The Optimal Uniform Schedules of Arbitrary Static Permutations on Superposed Parallel Buses

Bruce W. Arden

College of Engineering and Applied Science
University of Rochester
Rochester, N.Y. 14627

Toshio Nakatani

Department of Computer Science
Princeton University
Princeton, N.J. 08544

CS-TR-033-86

April, 1986

The Optimal Uniform Schedules of Arbitrary Static Permutations on Superposed Parallel Buses

Bruce W. Arden

College of Engineering and Applied Science
University of Rochester
Rochester, N.Y. 14627

Toshio Nakatani

Department of Computer Science
Princeton University
Princeton, N.J. 08544

ABSTRACT

The paper is concerned with the optimal schedule for the permutation of n^2 packetes on an $n \times n$ square grid of superposed, parallel time multiplexed buses. It is shown that *the uniform schedule* (that is, all two-step transfers consistently row first or alternately column first) is optimal. Moreover, it requires $n+1$ bus transfers, or cycle times. Although n-cycle, non-uniform schedules exist for specific permutations, it is shown that $n+1$ cycle, uniform schedule is optimal.

April 21, 1986

The Optimal Uniform Schedules of Arbitrary Static Permutations on Superposed Parallel Buses

Bruce W. Arden

College of Engineering and Applied Science
University of Rochester
Rochester, N.Y. 14627

Toshio Nakatani

Department of Computer Science
Princeton University
Princeton, N.J. 08544

1. Introduction

Consider the permutation of $N=n^2$ packets using two sets of n parallel, time multiplexed buses. If one set of buses is considered to be horizontal and the other vertical, they can be considered to be *superposed* with communication between the horizontal and vertical buses at the bus intersections (Figure 1.1). The time for the transmission of one packet on a bus is considered to be a single *cycle*. With n buses in parallel, there can be at most n simultaneous transmissions on one set of parallel buses. The overlapped, or *pipelined,* use of the two bus planes leads to an overall maximum of $2n$ packet transmissions in a single cycle.

Using the convetional, Cartesian notation for the n^2 bus intersections, a packet destination address $x_k y_l$ is initially present at each of the square array of address $x_i y_j$ ($0 \leq i,j,k,l \leq n-1$). For permutation, all n^2 distinct addresses occur. A permutation is accomplished by transferring a packet from $x_i y_j$ to $x_k y_l$ for all i,j, where $x_k y_l$ was the packet destination address initially in the $x_i y_j$ position. In the general case, there are two bus broadcast steps to accomplish this transfer, i.e., horizontal then vertical or vertical then horizontal (Figure 1.2). Bus pairs intersecting at $x_i y_l$ or $x_k y_j$ must be used. Clearly, there are *degenerate* cases where source and destination are in the same row or column, $x_i=x_k$ or $y_j=y_l$, and only one bus broadcast is required. Also none is required when $x_i=x_k$ and $y_j=y_l$.

For permutations without degenerate cases, that is, every transfer requires two steps, a total of $2n^2$ bus broadcast steps are necessary. Exploiting the bus parallelism and overlap, it is clear that the best schedule or shortest time to

complete a single permutation would therefore be n cycles. However, to complete a nondegenerate, single permutation within n cycles, n horizontal bus broadcasts and n vertical bus broadcasts must be accomplished every cycle including the first one. For the overlapped operations on the first cycle, *mixed* schedules, or mixtures of both the horizontal-then-vertical and the vertical-then-horizontal transfers, are necessary. It is interesting to know if there exists a minimum, mixed schedule for arbitrary permutations.

On the other hand, Hall's theorem on "Distinct Representatives" guarantees the existence of *a uniform* schedule†, that is, an unmixed schedule when the transfers are uniformly horizontal-then-vertical or vertical-then-horizontal. Because of the inherent serial aspect of the two bus broadcasts, the uniform schedule takes $n+1$ cycles. That is, there can be no bus overlap on the first cycle. The uniform schedule can be regarded *optimal* as well, because sequential permutations could achieve an overlap of the additional cycle and thus complete one of a sequence of permutations every n cycles.

The purpose of this paper is to show that no mixed schedule of n cycles exists for single permutations. Although a mixed, n cycle schedule exists for some permutations, one does not exist for all. Hence, the $n+1$ cycle, uniform schedule is optimal for single permutations on an $n \times n$ square grid of *superposed parallel buses.* In section 2, some notation and definitions are given for later use. In section 3, the main theorem is proved. Finally, in section 4, conclusions are stated.

2. Mathematical Notations and Definitions

In this section, notation and definitions are described for later use.

Definition 1: A $n \times n$ square grid of *superposed parallel buses,* denoted as $SPB(n,n)$, is a system with n^2 processors and two sets of n buses; one, called *the row buses* R_0 through R_{n-1}, can be viewed as horizontal and the other, called *the column buses* C_0 through C_{n-1}, as vertical. A processor is located at each cross point and each processor has two ports; one for a horizontal bus and the other for a vertical bus. *The address* of each processor is denoted by (x,y) or xy, where x represents x-coordinate, called *the column address,* and y represents y-coordinate, called *the row address.* There are n processors on each bus. *The row bus* R_i

† This is called *a self-pipelined* schedule in the previous paper (Arden and Nakatani[1986]).

supports n processors $(i,0)$ through $(i,n-1)$ for $0 \leq i \leq n-1$. *The column bus* C_j supports n processors $(0,j)$ through $(n-1,j)$ for $0 \leq j \leq n-1$. It is assumed in this paper that each processor at the address (x,y) has a packet with its mailing address, called *destination address* $D(x,y)$, and that a mapping from (x,y) to $D(x,y)$ forms a permutation from $SPB(n,n)$ to $SPB(n,n)$.

Definition 2: *A column selection* is a set of n processors at the addresses $(x_0,0)$ through $(x_{n-1},n-1)$, where x_i is an integer which is not necessarily unique $(0 \leq x_i \leq n-1)$. Similarly, *a row selection* is a set of n processors at the addresses $(0,y_0)$ through $(n-1,y_{n-1})$, where y_j need not be unique $(0 \leq y_j \leq n-1)$. *A column-row selection* is a set of n processors that forms both a column and row selections at the same time.

Definition 3: A column selection is called *compatible* when each processor i in a column selection has a packet with the destination address (x_i,y_i) and a set of n processors at the addresses $\{(x_i,y_i)\}$ for $0 \leq i \leq n-1$ forms a row selection. Similarly, a row selection is called *compatible* when each processor i in a row selection has a packet with the destination address (x_i,y_i) and a set of n processors at the addresses $\{(x_i,y_i)\}$ for $0 \leq i \leq n-1$ forms a column selection. A column-row selection is called *perfectly compatible* when it is compatible both as a column selection and as a row selection. Moreover, k sets of column-row selections are called *perfectly k-compatible,* (even if each column-row selection is not necessarily perfectly compatible but) if the processors in k sets of column-row selections have packets with exactly k distinct destination row addresses for each of n destination column addresses or with exactly k distinct destination column addresses for each of n destination row addresses.

Definition 4: *The uniform* schedule is n sequences of the horizontal-then-vertical transfers (or equivalently n sequences of the vertical-then-horizontal transfers) of packets chosen by n series of compatible column selections (or equivalently n series of compatible row selections). This is always possible according to Hall's theorem on "Distinct Representatives" (Hall[1935]). Since each selection is compatible, all the selected packets can be broadacst on the row (the column) buses right after they are broadcast on the column (the row) buses without further delays. Therefore, by the uniform schedule, a single permutaion takes $n+1$ cycles and each of sequential or multiple permutations takes n cycles in pipeline fashion. On the other hand, *a mixed* schedule is n sequences including both the horizontal-then-vertical and the vertical-then-horizontal transfers of packets chosen by a series of n compatible selections. The minimum schedule of n cycles

is possible for some specific permutations using the mixed strategy, but in general $n+1$ is the best that can be achieved. Hence, such schedules are optimal.

Definition 5: A permutation is called *degenerate,* if some pairs of source and destination are in the same row or column. Otherwise, a permutation is called *nondegenerate.*

Example 1: The bit-reversal permutation is nondegenerate and has an n-cycle mixed schedule (Figure 2.1).

Example 2: The permutation $D_1(x,y)$ is a bijection from the $SPB(n,n)$ to $SPB(n,n)$ by the following rule:

$D_1(x,y)=([x-2]_n,[y+1]_n)$† for $y\neq n-1$

$D_1(x,y)=([x-1]_n,[y+1]_n)$ for $y=n-1$

This permutation $D_1(x,y)$ is nondegenerate and has no n-cycle schedule, as will be shown in the following section (see Figure 2.2 and 2.3 for uniform schedules of $D_1(x,y)$).

3. Proof of Optimality of the Uniform Schedule

In this section, the main theorem is proved by showing an existence of a single permutation that requires at least $n+1$ cycles and n cycles, in pipeline fashion, for each of a sequence of permutations.

Theorem: The uniform schedule is optimal for arbitatry permutations on the $n\times n$ square grid of superposed parallel buses. That is, it takes $n+1$ cycles for a single permutation and n cycles in pipeline fashion for each of a sequence of permutations.

Proof: From Lemma 4, the permutation $D_1(x,y)$ has no n-cycle mixed schedule. The uniform $(n+1)$-cycle schedule exists for all permutations and is therefore optimal.□

Lemma 1: The permutation $D_1(x,y)$ is non-degenerate, that is it takes at least n cycles on the $n\times n$ square grid of superposed parallel buses. This can be achieved only by compatible selections.

Proof: Every destination address of $D_1(x,y)$ requires both column and row broadcasts. One cycle of $SPB(n,n)$ can transmit at most n packets on the column buses and at most n packets on the row buses. Therefore, at least $n^2\times 2/(2n)=n$

† $[m]_n=m\ (mod\ n)$

cycles are required and this can be achieved only by compatible selections.□

Lemma 2: Any compatible column selection of $D_1(x,y)$ is also a row selection but not compatible row selection. Therefore, any compatible column selection of $D_1(x,y)$ is a column-row selection but not perfectly compatible.

Proof: For any compatible column selection of $D_1(x,y)$, let $(0,y_0)$ through $(n-1,y_{n-1})$ be the destination addresses of the selected processors' packets. Then, $\{y_j\}$ for $0\leq j\leq n-1$ must form a permutation of integers $\{0,1,...,n-1\}$. Since (x,y_j) appears only on the row y_j+1 for any x, a set of n processors, which have the packets with the destination addresses $\{(j,y_j)\}$ for $0\leq j\leq n-1$, forms a row selection. However, in any compatible column selection of $D_1(x,y)$, exactly one processor must have a packet with the destination address $(x_{n-1},n-1)$ and the other processors must have packets with the destination addresses (x_i,i) for $0\leq i\leq n-2$. Moreover, one processor must be selected from the column $[x_{n-1}-1]_n$ excluding the location $([x_{n-1}-1]_n,0)$, on which any of the processors has a packet with the destination address (x_{n-1},h) for a h $(0\leq h\leq n-2)$. That is, two processors, which have the packets with the destination addresses $(x_{n-1},n-1)$ and (x_{n-1},h), must be selected as a part of any compatible column selection. Therefore, this is not compatible row selection, that is, not perfectly compatible column-row selection.□

Lemma 3: In any compatible column selection of $D_1(x,y)$, there are exactly two processors whose packets have the same destination column address and the rest of the processors have packets with distinct destination column addresses from each other and also from the two processors.

Proof: From the proof of Lemma 2, any compatible column selection contains exactly two processors at the addresses $([x_{n-1}-1]_n,0)$ and $([x_{n-1}-2]_n,h+1)$, which have the packets with the destination addresses $(x_{n-1},n-1)$ and (x_{n-1},h) respectively. The rest of the processors are located at the addresses $\{(j,i)\}$, for any i $(i\neq 0,h+1)$ and any j $(j\neq[x_{n-1}-1]_n,[x_{n-1}-2]_n)$, and have the packets with the destination addresses $\{([j+2]_n,[i-1]_n)\}$. Therefore, the rest of the processors have the packets with distinct destination column addresses $\{[x_{n-1}+t]_n\}$ for $2\leq t\leq n-1$ and only two processors have the packets with the same destination column address x_{n-1}.□

Lemma 4: The permutation $D_1(x,y)$ has no n-cycle mixed schedule.

Proof: From Lemma 3, in any compatible column selection of $D_1(x,y)$, the selected processors' destination column addresses are distinct except for the two. More precisely, the processors in a compatible column selection contains the

destination column addresses of two x_{n-1} and no $[x_{n-1}+1]_m$ and each of the other integers. Because of this rule, for any k $(1 \leq k \leq n)$, the processors in any k sets of compatible column selections cannot have the same number of the aliases for each destination column address except for the case of $k=n$. That is, any k sets of compatible column selections cannot be perfectly k-compatible column-row selections except for the case of $k=n$, when all the compatible selections are the column selections. Therefore, either n compatible column selections or n compatible row selections are the only compatible selections for the whole permutation of $D_1(x,y)$. From Lemma 1, the permutation $D_1(x,y)$ takes at least n cycles and this is possible only with compatible selections. The only way with compatible selections is either by n compatible column selections or by n compatible row selections. That is, the permutation $D_1(x,y)$ has no n-cycle mixed schedule.□

4. Conclusions

In this paper, it is proved that the uniform schedule is optimal for arbitrary static permutations on the square grid of superposed parallel buses. Further studies include the classifications of the permutations that can be completed on the square grid of superposed parallel buses within fewer cycles than by the uniform schedule.

References

Arden, B. W. and T. Nakatani [1986]. "Permutations on superposed parallel buses," Technical Report CS-24, Department of Computer Science, Princeton University.

Hall, P. [1935]. "On representatives of subsets," *J. London Math. Soc.*, 10, pp.26-30.

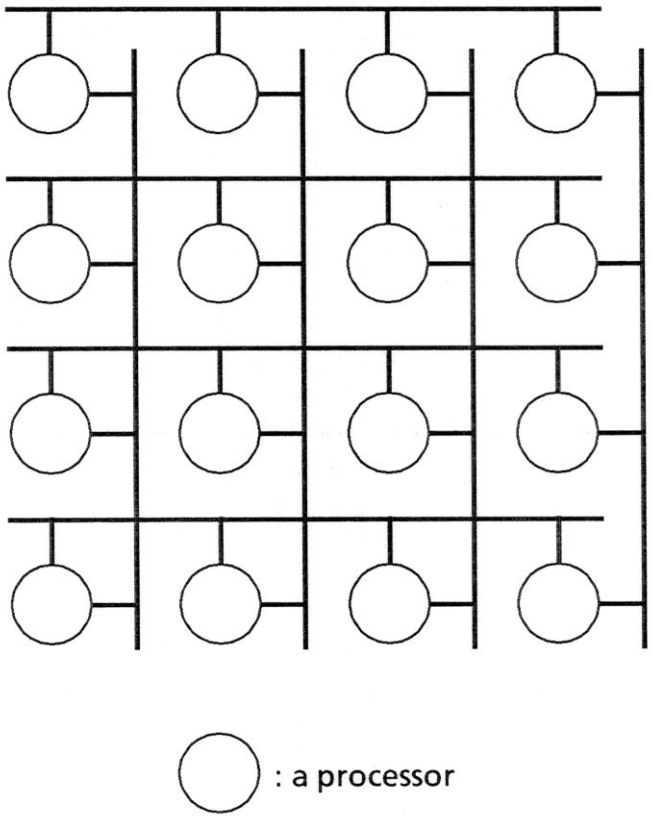

Figure 1.1: A 4x4 square grid of superposed parallel buses

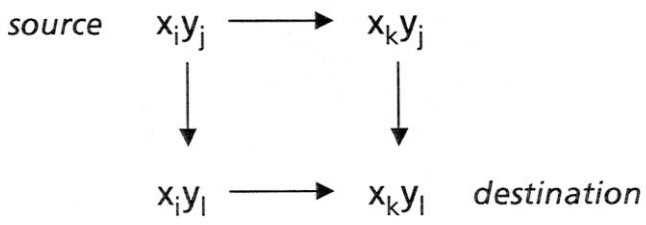

Figure 1.2: A packet transfer from source to destination

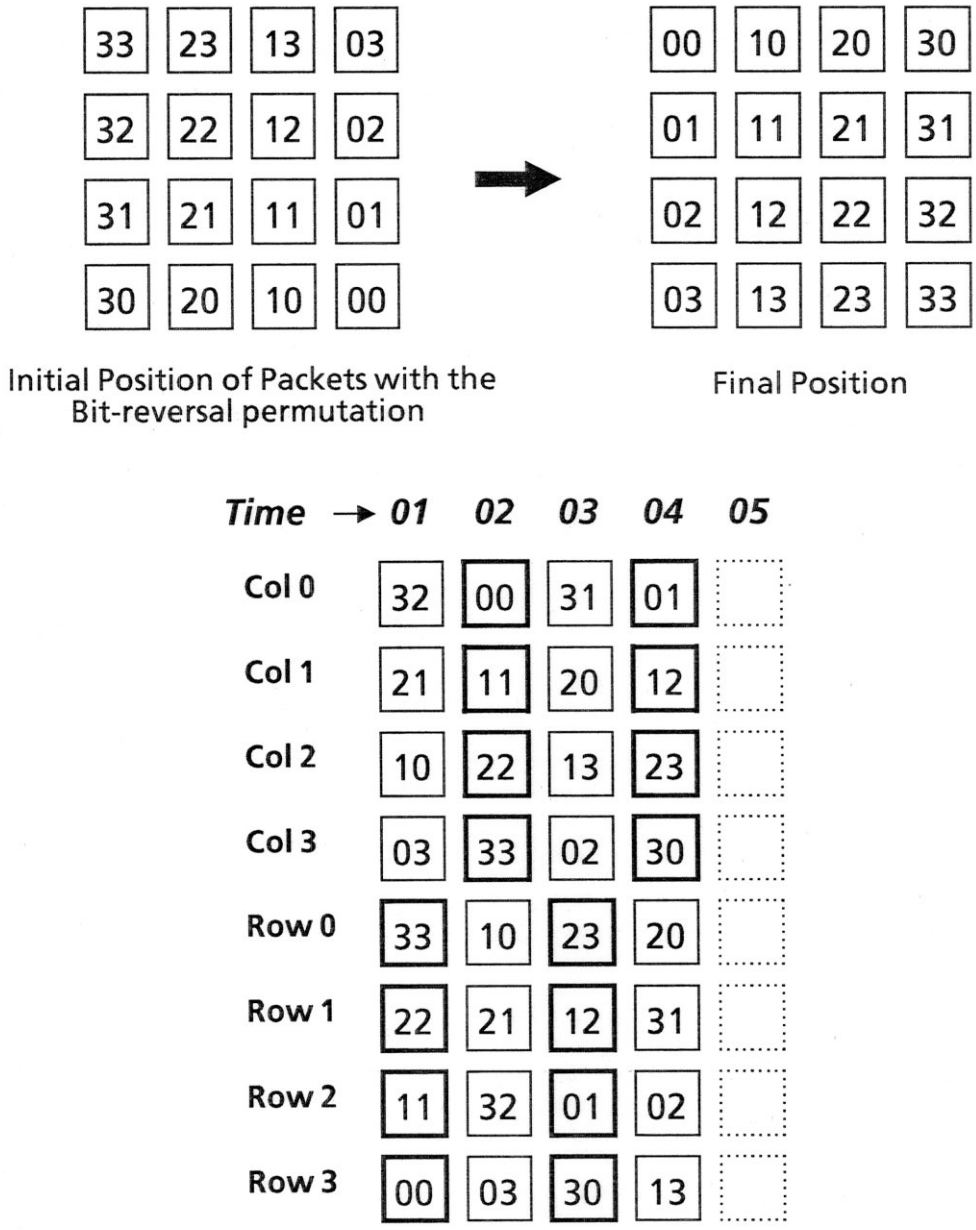

Figure 2.1: The *n*-cycle mixed schedule of the bit-reversal permutation

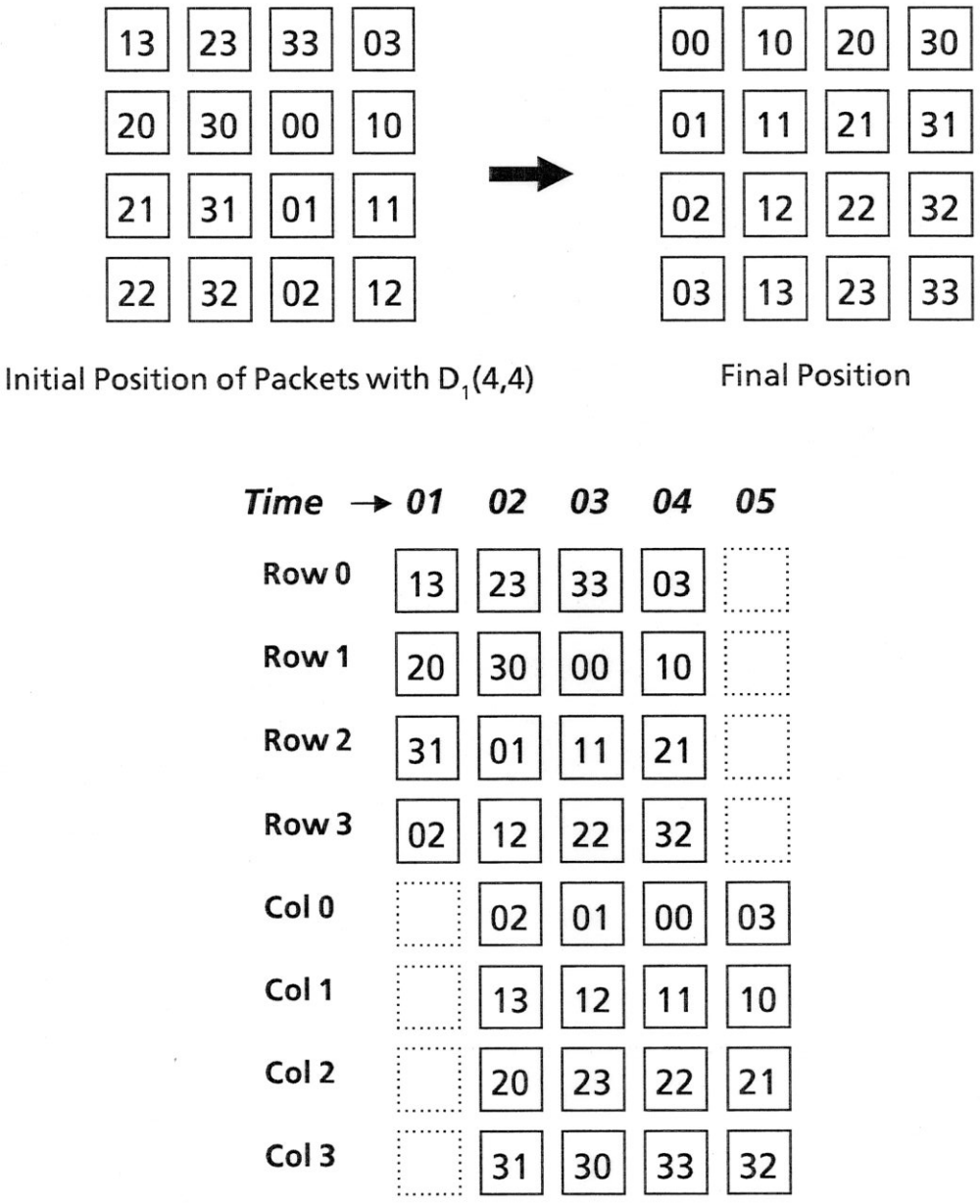

Figure 2.2: The uniform (row-column) schedule of $D_1(4,4)$

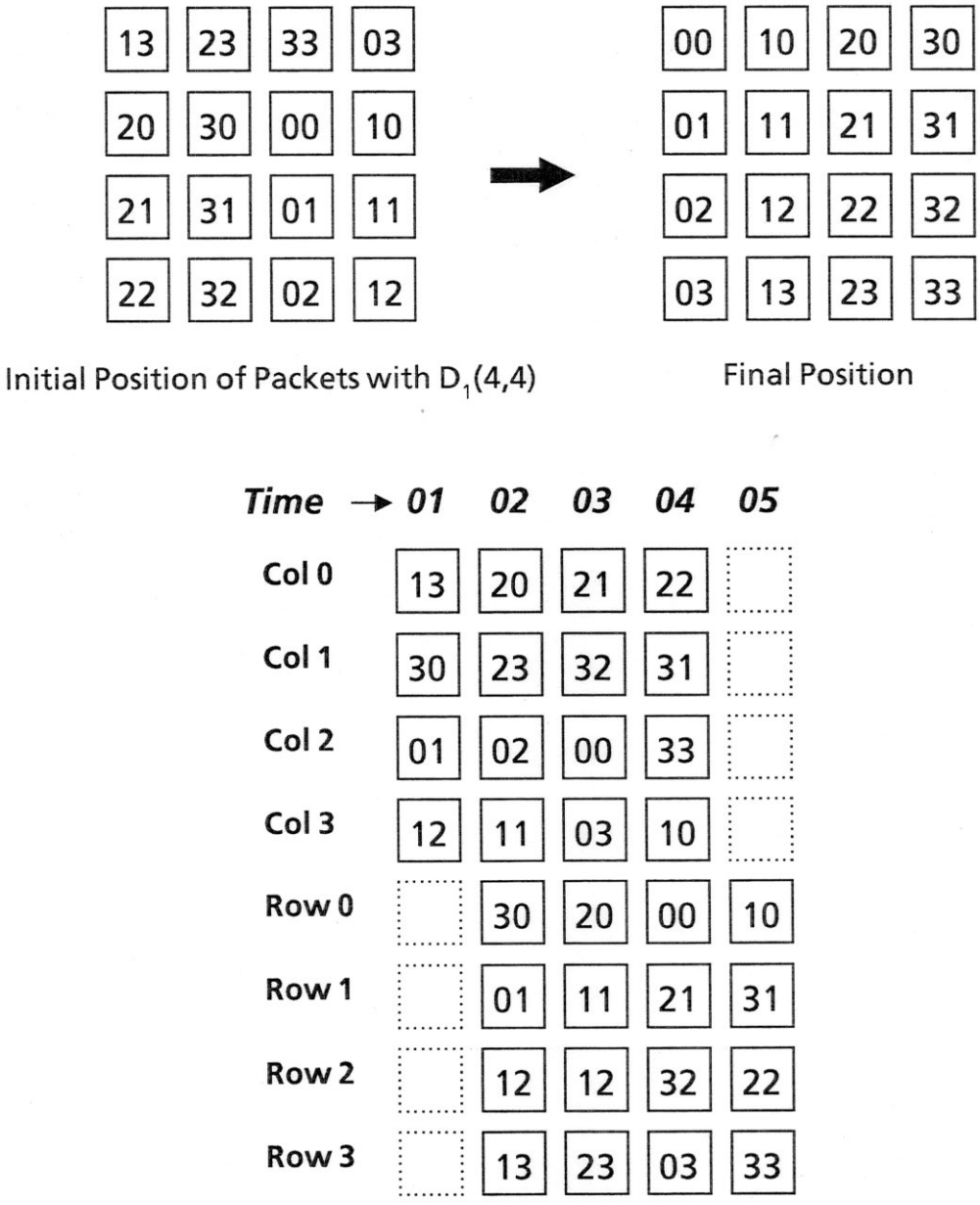

Figure 2.3: The uniform (column-row) schedule of $D_1(4,4)$