



Dynamic Channel Allocation for Wireless Zone Based Multicast and Broadcast Service

K. Selvamalathi

Lecturer (Senior Grade), Department of Computer Engineering Ayya Nadar Janaki Ammal
Polytechnic College, Sivakasi, Tamil Nadu, India

ABSTRACT

In wireless Multicast Broadcast Service (MBS), the common channel is used to multicast the MBS content to the Mobile Stations (MSs) on the MBS calls within the coverage area of a Base Station (BS), which causes interference to the dedicated channels serving the traditional calls, and degrades the system capacity. The MBS zone technology is proposed in Mobile Communications Network (MCN) standards to improve system capacity and reduce the handoff delay for the wireless MBS calls. In the MBS zone technology, a group of BSs form an MBS zone, where the macro diversity is applied in the MS, the BSs synchronize to transmit the MBS content on the same common channel, interference caused by the common channel is reduced, and the MBS MSs need not perform handoff while moving between the BSs in the same MBS zone. However, when there is no MBS MS in a BS with the MBS zone technology, the transmission on the common channel wastes the bandwidth of the BS. It is an important issue to determine the condition for the MBS Controller (MBSC) to enable the MBS zone technology by considering the QoS for traditional calls and MBS calls. In this paper, we propose two Dynamic Channel Allocation schemes: DCA and EDCA by considering the condition for enabling the MBS zone technology. Analysis and simulation experiments are conducted to investigate the performance of DCA and EDCA.

KEYWORD: *Broadcast Channels, Channel Allocation, Mobility Management (Mobile Radio), Multicast Communication, Quality of Service, Radiofrequency Interference, Telecommunication Services*

I. INTRODUCTION

To improve the system capacity and reduce the handoff delay for wireless Multicast and Broadcast Service (MBS), the MBS zone technology is being proposed in several Mobile Communications Network (MCN) standards, such as IEEE 802.16-2009, Enhanced MBS Zone in IEEE 802.16m, and Multimedia Broadcast multicast service Single Frequency Network Area in 3GPP Universal Mobile Telecommunications System (UMTS) and Long Term Evolution (LTE). Figure 1 illustrates the general MCN architecture with the MBS zone technology. The MBS Controller (MBSC; Figure 1 (3)) accommodates the functionalities including the MBS zone management, service announcement, membership management, security management, session management, session transmission, multicast connection identifier and IP address management. For more details of the functionalities, readers may refer. The MBS Server (Figure 1 (1)) is the MBS content provider. The Multicast Router (Figure 1 (2)) performs multicast routing (e.g., IGMP and PIM-SM) for the MBS content in the IP network. Two types of logical channels (including common channel and dedicated channel) are used to transmit the MBS content.

Figure illustrates the channel pool consisting of common channels and dedicated channels. The transmission of a dedicated channel is point-to-point between the Base Station (BS) and the Mobile Station (MS), which serves either an MBS call or a traditional call (i.e., unicast packet-switched or circuits' switched call). The transmission of a common channel is point-to-multipoint between the BS and a group of MSs.

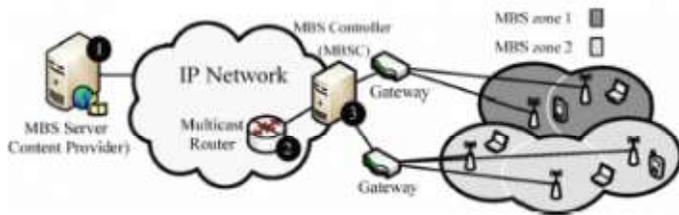


Fig. 1. The MCN architecture with the MBS zone technology

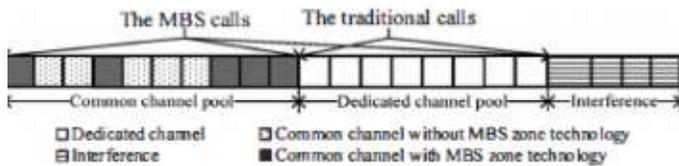


Fig. 2. The logical channel pool of a BS

The common channel can be used to multicast the MBS content to all MSs subscribing the same MBS in the coverage area of the BS (known as the cell). Compared with the dedicated channel, there is no associated control channel for the common channel to report the channel quality of the MS. To cover the whole cell, the transmission power of a common channel should be large enough. Therefore, the usage of a common channel may result in higher interference to the dedicated channels. It is recommended to use a common channel to deliver the MBS content while the number of MSs listening to the MBS content is sufficiently large. In the MBS zone technology, an MBS zone consists of a group of BSs synchronized to transmit the same MBS content using the same common channel. In the overlapped area of two or more BSs, the signals on the same common channel of different BSs can be aggregated at the MS (known as the macro diversity [1]). As pointed out in the studies, the transmission power required for a common channel with the MBS zone technology is less than that without the MBS zone technology due to the macro diversity, and the interference caused by the common channels is reduced. Suppose that it damages d_z and d_e (where d_z ...; d_e) dedicated channels to turn on a common channel with the MBS zone technology and without the MBS zone technology, respectively. The analysis of the values for d_z and d_e is out of the scope of this paper, and readers may refer to for more details. The channel allocation scheme of the standard MBS zone technology (namely scheme Basic) is described as follows. To simplify our discussion, we assume one MBS service in the MBS network, and assume that there are K cells in Z , and $Z = \{\text{Cell}_h, \text{Cell}_2', \dots, \text{Cell}_{K-1}' \text{ Cells}\}$. Suppose that an MS currently resides in Cells . For Z , the variable zone is maintained in the

MBSC. The zone = enabled (zone = disabled) indicates that the MBS zone technology is enabled (disabled) in Z . Scheme Basic consists of six procedures: Service Flow Creation (SFC), Service Flow Deletion (SFD), Procedure Service Join (PSJ), Procedure Service Leave (PSL), intra-MBS zone handoff, and inter-MBS zone handoff.

To make a traditional call, an MS performs SFC with Cells to obtain a dedicated channel. If there is an idle dedicated channel in Cells, the new traditional call is accepted. Otherwise, the new traditional call is blocked, which is referred as "new traditional call blocking". To complete a traditional call, the MS performs SFD with Cells, and a dedicated channel is released. The details of SFC and SFD can be found. To join the MBS service, the MS executes PSJ with Cells and the MBSC to obtain a common channel. For a new MBS call arrival, the MBSC checks whether the data flow to Z for the MBS service exists. If the data flow does not exist, the MBSC negotiates with all cells in Z for bearer resources, and establishes the MBS data flow to all cells in Z . Then, the MBSC sets zone to enabled. All cells in Z turn on the same common channel to activate the MBS zone technology. Afterward, the MS can receive the MBS content from the common channel of Cells. To disjoin the MBS service, the MS executes PSL with Cells and the MBSC. When an MBS call is completed, the MBSC determines whether to deactivate the MBS zone technology in Z . If there is no MBS calls in Z , the MBSC informs all cells in Z to turn off the same common channel to deactivate the MBS zone technology. The MBSC terminates the MBS data flow to Z , and sets zone to disabled. During a traditional call or an MBS call, an MS may move from the old cell Cello to the new cell Cell_n , which is referred as "handoff". The MS performs the MAC Layer Handoff procedure to synchronize the downlink of Cell_n . For a handoff traditional call, the MS performs SFC with Cell_n to re-establish the traditional call in Cell_n . If an idle dedicated channel in Cell_n is available, the handoff traditional call request is accepted. Otherwise, the handoff traditional call is forced to terminate in Cell_n . Suppose that Cello and Cell_n belong to Z_o and Z_n , respectively.

The existing system is usually called a time-free asynchronous distributed system prone to process crashes. In these systems, a system designer can only assume an upper bound on the number of processes that can crash and, consequently, design protocols

relying on the assumption that at least processes are alive. The protocol has no means to know whether a given process is alive or not.

Our model provides upper-layer applications with process state information according to the current system synchrony (or QoS). The underlying system model is hybrid, comprised of a synchronous part and an Asynchronous part. However, such a composition can vary over time in such a way that the system may become totally synchronous or totally asynchronous

II. Feasibility Study

A feasibility study is a test of the system proposal regarding to its work ability, impact on the organization, ability to meet user needs, and effective use of resources. Thus when a new application is proposed, it normally goes through a feasibility study before it's approved for development.

An important outcome of the preliminary investigation is the determination that the system requested is feasible.

The feasibility study was conducted under these aspects.

TECHNICAL FEASIBILITY:

Technical feasibility deals with hardware as well as software requirement. In the feasibility study scope was whether the work for the project could be done with the current equipment and the existing software technology. The outcome of the study was positive. It was found that all software and hardware specification were available.

OPERATIONAL FEASIBILITY:

The purpose of the operational feasibility study was to determine whether the new system would be used if it is developed and implemented. Will there be resistance from the users that will undermine the possible application benefits? From the output of the meeting that was help with the system user, it was found that all of them supported the development of the new system. The positive response from them encouraged us in the development of the new system.

ECONOMIC FEASIBILITY:

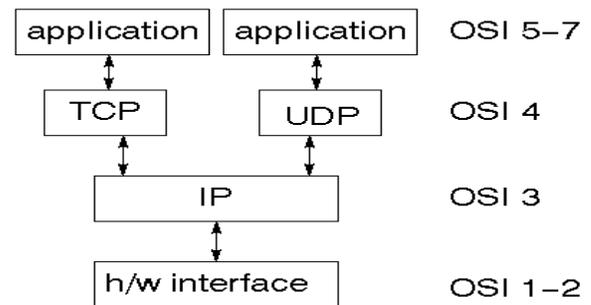
Economic feasibility is the most frequently used method for evaluating the effectiveness of a candidates system. More commonly known as cost/benefit analysis, the procedure is to determine the

benefits and savings that are accepted from a candidate system and compare them with costs. If benefits outweigh costs, further justification and alterations in the proposed system will have to be made if it is to have a chance of begin approved.

III. Networking

TCP/IP stack

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

Network address

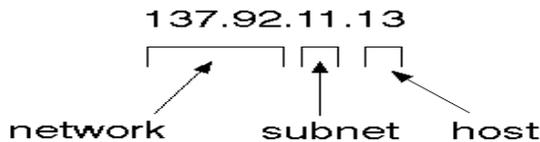
Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

Total address**Port addresses**

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

- sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);
- creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XY Plot class in J Free Chart;
- Testing, documenting, testing some more, documenting some more.

2. Time Series Chart Interactivity

Implement a new (to J Free Chart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time

series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of J Free Chart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

4. Property Editors

The property editor mechanism in J Free Chart only handles a small subset of the properties that can be set for charts. Extend (or reimplement) this mechanism to provide greater end-user control over the appearance of the charts.

IV. Design Analysis**Data Flow Diagrams**

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart.

Context Diagram:

The top-level diagram is often called a "context diagram". It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

V. System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its

requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Types of Tests

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional Test

Functional tests provide a systematic demonstration that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input:

Identified classes of valid input must be accepted.

Invalid Input:

Identified classes of invalid input must be rejected.

Functions:

Identified functions must be exercised.

Output:

Identified classes of application outputs must be exercised.

Systems/Procedures:

Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify

Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the Software without Any Knowledge of The Inner Workings, Structure or Language of The Module Being Tested. Black Box Tests, As Most other Kinds of Tests, Must Be Written From a Definitive Source Document, Such As Specification or Requirements Document, Such As Specification or Requirements Document. It is a Testing in Which the Software under Test is treated; As a Black Box .You cannot "See" Into It. The Test Provides Inputs And Responds To Outputs Without Considering How The Software Works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

VI. Implementation

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods

to achieve changeover and evaluation of changeover methods.

Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization.

The implementation can be preceded through Socket in java but it will be considered as one to all communication .For proactive broadcasting we need dynamic linking. So java will be more suitable for platform independence and networking concepts. For maintaining route information we go for SQL-server as database back end.

VII. Conclusion

This paper proposed DCA and EDCA for more flexible channel allocation for MBS, which improve the S f performance performance for users. However, more signalling overheads are caused by DCA and EDCA. Therefore, we proposed analytical and simulation models to study the S f performance for Basic, DCA, and EDCA. Our study provides the following guidance for MBS network operators when to use Basic, DCA, or EDCA:

- When the macro diversity functions well (i.e., d_z is small), Basic is the good choice since the S f performance for the three schemes are almost the same. On the other hand, when d_z is large, EDCA is the better choice.
- As the number of MBS calls in a cell is larger (i.e., higher $AO''i$ lower T/i , or lower $f. Lm$), EDCA is the better choice.
- The performance enhancement of EDCA over DCA decreases as $AO\,i$ increases. Therefore, when traditional call arrival rate $AO\,i$ is higher enough, DCA is suggested.
- The performance trend for the three schemes for different MS mobility patterns are almost the same.

VIII. References

1. IEEE. IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Broadband Wireless Access Systems. IEEE STD 802.16-2009, May 2009.
2. IEEE. IEEE 802. I 6m System Description Document. IEEE STD 802.16m, December 2008.

3. WiMAX Forum. WiMAX Forum Network Architecture Release 1.5 Version I Draft 0 - Stage 3: Detailed Protocols and Procedure. November 2009.
4. Wang, J., Venkatachalam, M., and Fang, Y. System Architecture and Cross-Layer Optimization of Video Broadcast over WiMAX. IEEE Journal on Selected Areas in Communications, 25(4):712-721, May 2007.
5. Jeng, J.-Y., and Lin, Y.-B. Equal Resource Sharing Scheduling for PCS Data Services. ACM Wireless Networks, 5(1):41-55, January 1999.
6. Lin, Y.-B., and Yang, S.-R. A Mobility Management Strategy for GPRS. IEEE Transactions on Wireless Communications, 2(6): 1178-1188, November 2003.
7. Lin, P. and Lin, Y.-B. Channel Allocation for GPRS. IEEE Transactions on Vehicular Technology, 50(2):375-387, March 2001.
8. Lin, P. Channel Allocation for GPRS with Buffering Mechanisms. ACM Wireless Networks, 9(5):431-441, September 2003.
9. Etemad, K., and Wang, L. Multicast and Broadcast Multimedia Services in Mobile WiMAX Networks. IEEE Communications Magazine, 47(10):84--91, October 2009.
10. Lai, y-c., Lin, P., Fang Y, and Chen, W.-H. Channel Allocation for UMTS Multimedia Broadcasting and Multicasting. IEEE Transactions on Wireless Communications, 7(11):4375-4383, November 2008.

