On considère ici un système de commutation par satellite utilisant l'accès multiple à répartition dans le temps (AMRT). Le satellite a plusieurs antennes qui couvert des zones différentes. Soit le nombre de répétiteurs à bord de satellite K, le nombre de liaisons Terre vers satellite N, et le nombre de liaisons satellite vers Terre M. Dans ce système, le satellite peut commuter des faiseux à son bord. Donc, on peut créer une liaison Terre vers satellite, par exemple, a liaison satellite vers Terre dans des endroits différents. Dans tel système, on doit considérer l'effet d'interférence parmi les liaisons dans le même sens. On propose un algorithme optimal qui établit l'horaire de connexion des créneaux de temps de chaque liaison. L'algorithme assure que tout le trafic peut être émis sans conflit, pendant la trame d'AMRT.

SUMMARY

In a typical satellite switched time division multiple access (SS/TDMA) system, the TDMA frame is divided into several time slots, and each slot has a switching configuration permitting a certain amount of traffic to be transmitted. The system main objective is to ensure that all the traffic can be transmitted without conflict within the TDMA frame. The satellite has a number of spot beam antennas covering geographical distributed zones and a solid state RF switch on board to provide connections between the various uplink and downlink beams. A high gain antenna on board the satellite provides various spot beam coverage. This antenna may allow interconnectivity between uplink and downlink beams, i.e., substantial power from one beam may spill over into a neighboring zone leading to what is called zone interference. The effect of zone interference has been studied in the special case of SS/TDMA system with N uplink and N downlink beams all of equal bandwidth and with at least N transponders on board.

In this work, we study the traffic scheduling problem with interfering beams in a more general SS/TDMA system. We consider a system with M uplink beams and N downlink beams, where uplink i has bandwidth $A$ and downbeam j has bandwidth $\alpha$. The maximum traffic which can be handled by the satellite (at any given time slot) is assumed to be $K \leq (M,N)$. The interference between zones can be characterized by the graphs $G_U$ and $G_D$ where $G_U = (V,E_U)$ represents the interference on uplink beams, with vertex set $V$ representing the M zones and edge set $E$ containing an edge $(v_i,v_j)$, $v_i , v_j \in V$ if and only if zone i interferes with zone j on uplink transmissions. Similarly, $G_D$ represents the interference on the downlink beams. The traffic demand is characterized by the traffic matrix $D$. The transmission schedule is defined as the decomposition of the traffic matrix $D$ into several switching matrices, i.e., $D = D_1 + D_2 + \ldots + D_m$, where $D_j$ represents the transmissions in time slot i of the TDMA frame. The sequence $(D_1, D_2, \ldots, D_m)$ is the transmission schedule. We look for a feasible transmission schedule which never requires two interfering zones to transmit or receive simultaneously. This can be achieved if no D matrix contains entries on rows corresponding to adjacent vertices of $G_D$ or on columns corresponding to adjacent vertices of $G_U$. The proposed algorithm provides the solution of this problem and it is optimal in the sense that it minimizes the transmission time for any given traffic matrix $D$. 
I- Introduction:

The satellite-switched time division multiple access (SS/TDMA) system utilizes the merits of high gain spot beam antennas along with the efficient TDMA method of providing complete connectivity of coverage areas. In such a system, a satellite embayles several spot-beam antennas and a solid-state RF switch periodically switches the connections of uplink and downlink beams to facilitate the connectivity of beam zones. A TDMA frame is divided into a number of switching modes, each switching mode is assigned a fixed on-board switch mechanism so that the traffic from various regions is routed to designated regions without conflict.

In an SS/TDMA network, all earth stations must share a given number of satellite transponders on a time base, it is desirable to maximize the transponder utilization according to the network traffic requirements. The time slot assignment problem is then to schedule the network burst traffic in a manner that maximizes the satellite transponder utilization with a minimum number of switching modes. More specifically, the traffic demand is characterized by a traffic matrix D, where entry $d_{ij}$ represents the traffic demand from uplink beam i to downlink beam j measured in units of slot time. A scheduling scheme then depicts a way to decompose the given traffic matrix D into distinct switching matrices i.e., D = $D_1 + D_2 + \ldots + D_k$, where $D_k$ is a matrix with its largest entry determines the duration of a switching matrix i.e. $[D_k]$. To maximize transponder utilization, one must minimize the total duration i.e., $\sum [D_k]$ needed to schedule (or switch) the complete traffic matrix D. On the other hand, the number of switching modes should be kept small. Let M and N denote the number of uplink and downlink beams respectively. The proposed algorithms deals with the real system where variable bandwidth of uplink and downlink beams, are considered, i.e., uplink beam i has bandwidth $\alpha_i$, and downlink beam j has bandwidth $\alpha_j$. In our case we may allow several entries in any row or any column as long as the sum of all entries of row i is less or equal to $\alpha_i$, and the sum of all entries of column j is less or equal to $\alpha_j$.

None of the algorithms discussed in (1) - (6) considers zone interference. However in practice side lobes in the antenna radiation patterns, insufficient antenna gain, or very small and closely packed zones may cause substantial power from one spot beam into a neighboring zone which is called zone interference. The proposed algorithms takes into account the presence of interference, i.e., the switching matrix $D_k$ must contain entries on rows corresponding to interference uplink zones or on columns which correspond to interfering downlink zones.

II- Notions And Definitions:

The uplink bandwidths are represented by the speed vector $\alpha = [\alpha_1 \alpha_2 \ldots \alpha_M]$, where $\alpha_i$ is a positive integer representing the bandwidth of the $i^{th}$ uplink, similarly, the speed vector $\beta = [\beta_1 \beta_2 \ldots \beta_N]$ represents the downlink bandwidths. The capacity of the satellite, K is the maximum amount the satellite can carry in one time slot. Assume $i \leq k < \min (\Xi, \alpha_j, \beta_j)$, and let $r_j$ and $c_j$ represent the $j^{th}$ row sum and the $j^{th}$ sum respectively. An $(\alpha \times \beta)$ matrix is defined as an $(M \times N)$ matrix with non-negative entries such that: $c_j \hat{\leq} \beta_j$ for $1 \leq j \leq N$ and, $r_j \hat{\leq} \alpha_i$ for $1 \leq i \leq M$. A quasi doubly stochastic (QDS) matrix $(\alpha \times \beta)$ is defined as an $(M \times N)$ matrix for which $c_j = S_i$ and $r_j = S_i$, $1 \leq i \leq N$ and $1 \leq j \leq M$, where S is a some positive integer called the line sum of the matrix. The normalized quasi doubly stochastic (NQDS) matrix is defined as an $(M \times N)$ matrix with $c_j \hat{\leq} \beta_j = S_i$ and $r_j \hat{\leq} \alpha_i = S_i$, $1 \leq i \leq N$ and $1 \leq j \leq M$ where S is a some positive integer called the normalized line sum of the matrix. The traffic of the matrix is defined as the sum of all entries in that traffic matrix. The interference between zones is characterized by the graphs $G_U$ and $G_D$. $G_U$ represents the interference on uplink beams with vertex set $V$ representing the $M$ uplink zones and edge set E containing an edge $(v_j, v_j)$, $v_j, v_j \in V$ and only if zone i interferes with zone j in uplink transmission. Similarly $G_D = (V, E)$ defines the interference on the downlink beams. $G_U$ and $G_D$ can be identical, but we consider the general case where they are different.

III- Problem Formulation:

The problem is: given a traffic matrix D $(M \times N)$ with uplink and downlink speed vector $\alpha$ and $\beta$, satellite capacity K, and uplink and downlink interference graphs $G_U$ and $G_D$, find the decomposition of D = $D_1 + D_2 + \ldots + D_k$ such that:

1) all the $D_k$’s are $\alpha \otimes \beta$’s with traffic not greater than K.
2) the number of switching matrices is minimized.
3) $D_k$ must not contain entries on rows which corresponds to adjacent vertices of $G_U$ or on columns which corresponds to adjacent vertices of $G_D$.

IV- Column and Row Expansion Algorithms:

For a feasible schedule D = $D_1 + D_2 + \ldots + D_k$ can never be smaller than S which is given by:

$$S = \max \left[ \sum_{i=1}^{M} T, \max_{1 \leq j \leq N} \sum_{i=1}^{M} \alpha_i, \max_{1 \leq j \leq N} \sum_{i=1}^{M} \beta_j \right]$$

where $T$ is the total traffic in $D$ and $[X]$ denotes the smallest integer greater or equal to X. From (8) and (9), an $(M \times N)$ NQDS matrix D with normalized line sum S can be expressed as sum of S-complete $\alpha \otimes \beta$’s.

The proof of this theorem is based on the constructions of an $(M \times H)$ QDS matrix from the $(M \times N)$ NQDS matrix of line sum. The construction proceeds as follows:

(a) Column expansion algorithm:

Expansion of the matrix D into $(M \times H)$ matrix $D'$ by expanding each column j in D into $(M \times \beta_j)$ matrix $A_j$ as follows:

Let $P(e)$ be the column sum of the column
in \( A_j \) and \( rk_b \) be the row sum of \( k \)th row in \( A_j \). Assuming first that all entries in \( A_j \) are zeroes, then we add the quantity \( x_{kj} \) to the \((k,1)\) entry in \( A_j \) where:

\[
x_{kj} = \min \left( S - cl_{g}, \frac{dk_j}{k} - rk_b \right)
\]

where \( dk_j \) represents the \((k,j)\) entry in \( D \). The process will be completed when no nonzero \( x_{kj} \) can be added.

The matrix \( A_j \) must have the same traffic as the \( i \)th column in \( D \) and all column sums of \( A_j \) is equal to \( S \), and the \( k \)th row sum of \( A_j \) is \( dk_j \). We construct such an \( A_j \) matrix for every \( j \), \( i \) \( \leq j \leq N \) so we will have \( A_1, A_2, \ldots, A_N \) which form a single \((N \times N)\) matrix \( D' \) which has the same traffic as \( D \) and has all column sums \( = S \), all row sums \( = \) row sums in \( D \).

(b) row expansion algorithm:

The second step is a similar expansion but this time on the rows of \( D' \) to obtain an \((N \times N)\) matrix \( D'' \). Row \( i \) of \( D'' \) is expanded into \((M+i)\) \((N+i)\) matrix \( B_j \). Let \( c_l \) be the sum of the \( l \)th column of \( B_j \), and \( r_k b \) be the sum of the \( k \)th row of \( B_j \). We add \( x_{ki} \) to the \((k,1)\) entry in \( B_j \) where:

\[
x_{ki} = \min \left( S - rk_b, c_l - cl_j \right)
\]

where \( c_l \) refers to the \((l,1)\) entry in \( D' \). The process terminates when no nonzero quantity can be added. The matrices \( B_j \)'s will form the \((N \times N)\) matrix \( D'' \) which has traffic equal to that of \( D' \) (which in turn is equal to that of \( D \)), and all rows and column sums of \( D' \) are equal to \( S \). Thus \( D' \) is a GDS matrix.

If \( D'' \) can be expressed as the \( S \) permutation matrices \( D_1'', D_2'', \ldots, D_S'' \), then we make compression to \( D_i'' \), \( 1 \leq i \leq S \), to get \( D_i' \) by first row compression algorithm followed by column compression algorithms for the \( D_i' \) to get \( D_i'' \). Thus each \( D_j \) is an \( oASM \).

Now we assume that the given \((M \times N)\) matrix is not a GDS matrix, thus we must transform it to \( D'' \) using the competition algorithm as follows:

\[ M = \sum_{i=1}^{N} a_i - K \]

\[ q_{i+1} = \sum_{j=1}^{M} b_j - K \]

for the extra row and assign

\[ q_{M+1} = \sum_{i=1}^{N} b_i - K \]

Thus for the matrix \( Q \), the vectors

\[ q' = \left( a_1, a_2, \ldots, q_{M+1} \right) \]

\[ b' = \left( a_1, a_2, \ldots, a_N \right) \]

Let \( r_{m+1} = \text{ith row sum of Q from left} \) and \( c_{n+1} = \text{jth column sum of Q from left} \). Initially let \( r_{M+1} = c_{N+1} = 0 \), and \( r_{M+1} = r_{i+1} + c_{j+1} + r_{i+j} \), where \( r_{i+j} \) and \( c_{i+j} \) refer to row sum and column sum of \( D \) respectively. Now, we add the quantity \( x_{i+j} \) to entry \( q_{i+j} \) for \( 1 \leq i \leq N \) where:

\[ x_{i+j} = \min \left( q_{i+1} - c_{i+j}, a_i - r_{i+j} \right) \]

The process terminates when no nonzero quantity can be added. Similarly we add entries to column \( N+1 \), to entry \( q_{i+N+1} \) for \( 1 \leq i \leq M \) and \( x_{i,N+1} \) where:

\[ x_{i,N+1} = \min \left( q_{N+1} - c_{i,N+1}, a_i - r_{i,N+1} \right) \]

We claim that, when this occurs, \( Q \) is an NQDS matrix.

* If \( Q \) is not a NQDS matrix, consider the submatrix \( Q' \) of \( Q \) defined by the first \( M \) rows and the first \( N \) columns. To entry \( q_{ij} \) in this submatrix add quantity \( x_{ij} \), where \( x_{ij} = \min \left( a_i - c_{ij}, a_j - r_{ij} \right) \). This quantity is called a dummy traffic which will be removed at the end of the algorithm.

The process terminates when no nonzero entry can be added, thus \( Q \) is a NQDS matrix.

V The Optimal Algorithm:

**Step (1)**
From the uplink graph \( G_U \) add the rows of the given traffic matrix \( D \) which has interference to each other. Calculate the new \( \alpha \) of the resulting row which will be equal to the sum of \( \alpha \)'s of these rows.

**Step (2)**
From the downlink graph \( G_D \) add the columns of the matrix (resulting from the previous step) which has interference to each others. Calculate the new \( \beta \) of the resulted column which will be equal to the sum of \( \beta \)'s of \( \beta \) this columns. The result is the \((N \times N)\) \( D' \) matrix.

**Step (3)**
If \( D' \) is not a NQDS matrix, we apply the competition algorithm to convert it to NQDS matrix w.r.t. \( S \). The new matrix is called the \( Q \) matrix.

**Step (4)**
Expand the matrix \( Q \) to \((M+1)\times(N+1)\) matrix using the column expansion algorithm.

**Step (5)**
Expand the resulted matrix to \((M+1)\times(N+1)\) matrix using the row expansion algorithm. \( D'' \) now is an NQDS matrix w.r.t. \( S \).

**Step (6)**
Using the Max -Min matching algorithm \((111)\), we can find the switching matrices corresponding to \( D'' \).

**Step (7)**
Convert the switching matrices resulted from the previous step to permutation matrices of entries \( 1 \)'s and \( 0 \)'s only.

**Step (8)**
Perform row compression algorithm to the permutation matrices resulted in step (7) by adding the entries of rows which have been expanded in row expansion algorithm in step (5).

**Step (9)**
Perform column compression algorithm to the matrices resulted from the previous step by adding the entries of columns which have been expanded in column expansion algorithm in step (4).

**Step (10)**
Remove the \((M+1)\)th row and \((N+1)\)th column from the switching matrices to obtain \((M \times N)\) switching matrices.

**Step (11)**
Subtract the dummy traffic, if it was added in the competition algorithm in step (3), from the switching matrices to obtain the switching matrices of the \((M \times N)\) \( D' \) matrix.
Step (12)
Obtain the corresponding feasible schedule for the original matrix D by associating on a one-to-one basis, each unit of traffic in D between the same uplink beams taking into account that any switching matrix must not contain on rows which correspond to interfering uplink zones or on columns which correspond to interfering downlink zones.

Example:
Consider SS/TDMA system with 4 uplinks, 3 downlinks, 4 transponder. The downlink beamwidth and uplink beamwidth are given by
\[ \beta = (2, 3, 2) \]
\[ \alpha = (2, 2, 2, 1) \]
\[ S = 5 \]
\[ K = 4 \]
and the uplink interference graph is
\[ G_U = 1 \quad 2 \quad 3 \]
\[ 4 \quad 2 \quad 3 \]
 interfere in the uplink transmission, similarly, the downlink interference graph is
\[ G_D = 1 \quad 2 \quad 3 \]

The traffic matrix D
\[
\begin{bmatrix}
2 & 5 & 3 \\
3 & 5 & 0 \\
5 & 0 & 0
\end{bmatrix}
\]

1- Add rows 2, 3 \( \alpha_2 = 2+2 \)
2- Add columns 2, 3 \( \beta_2 = 3+2 \)
3- \( H = \alpha \beta = 1 \times 1 = 7 \)
4- Permutation matrices
\[
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}

5- Row compression (1+2), (3+4+5+6), (7)
\[
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}

6- Column compression (1+2), (3+4+5+6, 7)
\[
\begin{bmatrix}
0 & 2 & 0 & 2 & 0 & 2 & 1 & 11 & 11 \\
2 & 2 & 2 & 2 & 2 & 1 & 13 & 13 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1
\end{bmatrix}
\]

7- The corresponding matrices of D are
\[
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

Type1 Type2 Type3 Type4 Type5

There are 3 matrices of the first type, 3 of the second, 2 of the third, 2 of the forth, and 2 of the fifth one. The total duration is 12

References