**Dynamic Channel Allocation for Wireless Zone-Based Multicast and Broadcast Service**

**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.

**Existing System:**

- 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.

Proposed System:

- 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.

System Requirements

Hardware:

- PROCESSOR: PENTIUM IV 2.6 GHz
- RAM: 512 MB
- MONITOR: 15”
- HARD DISK: 20 GB
- CDDRIVE: 52X
- KEYBOARD: STANDARD 102 KEYS
- MOUSE: 3 BUTTONS

Software:

- FRONT END: JAVA, SWING
- TOOLS USED: JFrame Builder
- OPERATING SYSTEM: WINDOWS XP

Modules:

- Identify the status of Node
- Message Transmission
- Change status
- Update status
Identify the Status Node:

In this Module we identify the Node is weather live or not. In this process we easily identify the status of the node and also easily identify the path failure.

Message Transmission:
In the module we just transfer the message to the destination or intermediate nodes. The intermediate node just forwards the message to destination. The receiver receives the message and sends the Ack.

Change Status:
In this Module we identify the changed status of node. The Status is

- Live
- Uncertain
- Down

Update Status:
In this module we update the status of the node. Then only we can identify whether the node is live or not.

REFERENCE: