3. Discrete-Rate System Model

The wireless forward link data channel is considered as a discrete rate system. In section 3.1, we describe how the forward link with rate control operates. In Section 3.2, we discuss how a MT chooses one data rate from a finite set of feasible data rates. A baseline time-based admission control algorithm is presented in Section 3.3.s

3.1 Forward link system

We consider a cell with a single forward link channel whose resource is time-shared between active MTs. Moreover, time is slotted, and a forward-link service scheduler assigns each slot to serve at most one MT. Additionally, assume that the BST has perfect information of feasible data rate for each MT. This can be achieved through sending feedback information over reverse channels. For example, the data rate control (DRC) message in [15] is periodically sent over DRC channel in reverse link to inform the system of the supportable forward link data traffic rate and the best serving sector for the forward link.

Consider that there is a memory queue associated with each service request. The memory queue buffers all data traffic to be served from the request. If a new or buffered service request is accepted for service, its memory queue is attached to the end of a queue bank whose requests have been accepted. Since the service scheduler adopts the Round-Robin policy to serve admitted service requests, it simply allocates time slots sequentially to each admitted MT which has data in the bank of memory queues. The data rate in each time slot is assumed to be constant, with a value according to feedback rate information. This requires that channel correlation time lasts much longer than a slot.

3.2 Discrete rate set

As described in Section 2.2, data rate function (2.1) or (2.2) can be used to obtain a feasible data rate for a target MT. The way is that each MT measures the received SINR of pilot signals from BST, uses either (2.1) or (2.2) to obtain an estimate of feasible data rate, and then sends the information over DRC channel to networks. This thus gives rise to continuous feasible rates. Practically, only a finite set of feasible data rates can be selected for transmission, due to the limitations of available channel coding, modulation, and spreading code schemes.

We thus consider only a finite set of feasible rates available. The minimum SINR required to support each data rate can be computed from (2.1) or (2.2) with E_b/N_0 set to a constant corresponding to the chosen modulation scheme and C set to the data rate. Hence, there will be a set of minimum SINR levels corresponding to the set of feasible data rates. The MT thus can choose a feasible data rate that corresponds to a minimum SINR level directly below but closest to its received SINR.

The study considers a single cell with omni-directional antenna, where only a few discrete feasible data rates, derived from the peak data rate function (2.3), are supported. Such assumptions have been made and well accepted in the literature, and are sufficient for our study of fundamental issues. Specifically, we consider a single cell configuration with circular radio coverage area, as shown in Figure 3.1. If a MT is located within a ring area with radius r_0, normalized to 1, its peak feasible data rate is C_0. For distance-related path-loss exponent $\alpha = 4$ in (2.3), the largest distance from BST for the forward link to support data rates $C_0/2$, $C_0/4$, $C_0/8$ are r=1.19, 1.41, and 1.68, respectively. These are listed in Table 3.1 with C_0=2457.6 kbps. Thus, a MT located at a distance between [1, 1.19], [1.19, 1.41], or [1.41, 1.68] is served in its scheduled time slot with the instantaneous data rate $C_0/2$, $C_0/4$, and $C_0/8$, respectively. This gives four data rates in four ring areas as illustrated in Figure 3.1.

Table 3. 1 Data rates and ring radius.

Ring k	Rate c_k (Kbit/s)	Radius r_k
0	2457.6	1
1	1228.8	1.19
2	614.4	1.41
3	307.2	1.68

It is possible to have a larger set of feasible data rates by increasing the radius of radio coverage area, using puncture technique between basic codes, or using hybrid ARQ. However, we should not exhaust the physical layer issues in the thesis.

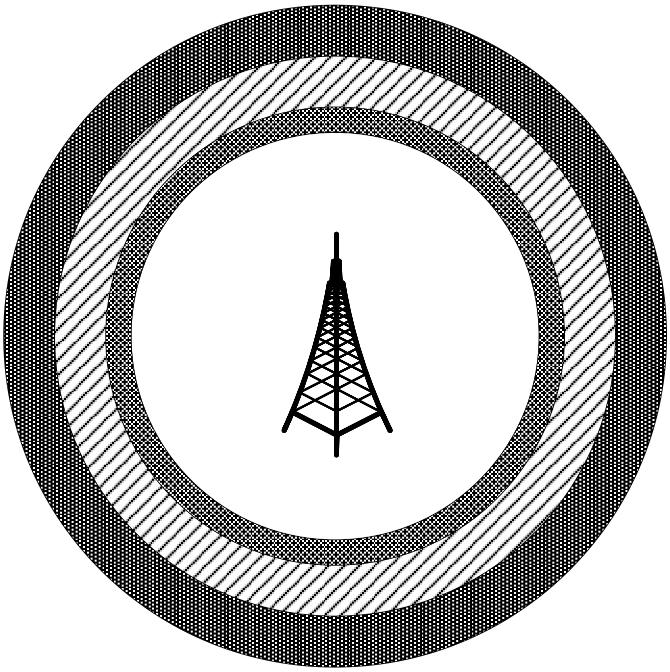

Figure 3. 1 A single cell with four ring areas for four different feasible data rates.

3.3 Time-based admission control

The forward link data service can be viewed as a work-conservative queueing system with one server, one service queue, and one waiting queue. Service requests that have been admitted but not finished are buffered in *service queue*. A new accepted request is attached to the end (or tail) of service queue. Let $N_q(t)$ be the number of unfinished service requests in service queue at time t. (Let $W_q(t)$ be the remaining work, residual service time, required to empty the service queue if no requests are accepted into service queue after time t. Then $W_q(t)=\sum_{i=1}^{N_q(t)} S_i(t)$, where $S_i(t)$ is the remaining work required to complete the service of admitted request *i*.) Service requests that have not been admitted and blocked are buffered in *waiting queue*. Let $N_w(t)$ be the number of requests in waiting queue at time t. A new service request that can not be accepted into service queue is buffered at the end (or tail) of waiting queue. The system then starts up a countdown timeout timer $T\_out_{N_w}(t)$ for the request. If the request can't be accepted into service queue when the timer goes off, it is discarded and removed from waiting queue.

We use in time-based admission control three thresholds, namely *busy threshold*, *admission threshold*, and *dropping threshold*, which are denoted by T_b, T_a, and T_d, respectively. Let Q(t) be a measure of system busy status at time t and S be the resource required to service a request.

The baseline strategy of admission control is that:

- The system of admission control alternates between ON and OFF states according to rule (a) in the following, which depends on the monitored system busy status Q(t) compared with busy threshold T_b.
- When a service request which requires channel resource S arrives, it is either accepted into service queue if $S \leqq T_a$ or the admission control mechanism is turned OFF, or buffered in waiting queue and governed by a timeout timer if $S > T_a$ and the admission control mechanism is turned ON. The algorithm to deal with such an arrival event is described in rule (b).
- If there exist a request, which requires channel resource size S, at the head of waiting queue, the system keeps track of opportunities to accept it into service queue; That is, the request is accepted into service queue when $S \leqq T_a$ or $Q(t)<T_b$, or removed from waiting queue and discarded when its timeout timer goes off. This is described in rule (c) in the following.
- When there exists unfinished work in service queue, the forward link system uses round-robin service discipline slot by slot, as described in rule (d).

a) While time t>0, the state of admission control is

```
    If Q(t) < T_b
        Admission control is OFF;
    Else
        Admission control is ON
    End if
```

b) A service request requiring channel resource size S arrives at time t, then

```
If Q(t)<T_b    // Admission control is OFF.
    $N_q$(t)← $N_q$(t)+1 and push S in service queue    // Accept the request
Else if S≦Ta  // Admission control is ON and requested resource size is less than Ta
    $N_q$(t)← $N_q$(t)+1 and push S into service queue // Accept the request
Else $N_w$(t)←$N_w$(t)+1, push S into waiting queue, and start timer $T\_out_{N_w}(t) \leftarrow T_d$.
End if
```

c) The head of waiting queue with time t, which requires channel resource size S:

```
While ($N_w$(t)>0)
    If Q(t)<T_b or S≦T_a
        $N_q$(t)←$N_q$(t)+1, $N_w$(t)←$N_w$(t)-1, pop S from the head of waiting queue and
            push it into service queue    // Accept the head request in waiting queue
    End if
    If $T\_out_1(t) = 0$  //The head request in waiting queue has waited for T_d time
    unit.
        $N_w$(t)←$N_w$(t)-1, pop S from the head of waiting queue and discard the
    request.
    End if
End while
```

d) Round-robin service in slot time $\left(\Delta\left\lfloor \frac{t}{\Delta} \right\rfloor, \Delta\left\lceil \frac{t}{\Delta} \right\rceil\right)$

Let Δ be the length of a slot time, $\lceil x \rceil$ and $\lfloor x \rfloor$ represent two consecutive integers satisfying $\lfloor x \rfloor \leq x < \lceil x \rceil$.

```
While ( $N_q\left(\Delta\left\lfloor \frac{t}{\Delta} \right\rfloor\right)$>0)
```

```
        i ← (i mod $N_q(\Delta\lfloor \frac{t}{\Delta} \rfloor)$)+1;   // The next MT’s turn

        $S_i(\Delta\lceil \frac{t}{\Delta} \rceil) \leftarrow S_i(\Delta\lfloor \frac{t}{\Delta} \rfloor) - \Delta$, while serve one slot’s worth of data for the MT;

        If   $S_i(\Delta\lceil \frac{t}{\Delta} \rceil) \leqq 0$          //If the service of a service request completes,

            For (j=i ; j< $N_q(\Delta\lfloor \frac{t}{\Delta} \rfloor)$; j++)

                $S_j(\Delta\lceil \frac{t}{\Delta} \rceil) \leftarrow S_{j+1}(\Delta\lfloor \frac{t}{\Delta} \rfloor)$;     // Push remaining works in service queue

            End for

            $S_{j+1}(\Delta\lceil \frac{t}{\Delta} \rceil) \leftarrow 0$                    // A service departure

            $N_q(\Delta\lceil \frac{t}{\Delta} \rceil) \leftarrow N_q(\Delta\lfloor \frac{t}{\Delta} \rfloor) - 1$           // A service departure

          End if
End while
```