

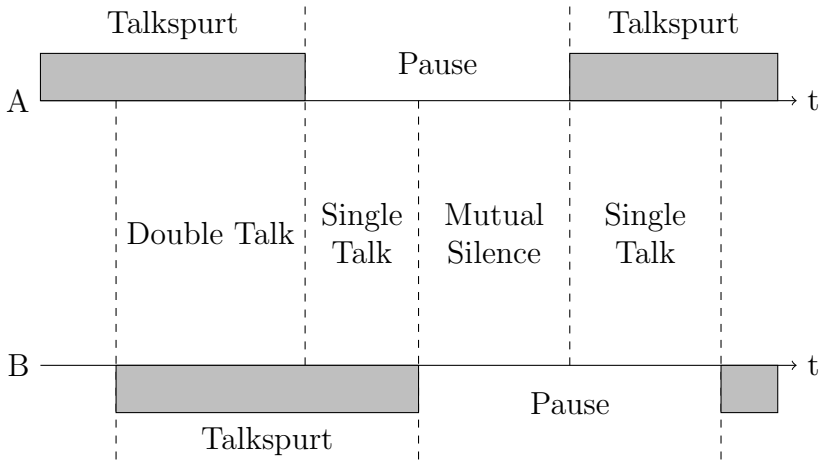



Figure 1: The model of human conversational speech

Table 1: Temporal parameters in conversational speech

| Parameter | Duration (s) | Rate (%) |
|---|---|---|
| Talk-spurt | 1.004 | 38.53 |
| Pause | 1.587 | 61.47 |
| Double talk | 0.228 | 6.59 |
| Mutual silence | 0.508 | 22.48 |

In [13], a statistic of temporal parameters of conversational speech is also shown (refer to Table 1). This table shows that in a conversation a person only spends 40% of time in speaking and keeps silent during the rest of the time. Also, we can see that in a conversation, once a person is talking, the talking rate of other people is reduced to only 6%. In other words, only few people are talking while others are listening in a conversation.

# 3 The System Model and Problems

## 3.1 System model

The virtual world of an MMOG is modelled as a two-dimensional plane, and the players are modelled as nodes moving and talking on it. The terms node, peer, player and avatar are used interchangeably in the following context. Each node has a unique id ($ID$), a coordinate $(X, Y)$, an AOI, and some other behavior parameters. The AOI of each node defines the area that this node can sense or interact, and the nodes in the AOI is defined as *AOI neighbors*. For example, in Figure 2, the big circle around node A is A's AOI, and nodes B,..,I are node A's AOI neighbors. We assume that each node's AOI is a circle centered at the node with a fixed radius and all nodes have the same AOI radius.

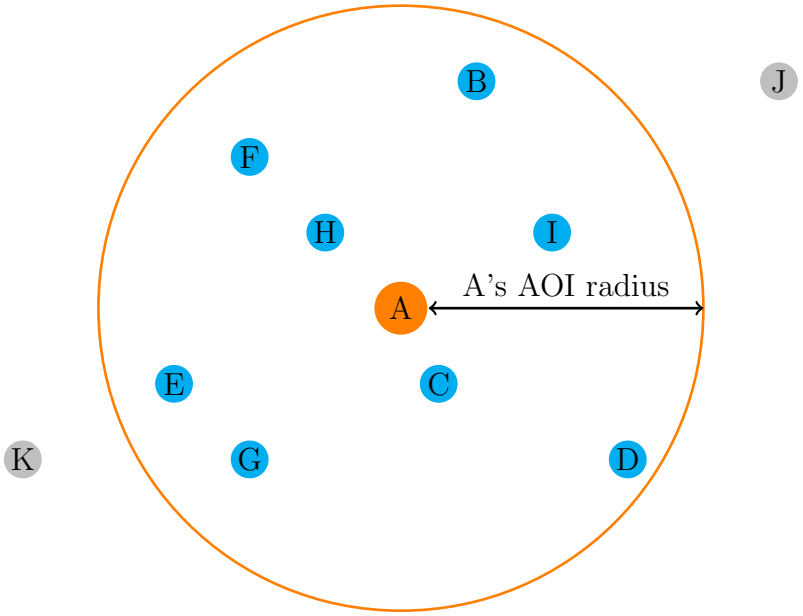



Figure 2: System model

Under the above-mentioned system model, the AOI voice chatting can be regarded as *voice packet multicast* within the AOI. When a node talks, the voice packets are multicasted to his AOI neighbors. In this way, a node's AOI neighbors can hear its voice, and vice versa. As illustrated in Figure 2, when node A talks, all its AOI-neighbors should receive the voice packet, but non-AOI-neighbors should not. In order

to correctly multicast the voice packets to AOI neighbors, we need a recipient list containing these AOI neighbors. In this thesis, we assume the MMOG system can takes a node's id as the parameter, and returns a recipient list containing this node's AOI neighbors. Each entry of the list contains an AOI neighbor's id, network address, coordinate, and some other properties. The assumption is practical. For example, the VON system can support such a recipient list.

## 3.2 The problems of a base solution

We begin with a base solution - NimbusCast. When a node talks, it first requests a recipient list from the MMOG system. After that, the node delivers the voice packets one by one to all nodes in the recipient list. For example, in figure 3, when node A talks, it first requests a recipient list from the MMOG system, and then delivers the voice packets to all nodes in the list. Node J and K are not in the recipient list because they are outside of A's AOI, and thus they do not receive any voice packets from A.

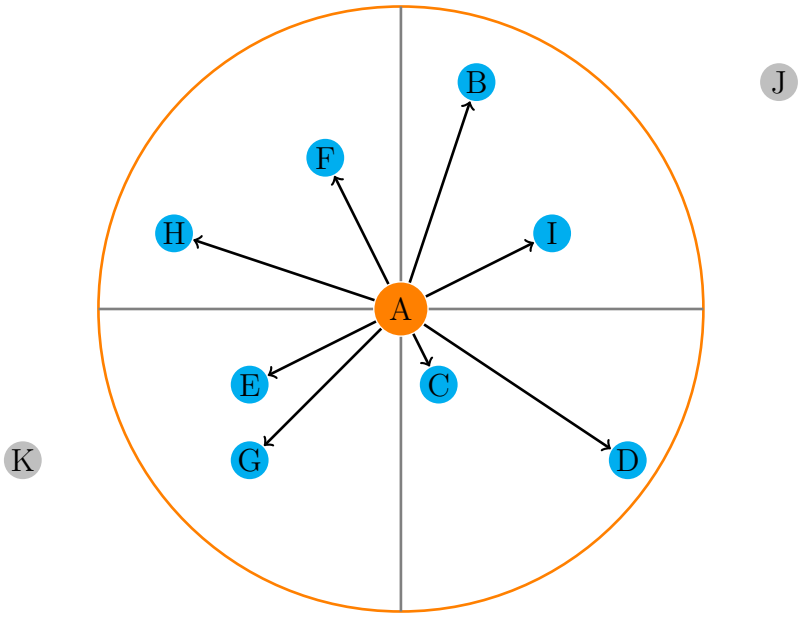



Figure 3: AOI broadcast

In this solution, once the MMOG system returns the correct recipient list, the accuracy transmission requirement can be easily confirmed because of the simplicity of the delivery algorithm. However, the bandwidth consumption of a voice source node

is proportional to the number of players in its AOI. If the required bandwidth exceeds a node's upload bandwidth limitation, the voice packets might be delayed or even dropped. Also, we observed a problem in this model:

- *Idle upload bandwidth*

  When a node talks, it must deliver voice packets to all AOI neighbors. This introduces a phenomenon of burst upload bandwidth usage. As shown in Figure 4, when a NODE talks, the upload bandwidth usage increases dramatically while the network traffic is low when it is silent. When the upload bandwidth usage exceeds a node's bandwidth limitation illustrated as the red line in the figure, those voice packets might be delayed or dropped, which reduces the quality of the voice chatting. For example, if it costs $16kbps$ to support a one-to-one voice chat, and the upload bandwidth limitation is $256kbps$. Once the number of AOI neighbors is more than 16, the bandwidth usage will exceeds the limitation.

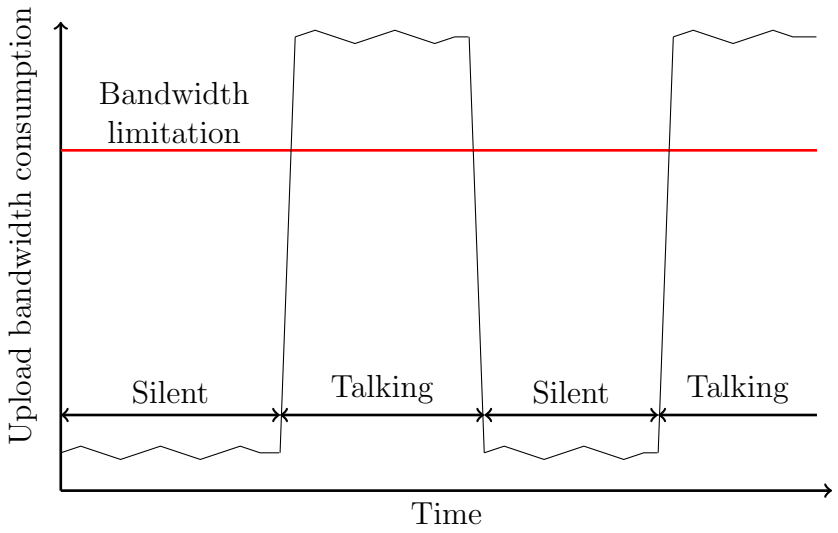



Figure 4: Burst network bandwidth usage in AOI broadcast

Furthermore, as we mentioned in Chapter 2, in a conversation, a node only spends 40% of the time talking, and is silent in the rest 60% of the time. If the idle upload bandwidth of those silent nodes can be used to help forwarding other nodes' voice packets, the instant upload bandwidth usage will decreases, which prevents the

upload bandwidth overloading. As a result, we adopted the packet forwarding in our solutions.

# 4 Proposed solutions

In this chapter, we proposed two AOI multicast algorithms to support the AOI voice chatting. The major problem of the design is the limited network bandwidth. When a node talks, it must deliver the voice packets to all its AOI neighbors. When the bandwidth is not enough for voice packet transmission, some voice packets might be delayed or even dropped, which greatly reduces the quality of voice chatting. Under the constraint of limited bandwidth, we have the design goal for the proposed algorithms.

- ***Accurate transmission***

  The proposed AOI multicast algorithms must guarantee the correctness of voice package delivery. First, only AOI neighbors receive the voice packets. To transmit voice packets to nodes outside the AOI is not necessary and causes bandwidth waste. Second, the proposed algorithms have to prevent a node from receiving the same voice packets more than once, which can reduce bandwidth waste.

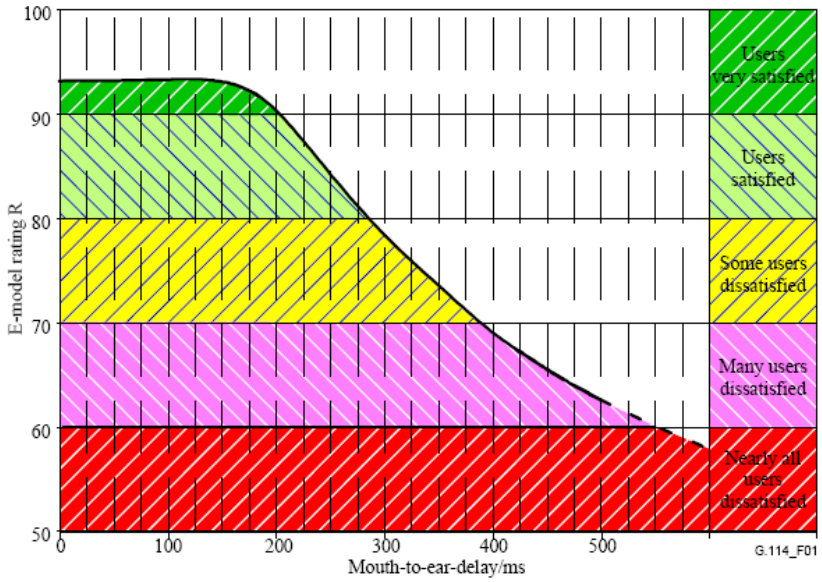



Figure 5: Determination of the effects of end-to-end delay by E-model

- ***Reasonable delay***

  ITU-T recommendation G.114[15] provides a guideline about the one way end-to-end (or mouth-to-ear) delay. In Figure 5, it is shown that most users satisfied

with the end-to-end delay between 150ms to 250ms while a delay above 400ms is not acceptable for most users. According to the recommendation, the end-to-end delay of the proposed solutions should be less than 400ms.

## 4.1 Quadrant-based forwarding (QuadCast)

In this section, we propose a AOI multicast solution by packets forwarding. The main concept of this solution is the *network traffic delegating*. In stead of directly transmit all voice packets, a player only transmits voice packets to *forwarding assistants* (FAs). These forwarding assistants then forward the voice packets to the remaining recipients by the information attached in the voice packets. In this way, because all players contribute their network resource to help forwarding the voice packets, the burst network traffic of the speaking players decrease, and the overall quality of conversation is improved.

As for the design goal of transmission correctness, we must guarantee correctness of voice packet transmission. In this thesis, we use a *forwarding list* attached in the forwarding voice packet to indicate the recipients of this voice contents. The format of the forwarding voice packet is shown in Figure 6. The forwarding voice packet contains a packet header, the voice contents, and a recipient list. This recipient list contains the recipients of the voice contents, and the FA is in charge of forwarding the voice contents to the remaining recipients.

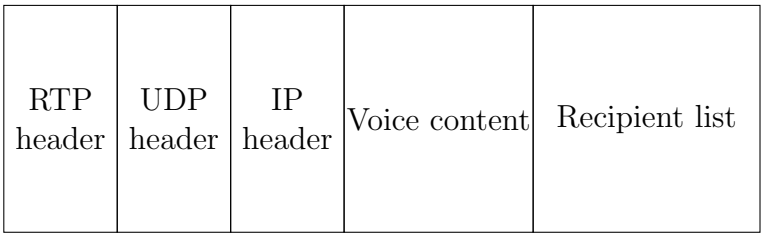



Figure 6: Format of a forwarding voice packet

When a player talks, s/he first acquires a recipient list from the MMOG system, then divides the players in recipient list by their coordinate into four quadrants, and

creates four sublists containing the players in each quadrant. In the next step, in each quadrant, a nearest player[1] is chosen as the FA. After the FA selection, the player creates four forwarding packets with the corresponding recipient lists and deliver them to four FAs. In this thesis, we choose the nearest player as the candidate FA to maximize the probability of packet aggregation, the detail of which will be discussed in Section 4.3.

When an FA receives a forwarding voice packet, the same algorithm is applied to forward the voice contents with different parameters of voice contents source and the recipient list. The voice packets are recursively forwarded until the forwarding list is empty. For example, in Figure 7, when player A talks, he first acquires a recipient list from the MMOG system, and classified the list to four sublists according to the coordinates of the players in the list. Then he delivers four forwarding packets to player D, E, C, and I as his FAs in first, second, third, and fourth quadrant. After receiving the forwarding packets, these FAs apply the same algorithm to forward voice packets to remaining players in the forwarding list.

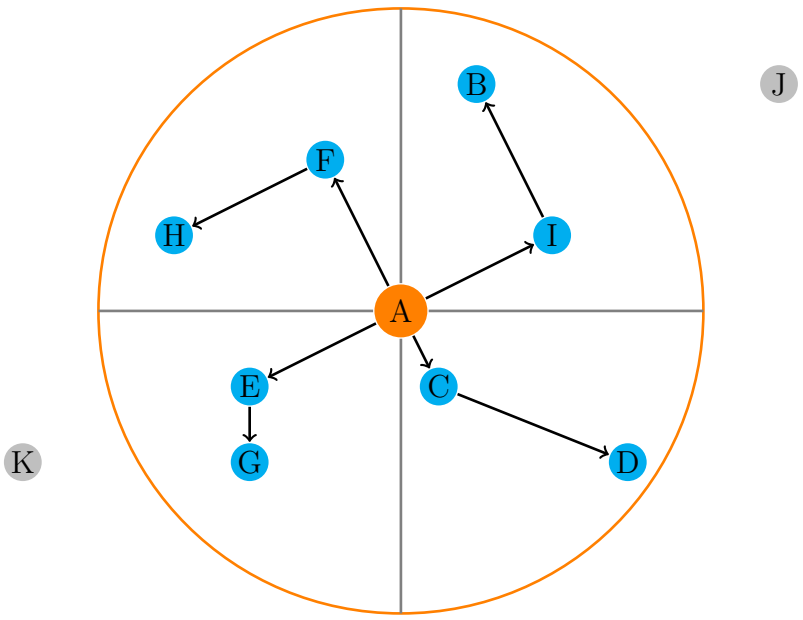
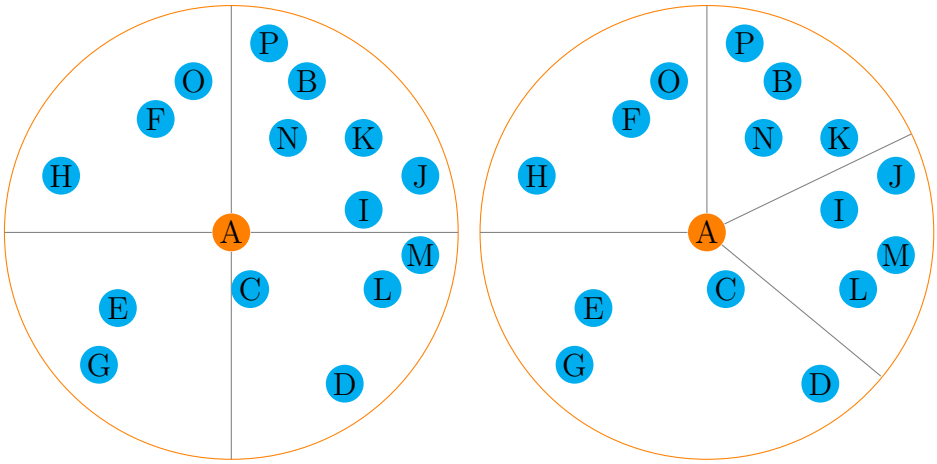



Figure 7: Quadrant-based voice package forwarding

[1]not always exist if there is no player in that quadrant

## 4.2 Sector-based forwarding (SectorCast)

In MMOG, players are usually grouped in some hot-spots like cities, or town squares. By quadrant-based forwarding model, when a player moves toward a player cluster, the number of players in a certain quadrant may be much greater than those in others. For example, in Figure 8(a), there are much more players in the first quadrant of player A's AOI. The message forwarding in this quadrant thus has more hops which causes longer processing time and transmission latency. If we can distribute equal number of players in each sectors in the AOI as shown in Figure 8(b), the latency between players will be close and the quality of service will be better.




(a) Unbalanced player grouping. (b) Balanced player grouping.

Figure 8: Player clustering in MMOG.

In this section, we proposed a sector-based voice forwarding scheme called SectorCast. SectorCast and QuadCast are similar. QuadCast divides the AOI into four quadrants with equal size, while SectorCast divides the AOI into four sectors containing equal number of players. The recipients division are different for the two schemes, and the rest are the same. For example, in figure 9, the AOI is divided to four sectors containing equal number of players, and the same procedure as QuadCast is applied to forward the voice packets. Because we divide the recipients equally to four sectors, the average transmission hops between them are balanced as well as the overall latency. However, the computation complexity of SectorCast is higher than QuadCast.

QuadCast only need to classify the players in the recipient list into four quadrants, but SectorCast need to sort them according to the polar angles of the lines from them to the source node to divide equally into four sectors.

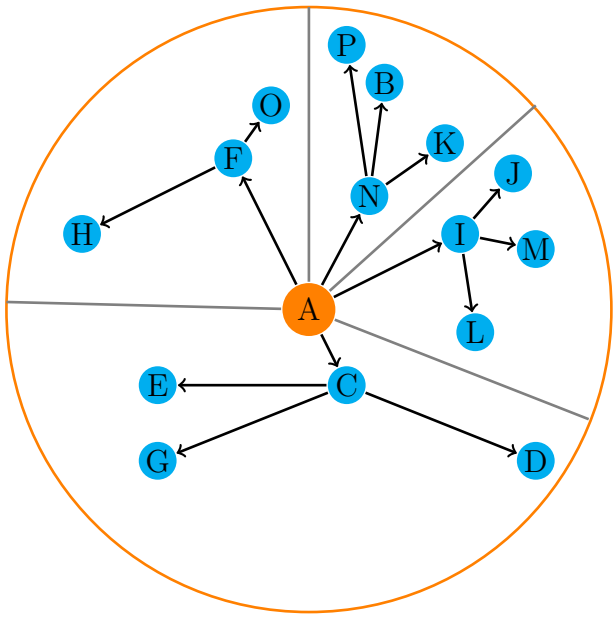
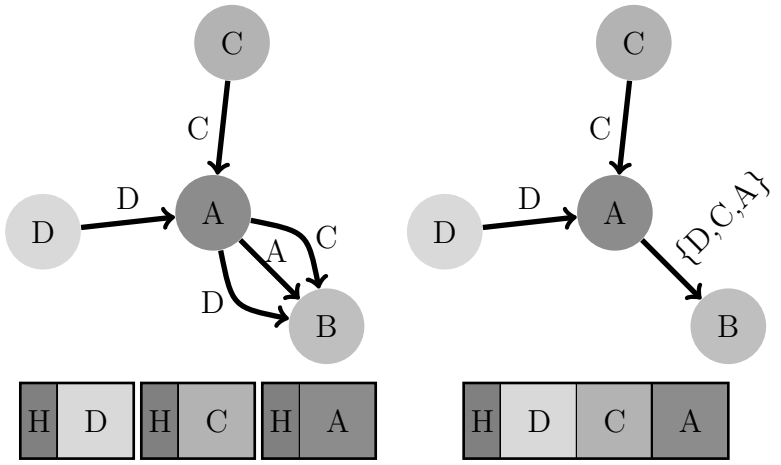



Figure 9: Sector-based voice packet forwarding.

## 4.3 Packet aggregation

In QuadCast and SectorCast schemes, an FA might forward different packets to a same recipient. These packets are sent separately. However, if we can apply aggregation techniques, such as header sharing or audio mixing, to these packets, the packet traffic can be decreased dramatically. For example, assume peer A is the FA of peer C and peer D; it is in charge of forwarding their voice packets to one of the recipients, peer B. At the same time, peer A also delivers its voice packet to peer B. As we illustrate in figure 10(a), without aggregation, three voice packets containing common headers and different voice contents are delivered to peer B. However, in figure 10(b), three packets are merged to be one packet by sharing the same header. As a result, network traffic of two packet headers is saved.

Since a FA peer forwards voice packets to many recipients on behalf of many source




(a) Forwarding without aggregation (b) Forwarding with aggregation

Figure 10: A scenario of packet aggregation.

peers, it must have a way to aggregate voice packets properly. Below, we model the aggregation of packets as the 2-power number addition. A voice packet is assigned an id of a 2-power number. And the id of the aggregated voice packet is defined to be the addition of the ids of packets from different sources. In this manner, a voice id corresponds to a unique combination of packets from specific sources. For example, in Figure 11, there are three peers (peer A, C, and D) talking simultaneously in this local area. The voice contents of these three peers are represented by the voice id 1, 2, and 4, respectively. Peer B will receive voice packets from peer A, C, and D, and thus the voice id of the aggregated voice packet is 7, addition of 1, 2, and 4. While peer D receives the voice packets from peer A and C, so he receives an aggregated voice packet with voice id 6, addition of 2 and 4. We calculate the id of the aggregated voice packet for each recipient. All recipients are then grouped according to the calculated ids. Finally, each group of aggregated voice packet is recursively delivered by QuadCast or SectorCast to corresponding recipients.

Also, the probability of aggregation or the aggregation rate depends on the number of common recipients. In the color mixing example, the closer the centers of different colors, the larger overlay area among them. And the larger overlay area implies more common recipients. This is why we choose the nearest peer as the FA in Section 4.1.

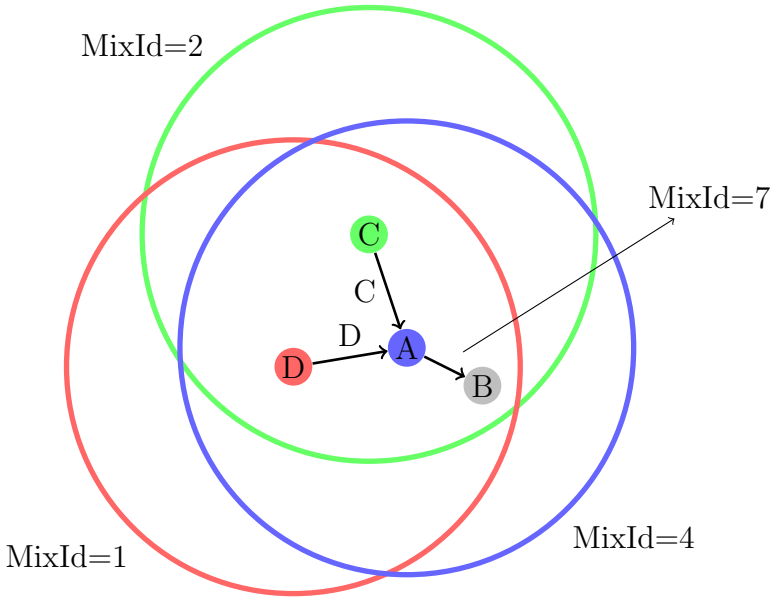



Figure 11: The concept of voice mixing by color mixing

## 4.4 Alternatives of the recipient list

In QuadCast and SectorCast, we adopt the concept of recipient list to guarantee the correctness of voice packet transmission. However, delivering the recipient list consumes large bandwidth. We would like to trade computation resource with network bandwidth. When a voice packet is forwarded, instead of appending a whole recipient list, we only append the position of the current FA for the next FA to calculate the recipients. Once there is no recipients, the algorithm stops.

In QuadCast, when a node talks, it appends its position to the voice packet and deliver this voice packet to the FAs. Because all nodes in the system has the same AOI radius, the FA can figure out the forwarding area of this voice packet by itself. Once the forwarding area is recognized, the FA can select the recipients from its neighbor list. In the same way, the FA also has to append its position to the forwarding voice packet in order to allow the next FA to recognize the forwarding area. In SectorCast, besides the position, the source and the FA need to further append the begin and end angle of the sector to the voice packets for next-hop FA to calculate the recipients. By the position and the two angles of the source and previous FAs, the current FA can easily recognize the correct forwarding area and thus choose the recipients from its

neighbors.

# 5 Evaluation

In this chapter, we compare our proposed AOI voice chatting schemes. In our simulation, we place from 200 to 1000 nodes in a 1000×1000 area to simulate different node density scenarios. The radius of AOI of all nodes is set to 100. The simulation processes for 1000 discrete time-steps, and the interval of each step is 100ms. In the initial step, the simulated nodes are created and each node is assigned a random initial position. A *talkative-threshold* value is also assigned to indicate a node's probability of speaking. In each step, every node moves toward a random direction by a distance of 10, and rolls a random number between 1 to 10. If a node rolls a number greater than his talkative-threshold, this node talks in this step. For example, if a node is assigned a talkative-threshold value of 4, and the rolled number is 5, he talks in this step. As we mentioned before in Chapter 2, in a conversation, a person only spends 40% of the time speaking, so the talkative-threshold value of each node is assigned as 6[2].

At the same time, when a node talks, a voice packet that contains a packet header as well as the voice content is created and put in a buffer. As for the packet size, we use G.729 [14] as our voice codec, and use Real-time Tranport Protocol (RTP) to transmit the voice packets. Under this assumption, the header size of a voice packet is 40 bytes $(20(IP) + 8(UDP) + 12(RTP))$, and the voice content is 100 bytes. The created voice packet is first put into a buffer, and wait for transmission. At the end of each step, all voice packets in the buffer will be processed by different algorithms according to different AOI voice chatting solutions, and be forwarded to their recipients. After that, the buffer is flushed and the next step begins. We then show the simulation results and analysis them in the following sections.

[2]A node talks when the rolled number is 7, 8, 9, and 10, which is the propagability of 40%.

## 5.1 Total upload bandwidth required to support AOI voice chatting

Figure 12 shows the total upload bandwidth consumption of three AOI voice chatting schemes: NimbusCast, QuadCast, and SectorCast. In Figure 12 and the following diagrams, the "-HA" stands for the header aggregation and the "-M" stands for the mixing. Without aggregation, the bandwidth consumption of three schemes are the same. The packet forwarding in QuadCast and SectorCast only mitigate the burst bandwidth consumption, and does not reduce the overall bandwidth consumption. This figure also has another meaning, which is the feasibility of three AOI voice chatting schemes. When the number of nodes is 1000, the average upload bandwidth of each node is about 18bytes/sec. This number is higher than recent wide area network solutions, but we believe that this can be provided pervasively in the near future.

By applying packet aggregation, the bandwidth consumption of QuadCast and SectorCast is reduced, and the reduction is proportional to the number of nodes. When the number of nodes in the system increases, the probability of aggregation also increases, which implies better aggregation performance.

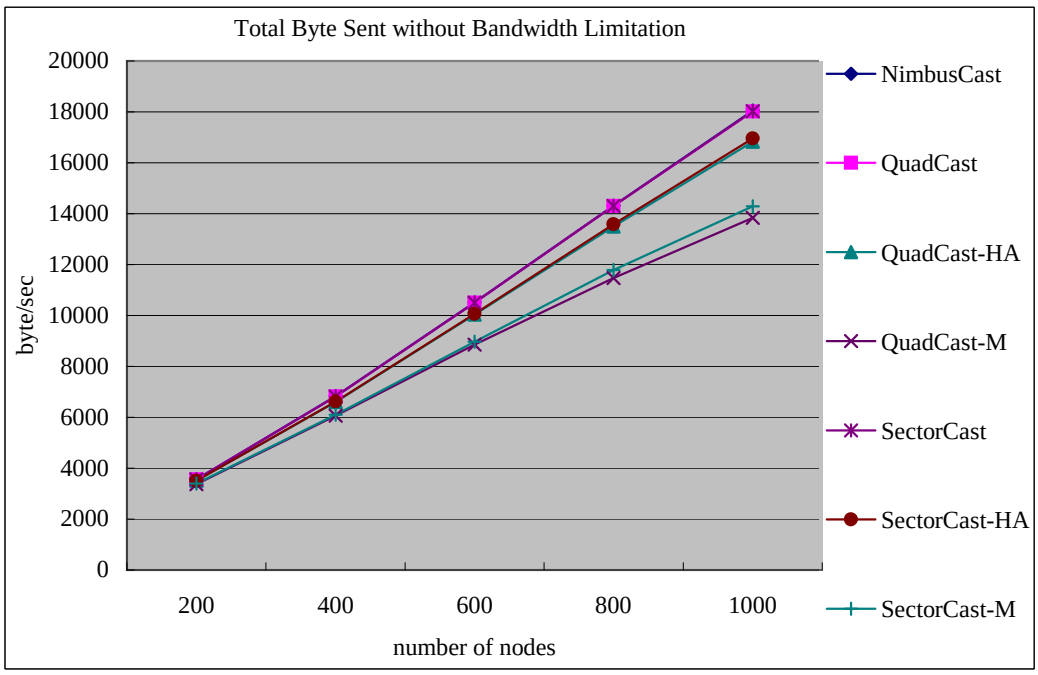



Figure 12: Total upload bandwidth required to support AOI voice chatting.

## 5.2 Simulation under bandwidth limitation

In the previous simulation, the system has a unlimited upload bandwidth, which means a node can deliver all voice packets in a single step. However, this is not realistic in wide area network (WAN) nowadays. We then simulated our proposed solutions on the network with limited bandwidth, and compare the bandwidth efficiency among all schemes. Due to the bandwidth limitation, when delivering a voice packet, a node first checks if the upload bandwidth consumption in this step will exceed the limitation after delivering this packet. If there is still enough bandwidth for packet transmission, the packet is delivered normally. However, if the upload bandwidth consumption will exceed the limitation, the packet will be dropped without delivering.

The total byte sent under bandwidth limitation is shown in Figure 13. We can see that forwarding based solutions, such as QuadCast and SectorCast, can deliver more voice packets under bandwidth limitation, while the amount of packets delivered in NimbusCast begins to reduce when number of nodes of the system reaches 800. As we discussed in Section 3.2, NimbusCast has the burst bandwidth usage problem, and the forwarding solves the problem by using nearby nodes' network resource to mitigate burst bandwidth usage of the speaking node. In other words, under the same network, forwarding based solutions (QuadCast and SectorCast) can deliver more voice packets than NimbusCast because of the mutual sharing of network resource.

Figure 14 shows the packet dropping rate of all schemes. We can observe that in this figure, the dropping rate of NimbusCast is above 20% when number of nodes in the system reaches 1000, while the dropping rate of other solutions are all below 5%. In [6], the author summarized that the dropping rate of voice packets lower than 5% in the internet voice chatting is acceptable, and our forwarding based solutions (QuadCast and SectorCast) do fulfill this requirement.

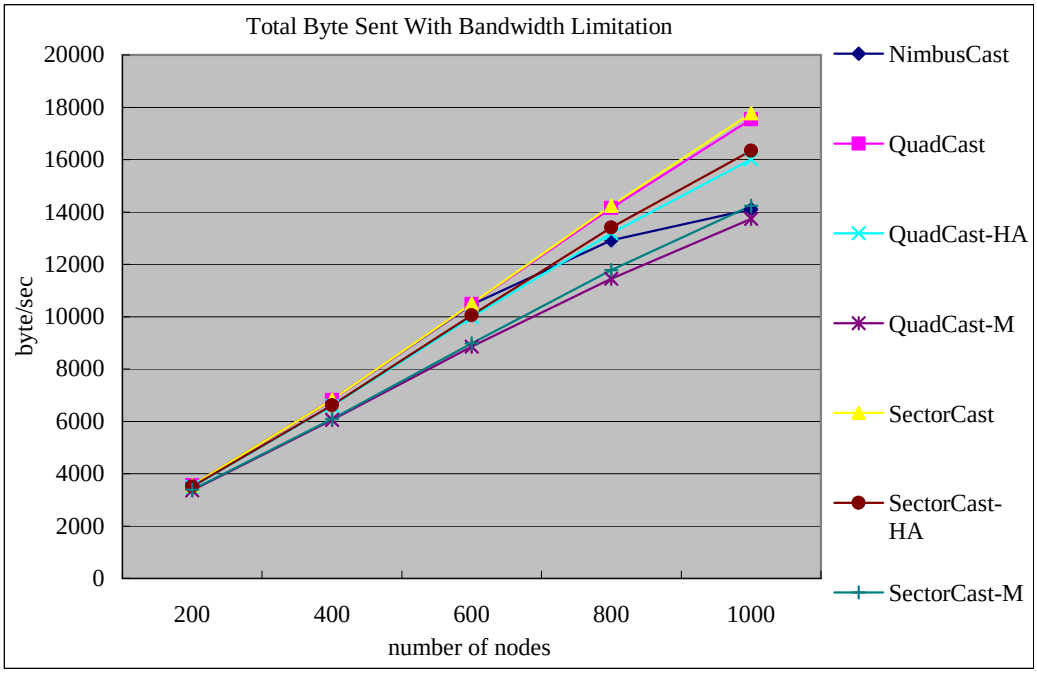



Figure 13: Total byte sent under bandwidth limitation.

## 5.3 Average end-to-end latency

We also simulated the average end-to-end latency. As we mentioned in Chapter 3, the end-to-end latency in a conversation should not exceed 400ms, and a delay below 250ms is considered as good voice quality. Before displaying our simulation results, we first describe our assumption of ene-to-end latency. We use the same assumption of end-to-end latency as in [16], the transmitting latency between hops is assumed as 70ms, and the processing time is 30ms. This means when a voice packet is forwarded through one more hop, the end-to-end latency is increased by 100ms.

In Figure 15, NimbusCast has the shortest end-to-end latency because a node delivers all voice packets directly, so all voice packets are only delivered though one hop. QuadCast has about 250ms end-to-end latency when the number of nodes is 1000. SectorCast has better latency performance than QuadCast because it divide the nearby nodes by equal amount, which yields smaller latency.

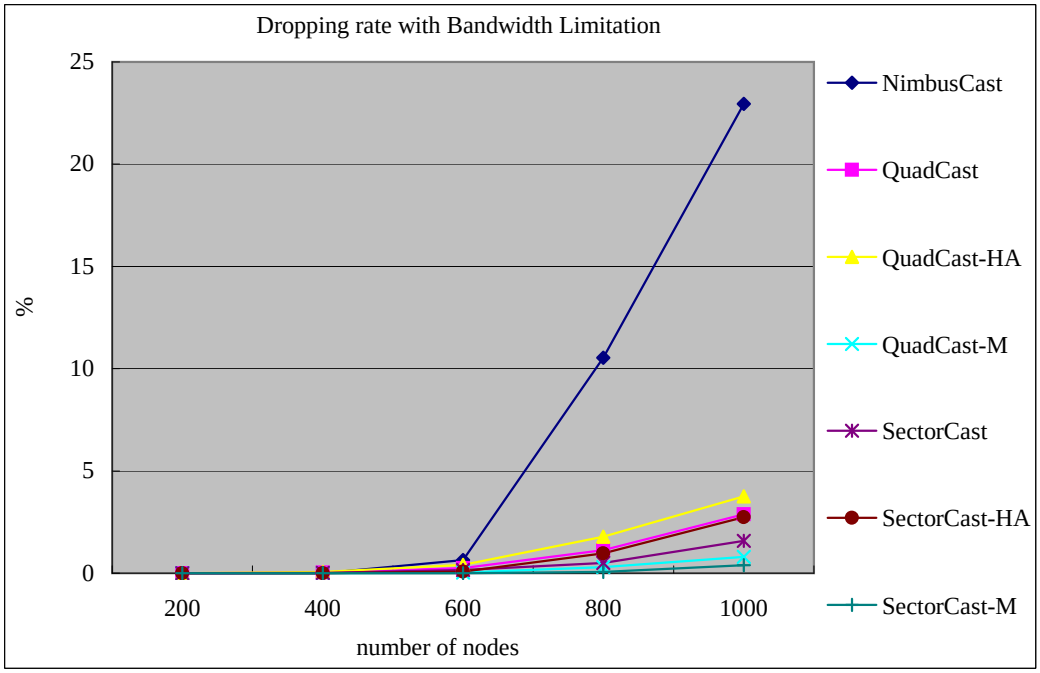



Figure 14: Drooping rate under bandwidth limitation.

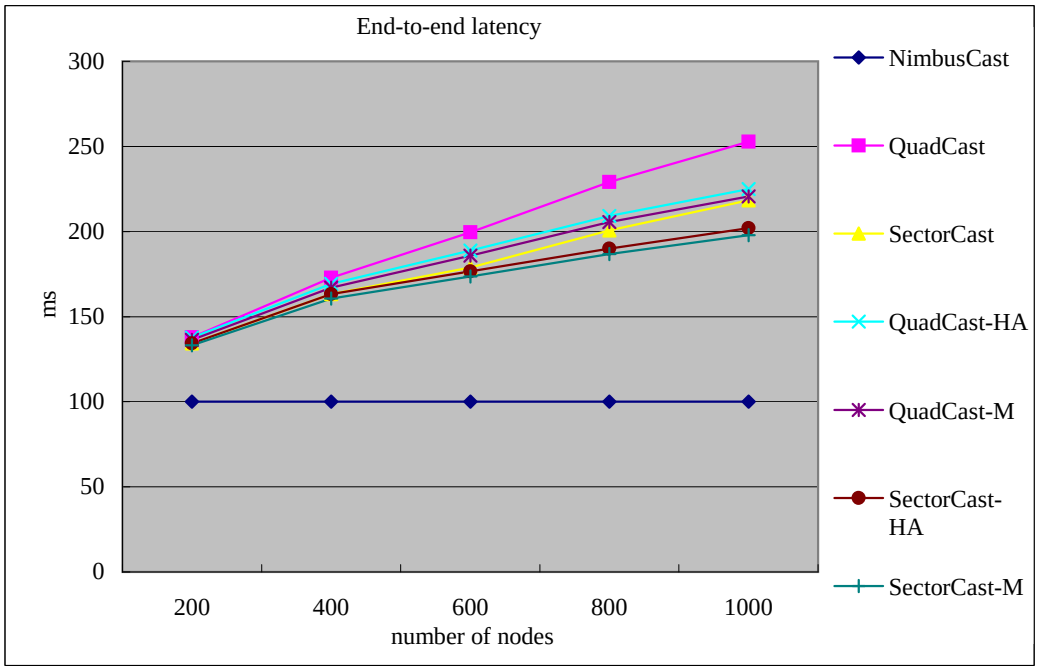



Figure 15: Average end-to-end latency.

# 6 Conclusion

In this thesis, we first describe the concept of AOI voice chatting and propose two solutions to achieve AOI voice chatting. By AOI voice chatting, a player in MMOG can talk with other players within his/her AOI. We first introduce NimbusCast as a basic solution, by which each node delivers all voice packets to all nodes within its AOI. This solution has the shortest end-to-end latency because all voice packets are only delivered through one hop. However, when the number of nodes in AOI increase, the burst bandwidth usage problem shows up, which introduces high packet dropping rate.

We then propose QuadCast. In QuadCast, a speaking node only sends four voice packets at a time, and chooses four forwarding assistants (FAs) to help forward voice packets to remaining recipients within AOI. The packet forwarding solves the burst bandwidth usage problem by the concept of bandwidth sharing. However, the packet forwarding also increases the end-to-end latency due to the additional transmission latency and the processing latency. We also observe that players will cluster together in some regions in MMOG. When QuadCast is adopted, nodes in a certain quadrant might suffer longer end-to-end latency because the number of nodes in that quadrant is much more than those in other quadrants. We proposed SectorCast to solve the problem. In SectorCast, the number of nodes in each sector is the same, which yields shorter average end-to-end latency than QuadCast. The simulation results of average end-to-end latency show that both QuadCast and SectorCast do fulfill the requirement that the end-to-end latency should not exceed 400ms.

# References

[1] Skype, http://www.skype.com/.

[2] Teamspeak, http://www.goteamspeak.com/.

[3] Ventrilo, http://www.ventrilo.com/.

[4] Western world mmog market: 2006 review and forecasts to 2011, http://www.screendigest.com/reports/07westworldmmog/readmore/view.html/.

[5] World of warcraft, http://www.worldofwarcraft.com/.

[6] L. Ding and RA Goubran. Assessment of effects of packet loss on speech quality in VoIP. *Haptic, Audio and Visual Environments and Their Applications, 2003. HAVE 2003. Proceedings. The 2nd IEEE Internatioal Workshop on*, pages 49–54, 2003.

[7] S.Y. Hu, J.F. Chen, and T.H. Chen. VON: a scalable peer-to-peer network for virtual environments. *Network, IEEE*, 20(4):22–31, 2006.

[8] Sunghwan Ihm Tcaesvk Gim Jinwon Lee, Hyonik Lee and Junehwa Song. Apolo: Ad-hoc peer-to-peer overlay network for massively multi-player online games. Technical report, 2005.

[9] J. Keller and G. Simon. Solipsis: A massively multi-participant virtual world. *Proc. of PDPTA*, pages 262–268, 2003.

[10] LS Liu and R. Zimmermann. Immersive peer-to-peer audio streaming platform for massive online games. *Consumer Communications and Networking Conference, 2006. CCNC 2006. 2006 3rd IEEE*, 2, 2006.

[11] K.L. Morse, L. Bic, and M. Dillencourt. Interest management in large-scale virtual environments. *Presence: Teleoperators and Virtual Environments*, 9(1):52–68, 2000.

[12] C.D. Nguyen, F. Safaei, and D. Platt. On the provision of immersive audio communication to massively multi-player online games. *Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on*, 2.

[13] International Telecommunication Union. *ITU-T Recommendation P.59 Artificial conversational speech*, 1993.

[14] International Telecommunication Union. *ITU-T Recommendation G.729 Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)*, 1996.

[15] International Telecommunication Union. *ITU-T Recommendation G.114 One-way transmission time*, 2003.

[16] R. Zimmermann and L.S. Liu. ACTIVE: adaptive low-latency peer-to-peer streaming. *Proc. SPIE*, 5680:26–37, 2005.