COMPARISON BETWEEN DISTRIBUTED AND CENTRALISED DEMAND ASSIGNMENT TDMA SATELLITE ACCESS SCHEMES

Nedo Celandroni, Erina Ferro, Francesco Potortì
CNUCE, Institute of National Research Council
Via S. Maria 36 - 56126 Pisa - Italy
Fax: +39-50-904052 / 589354
Phone: +39-50-593111
E-mail: n.celandroni@cnuce.cnr.it
e.ferro@cnuce.cnr.it
pot@cnuce.cnr.it

SUMMARY

This paper presents the performance measurements from a comparison between FODA/IBEA and two other satellite access schemes, FEEDERS and DRIFS. The three schemes differ in terms of the scheduling of the channel capacity: i.e. centralised control in FODA/IBEA and distributed in the other two. All these access schemes were designed at CNUCE, where the simulation tool used for the comparison was also developed. FODA/IBEA was developed and tested on several satellites (Olympus, Eutelsat and Italsat), and used in a project of LAN interconnection[(1)]. The two distributed control access schemes have been studied and simulated to analyse the behaviour of the capacity assignment algorithm, while the recovery procedures needed for the network stability are presented in [6] and [7]. All the access schemes support an *aggregated* traffic, and the quality of the data transmission is guaranteed even when the transmitting signal experiences a severe attenuation due to bad atmospheric conditions.

Keywords: satellite, TDMA access schemes, distributed control, centralised control, comparison, aggregated traffic, traffic generatorts, fade countermeasure.

1. INTRODUCTION

(1) "Thin route TDMA for LAN interconnection" project

The FODA/IBEA[(2)] access scheme [1] was developed on a prototype of hardware manufactured by Marconi R.C. (U.K.), and tested on three different satellites (Olympus, Eutelsat, Italsat) in order to evaluate its performance. References [2, 3] present the results of the tests on the Italsat satellite, obtained with four Italian active stations. These measures refer to the packet arrival time jitter, which affects the real-time data, and to the end-to-end delay of the non real-time data.

DRIFS[(3)] and FEEDERS[(4)] derive from FODA/IBEA. The goal of the study was to transform FODA/IBEA from a centralised into a distributed control scheme, to save one round trip time (RTT) between the times of the request and the assignment of the capacity. In fact, in a centralised control system, a master station is charged with computing the bandwidth allocation layout, which is broadcast to the traffic stations after their requests have been received. This causes a request-allocation delay of at least two round trips. On the other hand, although centralised control is more expensive in terms of end-to-end delay, it is more robust than distributed control, and its simpler implementation and greater robustness may compensate for the longer allocation delay.

The transformation had to be designed to maintain the main features of the FODA/IBEA scheme. The first one is the ability to support aggregated traffic, which is the characteristic of LANs interconnection traffic. It incorporates real-time traffic (telephony and video) and non real-time traffic (computer data exchange). The real-time traffic is often referred to as *stream,* and includes data from both fixed and variable rate applications. The non-real-time traffic consists typically of *bursty* traffic, classified in two categories: bulk (file transfer) and interactive (TELNET, RLOGIN). When FODA/IBEA was used in an experiment of LAN interconnection [4], the stream traffic arrived at the satellite network from a Token Ring, while the bursty traffic was generated on Ethernet.

The second feature is the capacity to maintain the quality of service requested by the various applications in any weather conditions, even when the transmitting signal is faded. The fade

---

(2) Fifo Ordered Demand Assignment/Information Bit Energy Adapter

(3) Distributed allocation with Request In Fixed Slots

(4) Faded Environments Effective Distributed Engineering Redundant Signalling

countermeasure technique adopted adapts the energy per information bit to each individual link status, which depends on atmospheric conditions. The total attenuation of each link (up-link plus down-link) is compensated for by varying the transmitting power, the data coding and bit rates. Assuming a multi-channel TDMA access to the satellite, the transmitting power variation ensures a suitable constant back-off at the satellite transponder input, to avoid excessive intermodulation noise. The power control can thus be used to compensate for all or part of the up-link attenuation, while the total compensation is completed by varying the data coding and bit rates as well. Data is made redundant according to the level of the fade detected. The redundancy is obtained by increasing the data coding rate first(5), then by reducing the data bit rate(6) as well, if the data coding redundancy is insufficient to counter the fade. This countermeasure technique was developed in the FODA/IBEA system by exploiting a very powerful feature of the modem prototype, which can dynamically change the data bit rates on a sub-burst basis. The *sub-burst* is the sequence of data addressed to a certain station within a *data burst*, which, in turn, is the total amount of data transmitted by a station during the time slot (*transmission window*) assigned to it. In other words, data is "more expensive" in terms of transmission time duration according to the weather conditions at the sending and at the receiving stations, because the redundancy introduced lengthens the burst duration. The price of this feature is the need for a short preamble between sub-bursts, in addition to the fairly long preamble needed before each burst to allow carrier and bit timing synchronisation. Unfortunately, all these preambles considerably reduce system efficiency.

Although all the three access schemes are in principle independent of the hardware used, if a preambleless modem were available, this would have a heavy impact on some design choices. As the goal of our study was to migrate from FODA/IBEA with centralised control to a system with distributed control, in the new schemes we will continue to refer to a modem which uses preambles for acquisition. However, all three access schemes proposed would obviously benefit from a more efficient preambleless modem.

The aim of this paper is to compare the three access schemes in order to highlight under which conditions an access scheme performs better than another one. Section 2 overviews the three

(5) Available values are: 1/2, 2/3, 4/5 and uncoded.
(6) Available values are: 1, 2, 4 and 8 Mbit/s.

access schemes, addressing the reader to the literature where each access scheme is detailed. Some other access schemes are referenced for comparison in [8÷13]. Section 3 describes the traffic generators used for the simulation and in Section 4 the comparison results are presented and discussed. Finally, conclusions are reported in Section 5.

2. DESCRIPTION OF THE ACCESS SCHEMES

In a system designed for aggregated traffic, such as FODA/IBEA, the request-allocation delay (RAD) of at least two RTTs has an impact both on the channel set-up and release times of the stream traffic, as well as on the end-to-end delay experienced by the packets during the transient due to a step of bursty traffic. Furthermore, since in the FODA/IBEA system each station adapts the coding and the bit rates of the transmitted data to the link quality, the attenuation level of each link must be estimated and predicted for a time which depends on the RAD. The accuracy of such a prediction very much depends on the prediction time, so the higher this quantity is, the higher the margin is on the link budget to be adopted in order to prevent an under-estimation of the link attenuation.

In order to reduce the RAD value, we have studied two distributed control systems for the capacity allocation. To achieve acceptable system stability in the distributed schemes, sophisticated recovery procedures were studied, taking into account the burst miss probabilities, the request miss probabilities, and the non-transmission probabilities. For the sake of brevity, we only give a short description of how each scheme utilises the channel.

2.1 FODA/IBEA

Some concepts are hereafter reported in relation to their use in FODA/IBEA, but where necessary we will outline how DRIFS and FEEDERS make use of these concepts too.

• The frame is the interval of time between two consecutive reference bursts which are sent by the master station for synchronisation purposes other than broadcasting the burst time plan to all the traffic stations. The frame length is fixed at 20 ms.

• The burst time plan (BTP) contains the transmission windows layout of all the traffic stations. In each transmission window both stream and bursty data are transmitted, thus only one preamble is needed for each transmitting station. The BTP must be known by each station, so as to transmit, as well as to receive all the incoming bursts. This constraint is imposed by the hardware available at present, in which the modem must be quenched immediately before the next burst arrives, in order to avoid false acquisitions of the carrier frequency. The incoming burst bit rate must also be known in advance by the modem. The access schemes are designed considering these limits. Hardware without these limits would simplify some procedures.

• Stream is a connection-oriented type of traffic. Only variations in the size of the allocation need to be specified in the allocation request. A request for fixed bit rate (FBR) or for variable bit rate (VBR) stream data is only sent at the opening of the stream session. If accepted, the allocation is kept until it is explicitly relinquished or the requesting station is declared dead.

• Bursty transmission is connectionless, and an allocation is granted only if requested. Bursty requests are sent as frequently as possible. The request (r) is computed as

$$r = q + H\,i \tag{1}$$

where q is the station backlog, i.e. the volume of data already waiting for transmission to the satellite, i is the amount of traffic coming into the station, and H is a temporal constant. The traffic stations issue their bursty requests as frequently as possible, in order to give the master an up-to-date situation of their input traffic. The requests are mostly sent piggy-backed with data or by using the control slot, a portion of the frame assigned, on a round robin basis, to the stations which have no assignment for that frame. The control slot contains the allocation requests plus some extra information, such as the up- and down-link attenuations. In FODA/IBEA one control slot is scheduled for each group of eight stations. If all the stations get an assignment, the space devoted to the control slot(s) is added to any unused space in the frame. The master organises the bursty requests of all the slaves into a *ring,* which it scans cyclically to compute the assignments. The length of the bursty assignment (a) is proportional to the request (r) in a range of values between a minimum ($T_{\min}$) and a maximum ($T_{\max}$) threshold. We have:

$$a = \min(T_{\max}, \max(T_{\min}, fr)) \qquad (2)$$

where f is the coefficient of proportionality in the assignment, set equal to the number of active stations N divided by 100, with 5% as minimum and 50% maximum. $T_{\min}$ was introduced for efficiency purposes, to avoid too small allocations when the transmission overheads are too big with respect to the information data. $T_{\max}$ prevents an overloaded station from removing too much capacity from the other stations. After each bursty assignment, the relevant request in the ring is decreased by the assignment itself and the next request is analysed, if space is still available in the frame. The ring is not scanned more than once in a frame. We call *assignment cycle* a complete scan of the ring. Thus, no more than one assignment cycle is made in a frame. After an assignment cycle, any unused space in the frame is redistributed among the stations [2].

The control slot and the bursty space assignment algorithms were studied to allow a quick response time to traffic spikes at the station input. When the system is lightly loaded, its behaviour approaches that of a fixed TDMA. This situation is called *pre-assignment* mode. The system gradually migrates towards a “pure” FODA/IBEA assignment scheme when the channel load exceeds a threshold. A moderately loaded system can absorb abrupt traffic variations without appreciable delays, because each traffic station has some spare capacity.

In pre-assignment mode the assignment cycle is always one frame long, and some spare capacity is always available to be shared among all the stations. When the assignment cycle exceeds one frame, capacity is no longer available to be shared.

• In unfaded conditions, the stream traffic cannot go over a fixed limit in the frame, while the remaining portion of the frame is occupied by the bursty traffic. On the other hand, bursty traffic can temporarily expand in the frame if the amount of stream traffic is insufficient to reach the boundary. The stream traffic can only go beyond this boundary in faded conditions. In this case, an effort is made to support the already set-up stream sessions whose data needs to be made redundant to counter the fade. In this situation the bursty traffic may momentarily be suspended and the whole frame devoted to stream. No new stream requests are accepted during the boundary overflow time.

• Fading conditions cause an increase in the backlog and in the instantaneous traffic of the faded station and of the stations transmitting to it, because the same information needs more channel space to be transmitted with the same BER. This automatically increases the bursty request. Due to this increase and to the possible simultaneous expansion of the stream data, the overall capacity of the system devoted to the bursty traffic may become significantly smaller under heavy load conditions. Such a squeezing of the bursty capacity generally requires a congestion control scheme. FODA/IBEA incorporates a simple back-pressure method to relieve congestion, by blocking the growth of the backlog for a while when the internal queue length is such that the queueing time estimated goes beyond a threshold. The effect of this procedure is to slow down the bursty traffic coming from the remote hosts for a convenient interval of time.

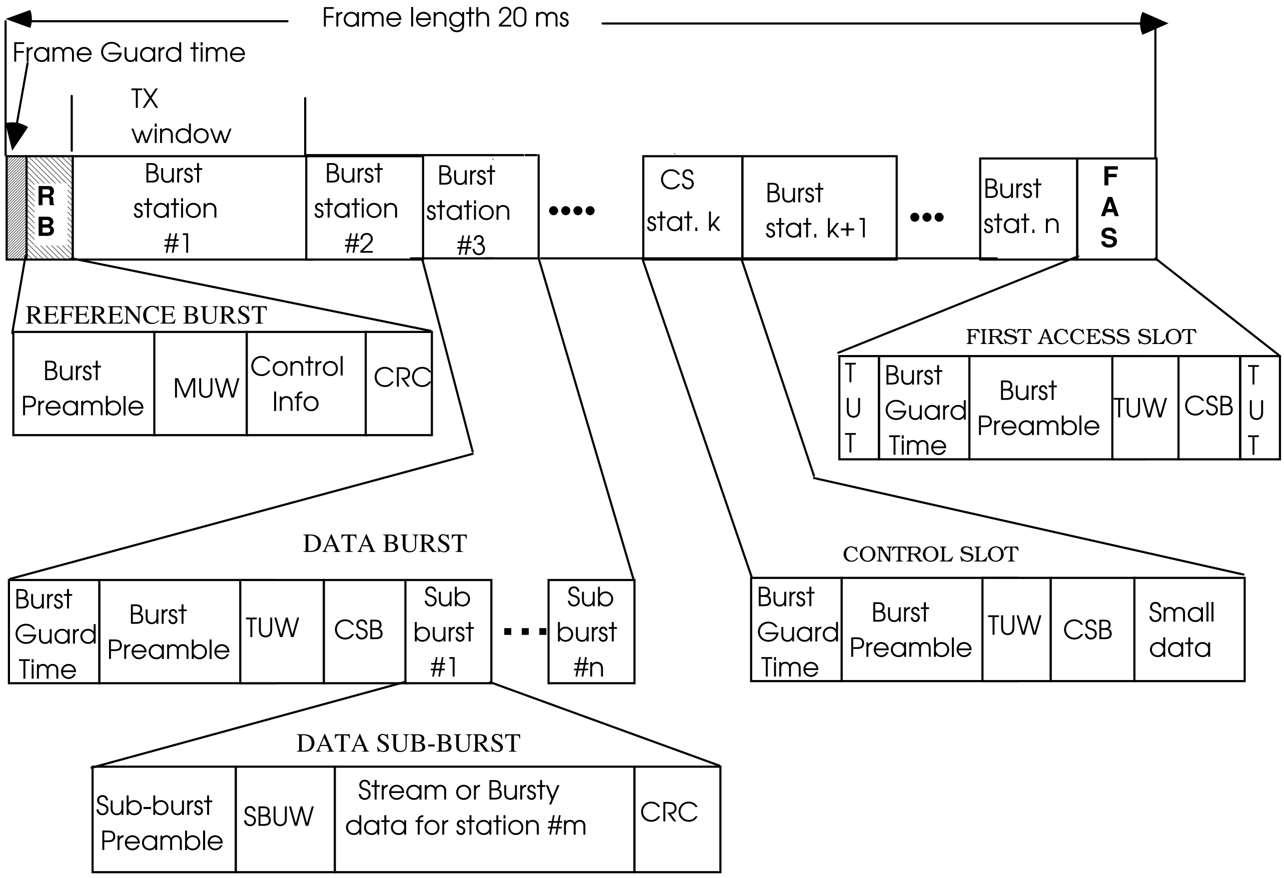
• A new station is given the opportunity to enter the satellite network by using the FAS[(7)] space in the frame. FAS has a fixed position, before the end of the frame and its frequency is every 32 frames. As it is accessed in contention mode, if there is a collision with another station entering the system, the colliding stations wait for a random number of frames before repeating the operation. When the maximum number of active stations is reached, the FAS space is temporarily deallocated because, during this period, no other station is allowed to enter the system. The FAS is allocated again when at least one more entry is possible in the system.

• Due to the TDMA controller hardware considered, each data burst must be preceded by a control sub-burst (CSB) which carries information relevant to each sub-burst in the burst, such as the data destination, length, coding and bit rate, bursty and/or stream requests, estimated fade levels, etc.

---

(7) First Access Slot

The format of the FODA/IBEA frame is shown in Fig. 1. The burst and sub-burst structures are valid for all the access schemes presented.



<u>Legend</u>

RB	Reference Burst
CSB	Control Sub-Burst
CS	Control Slot
FAS	First Access Slot
Burst Preamble	For modem acquisition.
Sub-burst preamble	For modem and decoder synchronisation.
MUW	Master Unique Word. Only relevant to the reference burst.
TUW	Traffic Unique word. Relevant to each data burst.
TUT	Time Uncertainty Tolerance
SBUW	Sub-burst Unique Word. Half the length of the MUW or TUW.
Control info	This part of the RB contains the BTP, the fade level measured by each station, the frame number, etc.
CRC	Cyclic Redundancy Check.

Fig. 1. Frame structure in the FODA/IBEA scheme

2.2. DRIFS

Figure 2 shows the format of the frame in the DRIFS scheme.

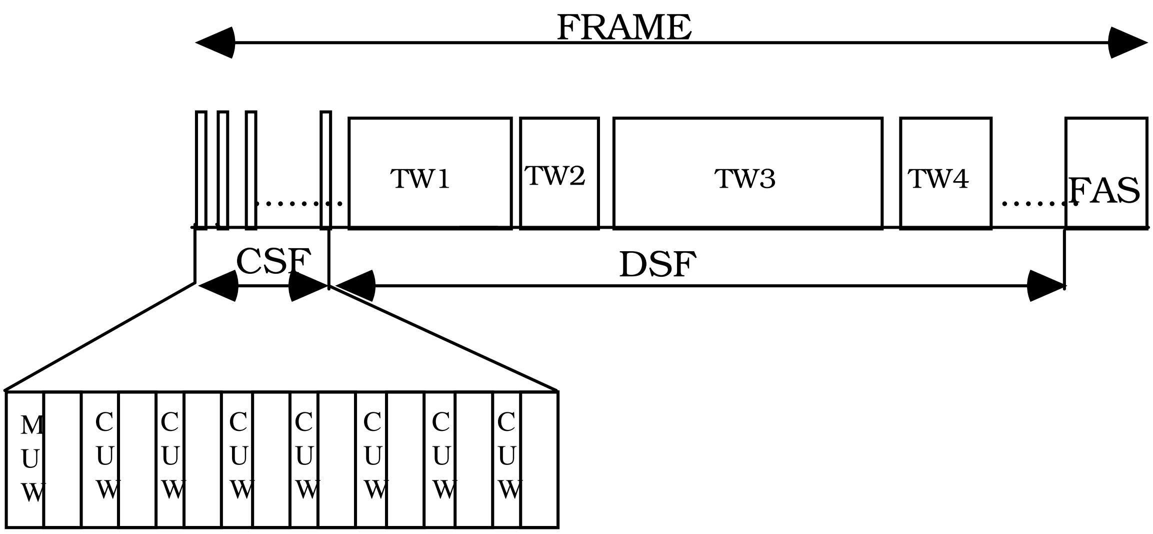


Fig. 2. Frame structure in the DRIFS scheme. TW = Transmit Window

Each frame consists of a *control sub-frame* (CSF), used in Fixed-TDMA, followed by a *data sub-frame* (DSF), used in Demand Assignment-TDMA, and finally by an FAS. No reference burst is supported.

The control sub-frame is used for information such as allocation requests and fade levels, while the application data is sent in the data sub-frame. The first access slot is used by those stations willing to enter the system, exactly as described in FODA/IBEA. The only difference is that, if the maximum number of stations has not been reached, its frequency is every frame.

Each active station has its own control slot in the CSF, whose size is thus proportional to the number of active stations. The position of the control slot of a station is fixed; it is shifted back only when a preceding station leaves the system. The first control slot is preceded by the MUW, to allow system frame synchronisation. The other control slots in the CSF are preceded by the CUW (Control Unique Word). The data bursts sent in the transmit windows are preceded by the TUW (Traffic Unique word), as in FODA/IBEA and in FEEDERS.

Each active station with an allocation for transmitting data sends one burst in the data sub-frame. The data burst and sub-burst format is shown in Fig. 1.

Since the CSF length is not negligible with respect to the frame length (at least using traditional modems that need a preamble), no more than U control slots are accommodated in a frame. If more than U active stations are present in the system, the control slots are spread in $C_c =]N/U[$ frames, where N is the number of stations and $]x[$ is the smallest integer not smaller than x. The control data must be strongly protected against either loss of acquisition or transmission errors. Assuming we use the same modem that we have at present, we devise a bit rate of 2 Mbit/s and a coding rate of 4/5.

The request and allocation algorithms are the same as in FODA/IBEA. The difference is in the assignment cycle which in FODA/IBEA can be any length, while in DRIFS it is always equal to C_c frames.

The critical points of these algorithms are: i) the correct reception of the data transmitted in the control slots; ii) the channel overhead due to the CSF.

As far as point i) is concerned, a station must be able to distinguish a control slot reception error caused by a noisy channel from one caused by an empty channel in order to make the right scheduling adjustments. In the first case, a *CRC error* is detected when the data is received; in the second case a *CUW miss* is signalled. In both cases, the station cannot compute the BTP and, in order to avoid possible collisions, the station does not transmit data in the DSF for the next assignment cycle.

Regarding point ii), with the hardware currently at our disposal one control slot occupancy is equal to 2.4% of the frame, considering the long preamble which is sent at 2 Mbit/s (four times slower than the maximum bit rate of 8 Mbit/s allowed by the hardware considered). If we assume that no more than eight control slots are set up in each frame, then the CSF produces an overhead of about 20% of the available bandwidth.

In [6] details on the DRIFS scheme can be found, together with the estimation of the wrong reception and non transmission probabilities.

2.3 FEEDERS

Figure 3 shows the frame format for this access scheme.

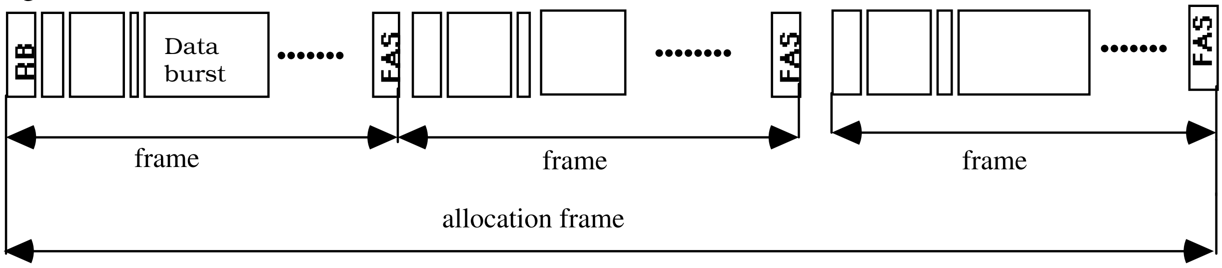


Fig. 3. Frame structure in the FEEDERS scheme

In FEEDERS the concept of a master station still exists, not as used in the centralised control scheme but only as a reference in case of errors and for entering the system. The frame is the interval of time in which all the active stations transmit a data burst which contains control information which may or may not be followed by application data. The format of any data burst is the same as depicted in Fig. 1. The reference burst is sent every n_a frames with an MUW, allowing all the traffic stations to synchronise with the network. Within the reference burst a reference BTP is also sent to allow any new station wanting to enter the system to set up the BTP for receiving. The reference BTP is also needed in the algorithm to recover missed allocation requests. The *allocation frame* (AF) is the time between two consecutive BTP applications. The AF coincides with the time between two consecutive reference bursts sent by the reference station. The AF duration time is denoted by $t_a = n_a t_f$, where n_a is the number of frames contained in an AF (see Figure 3), and t_f is the time frame length. In each AF the stations have n_a chances to transmit, so each individual control message (and consequently each allocation request) is repeated n_a times. This redundancy dramatically reduces the probability that none of the stations understands the control information. The BTP is applied to all the frames in an AF.

The choice of the value of the n_a parameter is strictly related to the control message miss probability that the system wants to support, and, consequently, to the number of active stations, as explained in [7]. In the simulations, whose results are reported in Section 4, a value of n_a equal to 4 was chosen.

The allocation request for stream capacity is computed similarly as in FODA/IBEA.

The allocation request for bursty traffic is computed by each station as:

$$r = \min [f\ (q + H\ i),\ r_{\max}] \qquad (3)$$

where f (less than one) and H are the coefficients to be chosen. The request does not have a lower limit, unlike in FODA/IBEA, because there is already one burst per station, so there is no need to allocate a minimum amount for efficiency reasons.

Both the allocations are computed in a distributed way. They can vary on an AF basis. Each station must be able to compute the BTP for the next AF at the end of the current one. Usually, the BTP is computable after the first frame in each AF. However, if bursts are missed, data needs to be collected in the subsequent frames, in order to build a complete request plan with the allocation requests from all the stations.

The allocation algorithm is similar to the one adopted in FODA/IBEA. The stream bandwidth is allocated equal to the request and maintained until an explicit relinquish indication is sent.

The bursty allocation is computed as follows. For each station an allocation equal to the request, increased by the preamble overhead, is allocated beforehand in every frame. If a residual capacity C_r still remains, after a complete allocation cycle, it is shared evenly among all the active stations, even those ones which sent a null request. If C_r is negative, all the requests are reduced by a suitable factor, to make the allocations occupy exactly the capacity C_d reserved for datagram. In this case the stations which sent a null request receive an allocation which is only enough to send one control message per frame. No control slot is used.

In FEEDERS, like in FODA/IBEA, the bit and coding rates of the reference burst are fixed to 2 Mbit/s and 4/5, respectively, as a compromise between the frame overhead and the fade range to support. We did not choose the maximum protection allowed by the system, i.e. 1 Mbit/s and 1/2, because we assume that the master is one of the least faded stations. In this case, in fact, the up-link attenuation is completely compensated for by the up-power control system of the master and the down-link attenuation only of the station that is in the worst conditions must be compensated for by bit and coding rate reduction. In any station, except the master, bit and coding rates must also compensate for any up-link attenuation portion that remains uncompensated after the up-power control. When the master is no longer able to compensate for its up-link attenuation, the role of master is assumed by another station in good conditions. The maximum protection allowed by the system is reserved for the stations that are deeply faded. In FEEDERS, the preamble bit rate of the other bursts is the same for all the stations. It is selected dynamically as the highest one which guarantees an E_b / N_0 ratio better than a minimum threshold for all the stations.

Details on FEEDERS can be found in [7], together with the average system efficiency and the probability that a station misunderstands the control information of all the others and provokes collisions.

2.4 Features Summary

Table 1 summarises the respective features of FODA/IBEA, DRIFS and FEEDERS, and highlights the similarities and the differences.

	FODA/IBEA	**DRIFS**	**FEEDERS**
Time frame length	20 ms	as in FODA/IBEA	as in FODA/IBEA
Hardware features	Marconi hardware	as in FODA/IBEA	as in FODA/IBEA
traffic supported	stream and bursty traffic	as in FODA/IBEA	as in FODA/IBEA
burst, FAS & CS formats	as depicted in Fig. 1	as in FODA/IBEA	as in FODA/IBEA
RB presence	Yes, 1 per frame	NO	Yes, 1 every n_a frames
RB purpose	synchronisation and to broadcast the BTP	___________	synchronisation and reference in case of errors
control slot (CS) usage	1 every 8 stations, assigned on a round robin basis to the station with no assignment in that frame. If none, added to any unused space in the frame	8 per frame	No
first access slot (FAS) position & frequency	before the end of the frame. 1 every 32 frames until the maximum number of active stations is reached	before the end of the frame. 1 per frame until the maximum number of active stations is reached.	as in DRIFS
stream request/relinquish	sent at the stream session opening/closing	as in FODA/IBEA	as in FODA/IBEA
stream allocation	equal to the request, if accepted	as in FODA/IBEA	as in FODA/IBEA
bursty request	r = q + H i	as in FODA/IBEA	$r=\min[f\ (q+Hi), r_{max}]$
bursty allocation	$\min(T_{max},\max(T_{min},fr))$	as in FODA/IBEA	a=kr, where k depends on the system overall load
allocation cycle (AC) (a complete scan of the bursty request ring)	Any length but no more than 1 AC per frame	Always equal to C_c=]N/U[frames	equal to the frame length

fade countermeasure supported	YES	YES	YES
fade countermeasure technique adopted	up-power control plus data coding and bit rate reduction	as in FODA/IBEA	as in FODA/IBEA

TABLE 1. Features of FODA/IBEA, DRIFS and FEEDERS

3. THE TRAFFIC GENERATORS

Our aim is to compare the three methods in terms of the bursty data capacity assignment, since the stream capacity allocation does not present particular problems and the algorithm is the same for all three methods.

Many simulation runs were thus made with three different traffic generators: Poisson, Two-states Markov-modulated Poisson and Fractional Gaussian Noise. Recently [14] there seems to be evidence about the self similar nature of LAN traffic, whose behaviour is not captured by any of the commonly used traffic generators, such as Poisson or Markov-modulated Poisson. Even the argument, once generally accepted, that the aggregate LAN traffic becomes less bursty as the number of traffic sources increases seems quite far from reality. In fact, measures seem to show that the burstiness of LAN traffic does not decrease as the number of active traffic sources increases (the self-similarity discussed in [14]).

In order to highlight different aspects of the protocols' behaviour we used three different types of traffic generators and compared the results. The behaviour of the protocols simulated is quite different under the different types of load, thus allowing a wide range comparison.

Poisson traffic is simple and has attractive theoretical properties, which is why it was widely used until recently. Interarrival times in Poisson traffic have negative exponential distribution and are independent, so the number of packets in any time interval follows a Poisson distribution. The only parameter needed for characterising Poisson traffic is its mean throughput. For wide area traffic, Poisson traffic models well the arrival of user initiated sessions [15], but it is not good for capturing the features of the data traffic within sessions. However, Poisson traffic was used to load the FODA/IBEA system for measurement purposes on the Italsat satellite, so we used a Poisson load to compare the simulation results with the measurements on the satellite. Poisson is a particularly well-behaved traffic, from a burstiness point of view. It also does not exhibit long range dependence, as its autocorrelation function decreases exponentially. It can be considered as a sort of best case loading, and indeed our measures show that the protocols manage to make the best use of the channel under this kind of load.

The Two-state Markov-modulated Poisson traffic we used still does not exhibit long range dependence, but it does exhibit high burstiness on average. It is made of two independent Poisson generators with different mean throughputs. A two-state Markov chain is used to choose which of the two generators is active at any given time; the two states are called the *high traffic* state and the *low traffic* state. This generator is defined by four parameters. We chose mean throughput, mean time of permanence in high and low traffic states, and ratio of mean throughputs in the two states. The high traffic state lasts for a mean of 0.5 seconds and the low traffic state for a mean of 2.5 seconds. The traffic generation mean rates are in a 17 to 1 ratio, resulting in a burstiness (peak to mean value ratio) of about 5. High burstiness is observed in LAN-to-LAN communications, as can be seen in the Bellcore traces [17], shown for example in [18]. The parameters of this generator, equal for all the stations, were chosen in order to represent a worst-case load for satellite protocols from a burstiness point of view, yet to prevent any single station from generating a peak traffic greater than half the capacity of the channel. Indeed, the length of the bursts are chosen so as to maintain the station request algorithm in a “continuously transient” state, where the input traffic jumps to a different value as soon as the request of the station and its assignment have stabilised. As soon as each burst begins, the station has an insufficient bandwidth allocation, so the input packets are queued and their delay increases. When the burst ends, the station has an allocation greater than is necessary, thus lowering the overall efficiency of the channel. These effects lead to a very low channel utilisation compared with the other kinds of traffic we have used, as our measures show.

The fractal generator is an approximated Fractional Gaussian Noise generator implemented with a simple Random Midpoint Displacement algorithm. In [16] there is an initial analysis of the statistical properties of this algorithm, with results that we considered to be good enough for our needs. This generator exhibits relatively low burstiness and a long-term correlation, which we truncated to about 10 minutes of simulation time (about 30000 frames); so our input traffic traces are made of consecutive 10 minutes long independent batches. Fractional Gaussian noise is defined by three parameters, namely mean, peakedness and Hurst parameter. We used a Hurst parameter equal to 0.85, in line with the findings published in [14]. The peakedness, defined as the ratio between the variance and the traffic distribution mean values, was set equal for all stations. Its value is 1/10 of the mean total channel load of each simulation run: this proportionality is aimed at obtaining the same form factor (standard deviation / mean value) for the total load in all the simulation runs. Based on the measurements reported in [17], many authors believe that fractional Gaussian noise is good for modelling the aggregate output of a great number of sources whose traffics exhibit long range dependency ([14], [16], [18]). If the satellite network is used as a bridge through many LANs with high traffic between them, this modelling is probably a good fit for the real traffic.

4. COMPARISON RESULTS

In the legends of some of the following figures -P stands for Poisson traffic, -I for Impulsive, or Two-states Markov-modulated Poisson, and -F for Fractal traffic. Moreover, the "single station load [%]" and the "channel load [%]" are expressed as percentages of the total channel capacity, i.e. 8 Mbit/s. The end-to-end delay reported in all the figures is the total delay experienced by a packet in the satellite network crossing. It is the sum of the queuing delay at the sending station, the round trip time and the CPU processing time. This last quantity was introduced by the simulator only in the case represented in Fig. 4, where a comparison with the real case is made.

All the tests presented were averaged over 30,000 frames and refer to the case in which the whole channel capacity is devoted to the bursty data.

Figure 4 shows the reliability of the simulation tool used. The mean end-to-end delay, averaged over 30 s, is shown as a function of each of the four stations' loads. Each quasi-horizontal line represents a run characterised by an overall channel load level, while each quasi-vertical line represents the delay of one of the four stations. The results obtained by simulating

FODA/IBEA agree sufficiently with the ones obtained with the real tests on the Italsat satellite. In this case, the simulation tool was adjusted to introduce a minimum delay of 292 ms, considering the pure Round Trip Time (252 ms) plus two frames. This is because an extra frame is introduced by the present system implementation, due to insufficient CPU power in the satellite controllers. This additional frame could be easily saved using a more powerful CPU board, which is now on the market.

In the DRIFS scheme, the number of control slots per frame, U, in the CSF is kept constant and equal to 8, whatever the number of stations is. In the simulation of the FEEDERS method a value of n_a equal to 4 has been chosen.

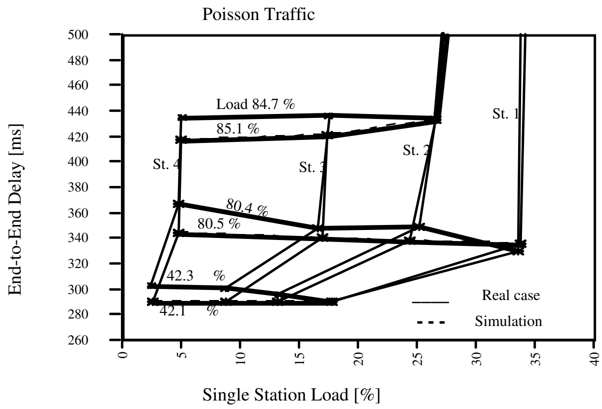


Fig. 4. Comparison between the FODA/IBEA real and simulated cases.
Mean end-to-end delay versus single station load.
4 stations. Poisson Traffic. 30s run

In Figs. 5÷8 the mean end-to-end delay of the overall channel is reported as a function of the channel load for the three traffic types, and for 4, 12, 32 and 48 stations, respectively.

It can be seen that for a low number of stations (cases 4 and 12) FEEDERS performs much better than the other schemes. This is principally due to the lower overhead of this method. The distributed methods generally perform better than the centralised one, as the traffic burstiness increases for all the loading levels. With 32 stations FEEDERS is penalised by the higher number of preambles per frame which considerably reduce channel efficiency. This method performs still better than FODA/IBEA for impulsive traffic, but is the worst in the other two traffic cases. DRIFS is the best with 32 stations, for high loads, while it is penalised by having a fixed allocation cycle (4 frames) at low-medium loads.

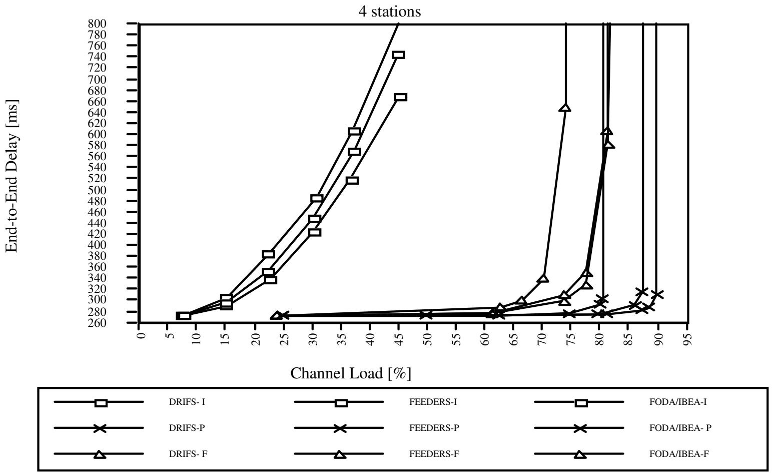


Fig. 5. Comparison between FODA/IBEA, DRIFS and FEEDERS for Fractal, Poisson and Impulsive traffic. Mean end-to-end delay versus single station load. 4 stations.

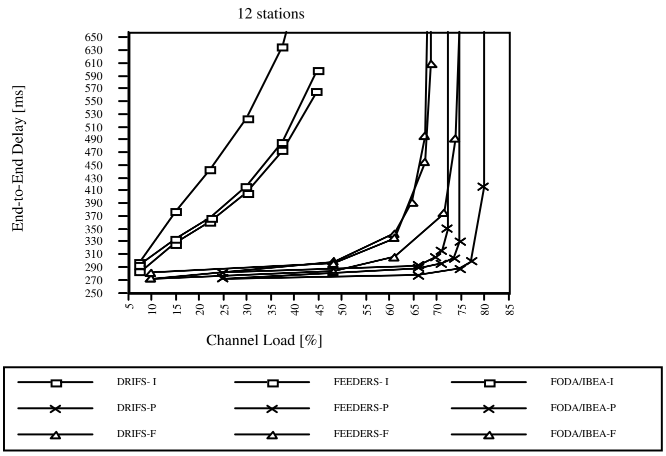


Fig. 6. Comparison between FODA/IBEA, DRIFS and FEEDERS for Fractal, Poisson and Impulsive traffic. Mean end-to-end delay versus single station load. 12 stations.

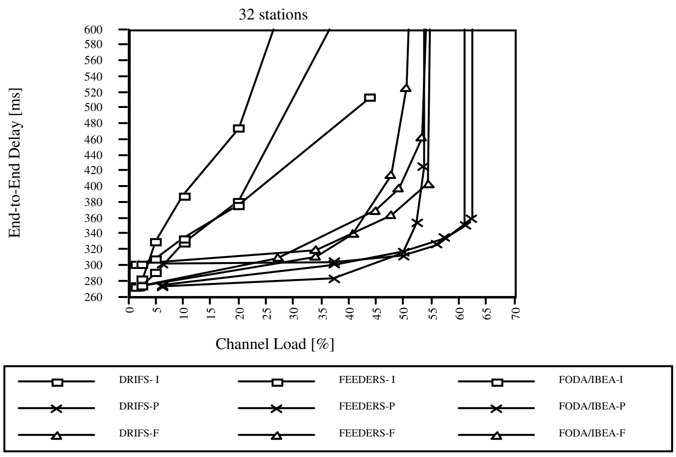


Fig. 7. Comparison between FODA/IBEA, DRIFS and FEEDERS for Fractal, Poisson and Impulsive traffic. Mean end-to-end delay versus single station load. 32 stations.

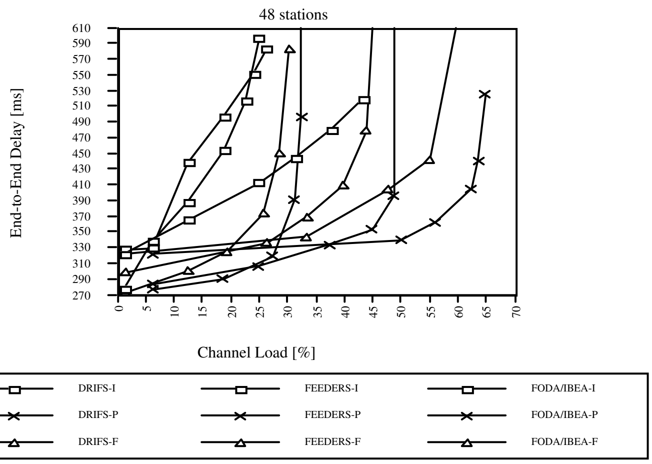


Fig. 8. Comparison between FODA/IBEA, DRIFS and FEEDERS
for Fractal, Poisson and Impulsive traffic.
Mean end-to-end delay versus single station load. 48 stations.

The respective merits of the considered schemes are summarised in Table 2.

	FI-I	DR-I	FR-I	FI-P	DR-P	FR-P	FI-F	DR-F	FR-F
4 stations	+	++	+++	**	*	***	##	#	###
12 stations	+	++	+++	**	*	***	#	##	###
32 stations	+	+++	++	**	***	*	##	###	#
48 stations	++	+++	+	**	***	*	##	###	#

Table 2. End-to-end delay of the considered access schemes.
FI=FODA/IBEA; DR=DRIFS; FR=FEEDERS
Three symbols = best behaviour (in decreasing order)

Table 2 highlights that the advantages in using distributed schemes are higher when the traffic burstiness increases. This result confirms what intuition suggests.

The 12 station case was investigated in more detail. Figures 9÷11 show the 95, 99 and 99.8 percentiles of the channel delay versus the channel load for the three schemes and for the three traffic models, respectively. Figures 12÷20 show the end-to-end mean delay of each station as a function of the station load for the three schemes and for the three traffic types, respectively. Stations 7÷9 are equally loaded, so they are represented by only one line. The same is true for stations 10÷12. The original goal was to make all the stations experience the same delay when the channel is not saturated, independently of the loading condition of each station. This aim has roughly been reached with Poisson and Fractal traffics by all three schemes, though the distributed schemes perform better than the centralised one. When the channel approaches saturation, in all cases the most loaded stations are penalised. The effect is due to the maximum allocation limit present in all the schemes. When the system is loaded with Impulsive traffic, the delay tends to increase almost linearly with the load of the stations in all three schemes. Note, however, that the Impulsive traffic pattern used represents the most severe loading condition. The best performance of FEEDERS is evident for all the traffic types, as already seen by looking at the channel delay characteristics.

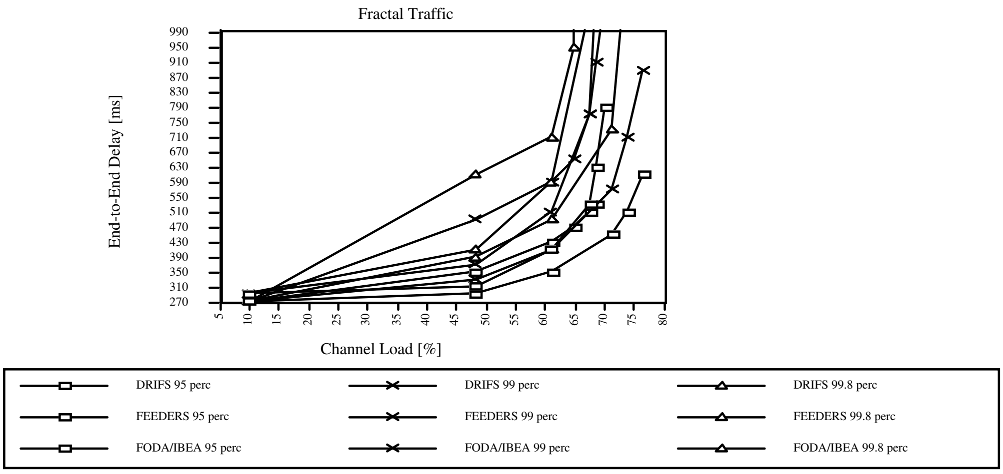


Fig. 9. 12 stations. Fractal Traffic
Percentiles of the channel end-to-end delay versus channel load.

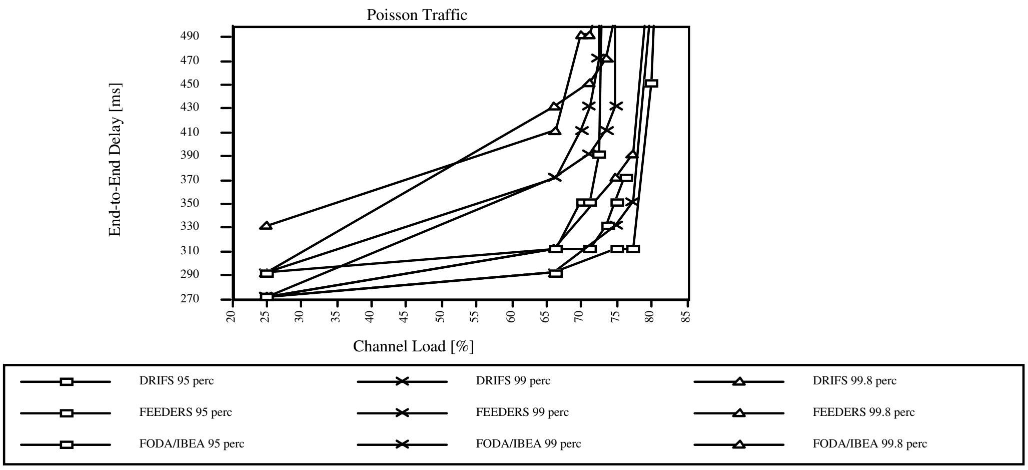


Fig. 10. 12 stations. Poisson Traffic
Percentiles of the channel end-to-end delay versus channel load.

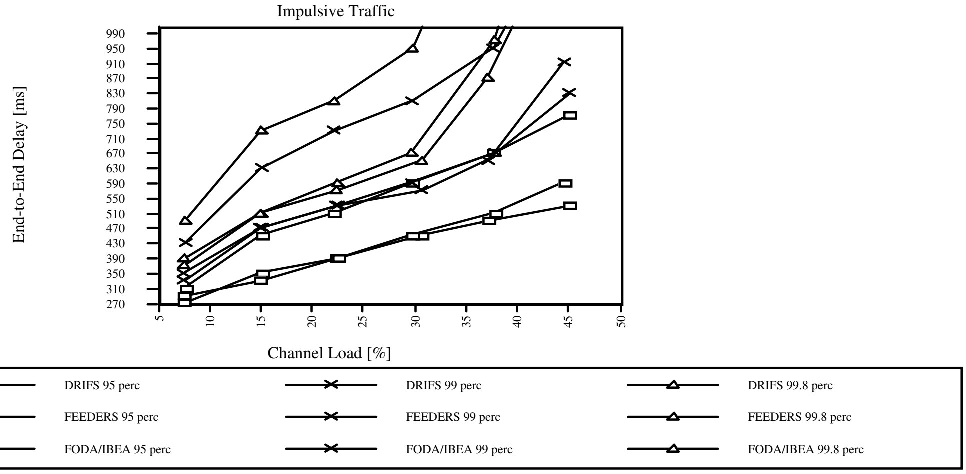


Fig. 11. 12 stations. Impulsive Traffic
Percentiles of the channel end-to-end delay versus channel load.

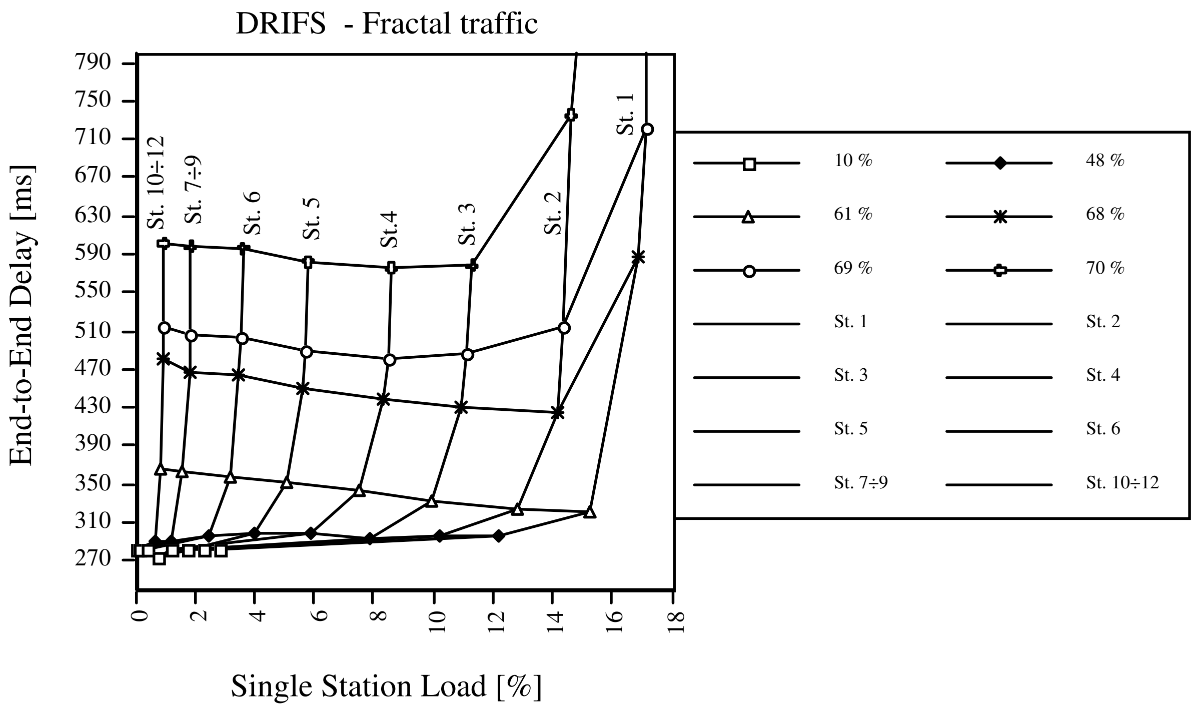


Fig. 12. 12 Stations. Fractal Traffic. DRIFS scheme
Mean end-to-end delay versus single station load

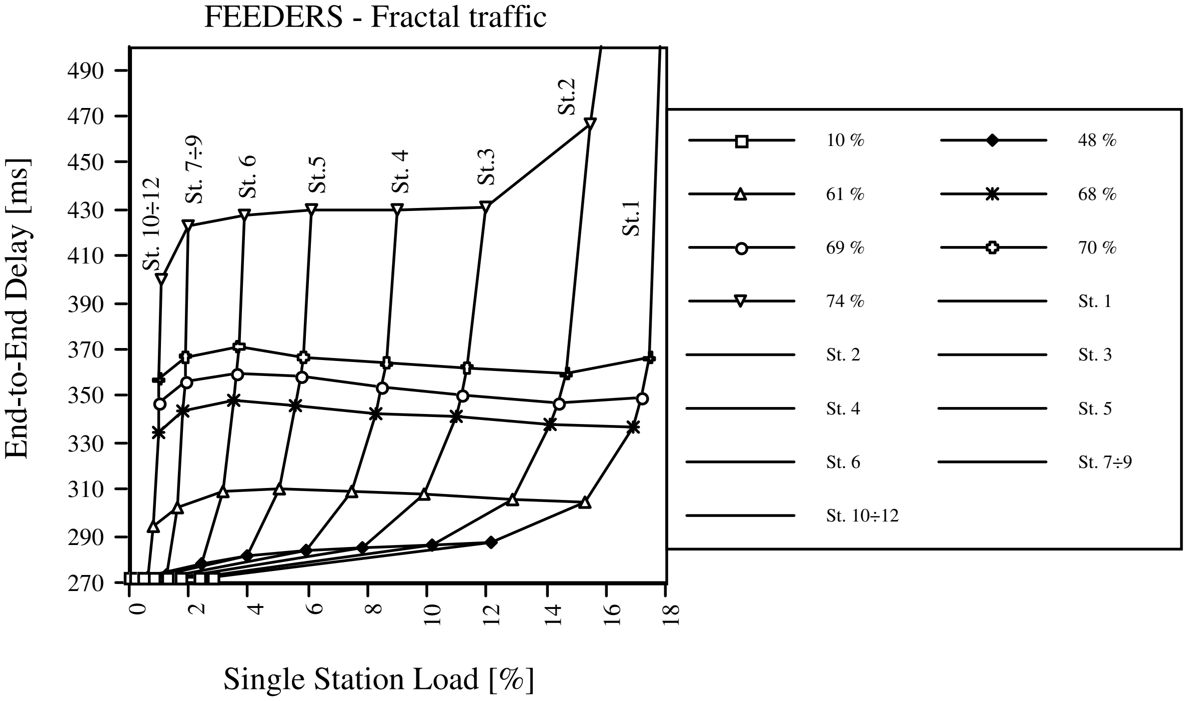


Fig. 13. 12 Stations. Fractal Traffic. FEEDERS scheme
Mean end-to-end delay versus single station load

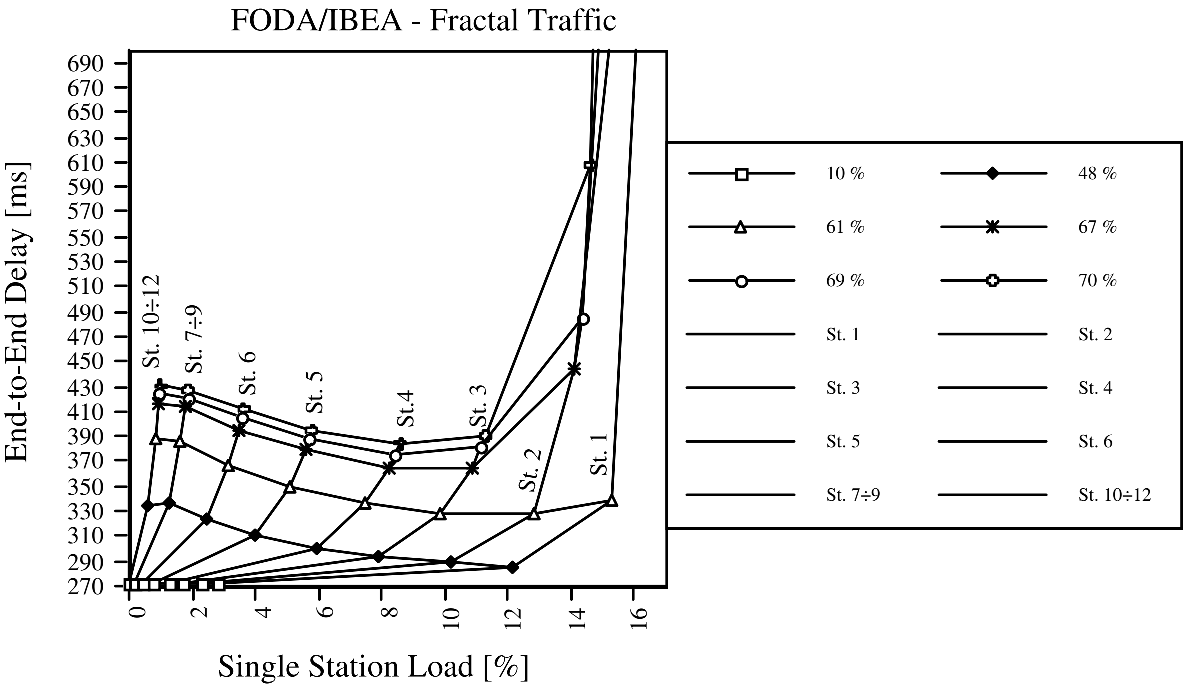


Fig. 14. 12 Stations. Fractal Traffic. FODA/IBEA scheme
Mean end-to-end delay versus single station load

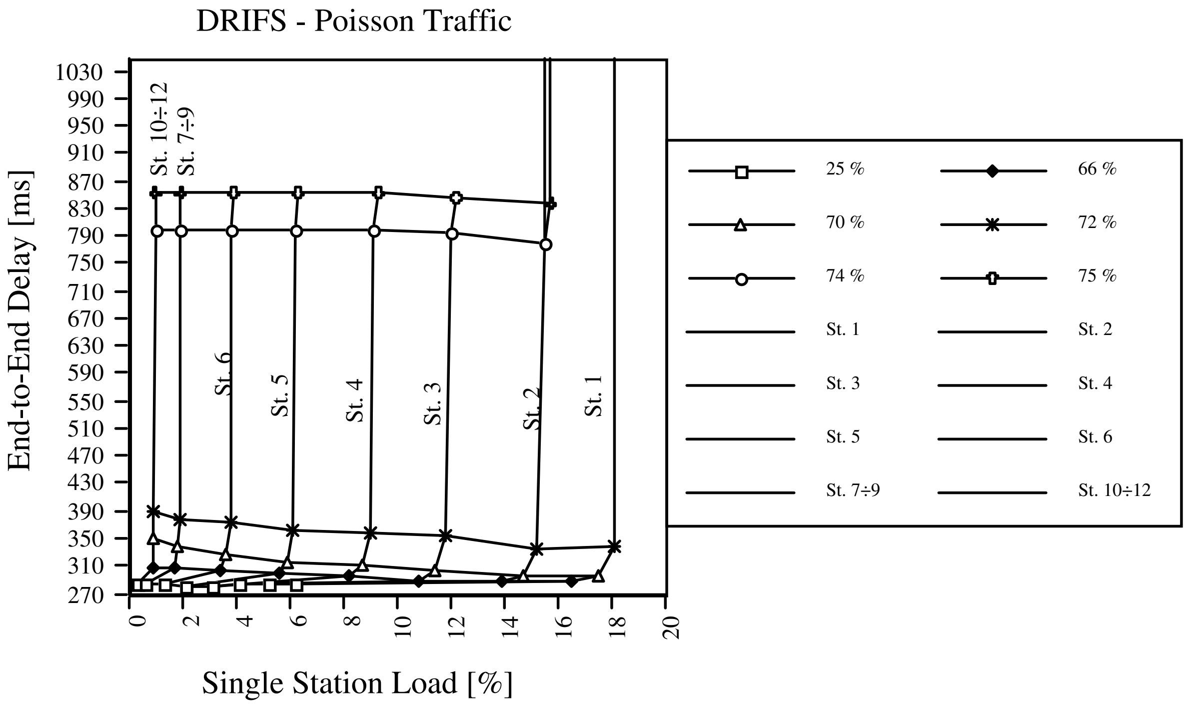


Fig. 15. 12 Stations. Poisson Traffic. DRIFS scheme.
Mean end-to-end delay versus single station load

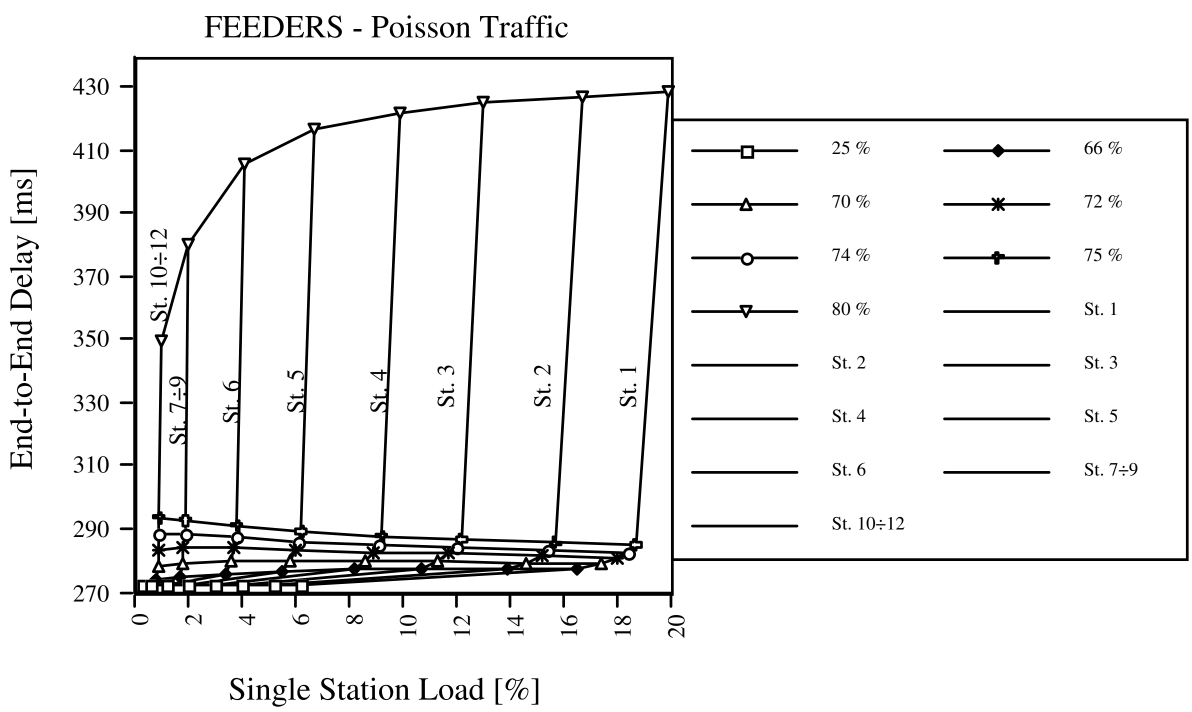


Fig. 16. 12 Stations. Poisson Traffic. FEEDERS scheme.
Mean end-to-end delay versus single station load

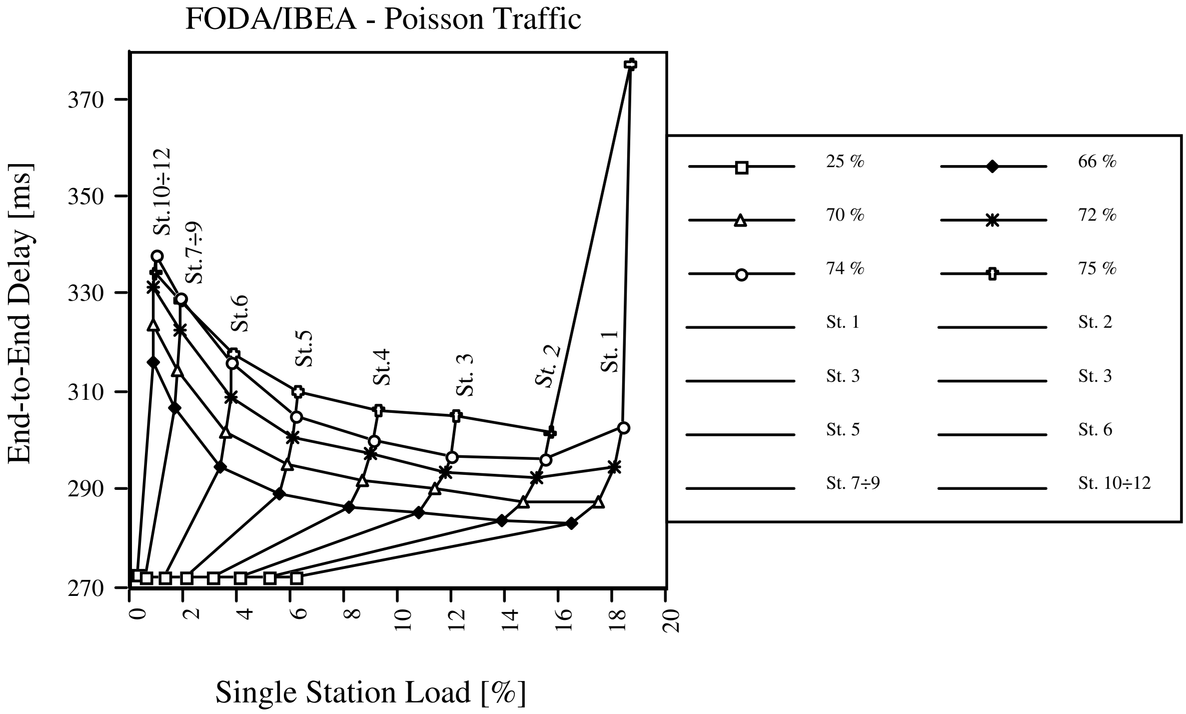


Fig. 17. 12 Stations. Poisson Traffic. FODA/IBEA scheme.
Mean end-to-end delay versus single station load

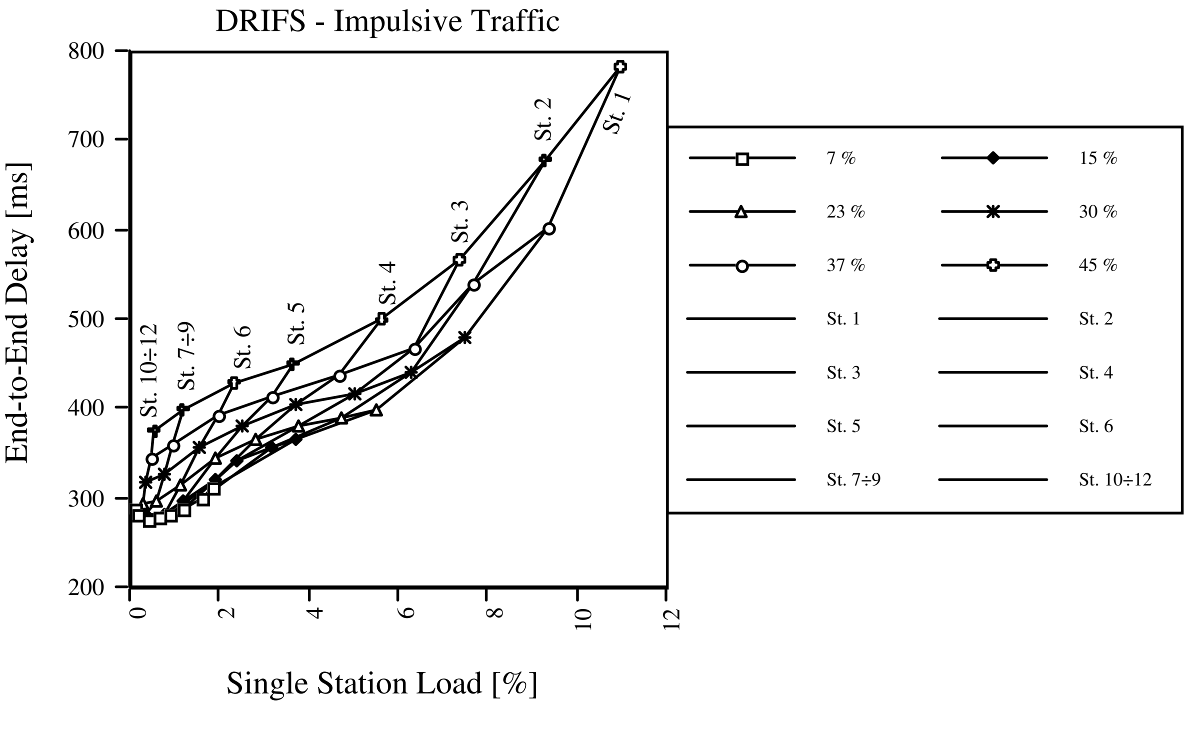


Fig. 18. 12 Stations. Impulsive Traffic. DRIFS scheme.

Mean end-to-end delay versus single station load

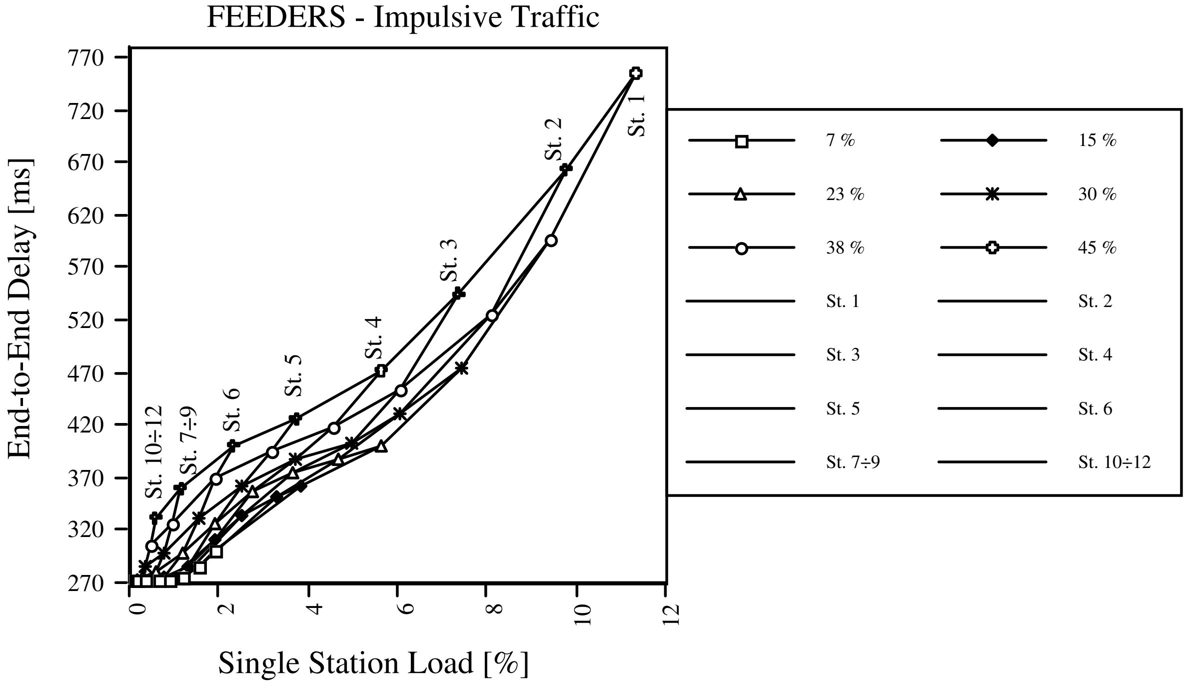


Fig. 19. 12 Stations. Impulsive Traffic. FEEDERS scheme.
Mean end-to-end delay versus single station load

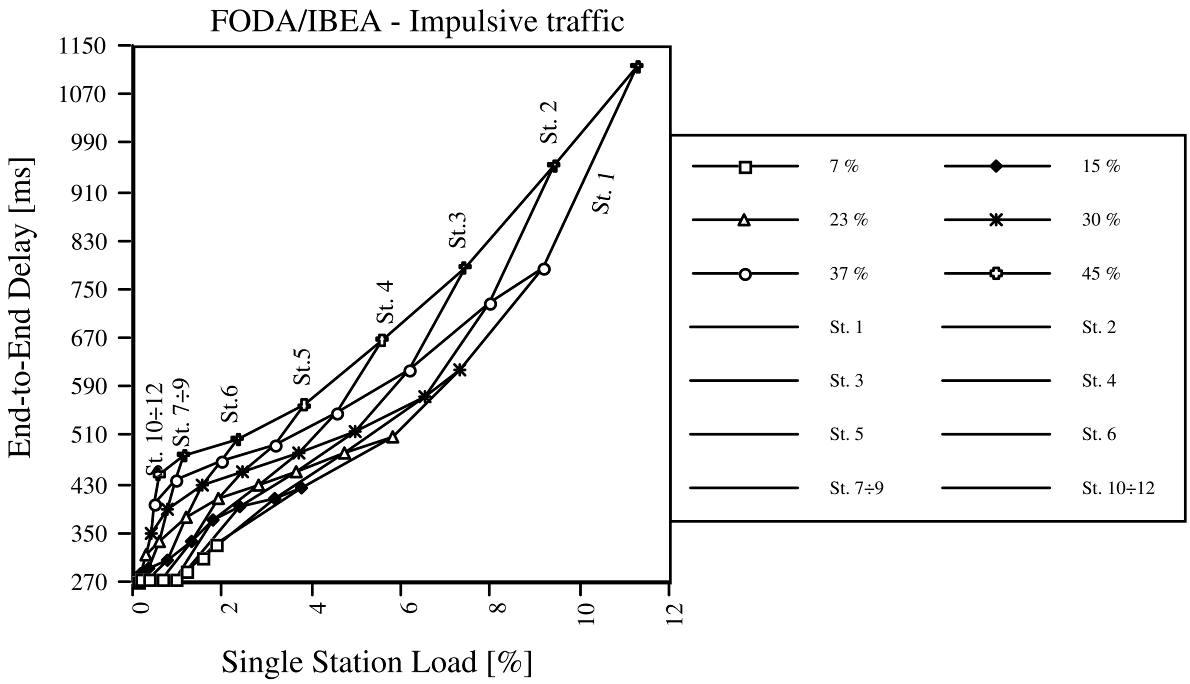


Fig. 20. 12 Stations. Impulsive Traffic. FODA/IBEA scheme.
Mean end-to-end delay versus single station load

5. CONCLUSIONS and FUTURE WORK

The simulation results highlight that the FEEDERS access scheme works much better than FODA/IBEA and DRIFS with any type of traffic if the number of stations is not very high (less than 32). From 32 on, DRIFS and FODA/IBEA perform better, since FEEDERS cannot accommodate too many stations in one frame. The optimal distributed algorithm is thus a compromise between FEEDERS and DRIFS. This scheme should work like FEEDERS until the number of stations is less than 32, then, according to the increasing number of stations, it should enlarge the allocation frame to C_c frames, as designed for DRIFS. This is an interesting result, and the performance of such a scheme could be investigated as a future work. Another interesting result is that FODA/IBEA is competitive with DRIFS. Since both the real experiment and the simulation results indicate that FODA/IBEA is a rather good scheme, it will be interesting to compare it with the centralised scheme named CFRA (Combined/Fixed Reservation Assignment) whose performance has been studied with the support of the European Space Agency [19].

REFERENCES

1. Nedo Celandroni, Erina Ferro, Nick James, Francesco Potortì
"FODA/IBEA: a flexible fade countermeasure system in user oriented networks", Inter. Journal of Satellite Communications, 10, No. 6, pp. 309-323, 1992.
2. Nedo Celandroni, Erina Ferro, Francesco Potortì
"Experimental results of a demand-assignment thin route TDMA system", submitted to the International Journal on Satellite Communication, October 1995.
3. Nedo Celandroni, Erina Ferro, Francesco Potortì

"The performance the FODA/IBEA satellite access scheme measured on the Italsat satellite", proceedings of the ICDSC-10 Conference, pp. 332-328, Vol. 1, Brighton 15-19 May 1995, UK.

4. N. Celandroni, E. Ferro, F. Potortì, A. Bellini, F. Pirri
"Practical experiences in interconnecting LANs via satellite", CCR ACM SIGCOMM Computer Communication Review, Vol. 25, No. 5, pp.56-68, October 1995.

5. Nedo Celandroni, Marco Conti, Erina Ferro, Enrico Gregori, Francesco Potortì
"A bandwidth assignment algorithm on a satellite channel for VBR traffic", submitted to the International Journal on Satellite Communication, February 1995.

6. Erina Ferro, Francesco Potortì
"Study of distributed algorithms for satellite channel capacity assignment in a mixed traffic and faded environment. Part I. The DRIFS-TDMA proposal", submitted to the International Journal on Satellite Communications, February 1995.

7. Nedo Celandroni, Francesco Potortì
"Study of distributed algorithms for satellite channel capacity assignment in a mixed traffic and faded environment. Part II. The FEEDERS-TDMA proposal", submitted to the International Journal on Satellite Communications, February 1995.

8. Petr David W., Murthy K.M.S., Frost Victor S., Neir Lyn A.
"Modelling and Simulation of the Resource Allocation Process in a Bandwidth-on-demand Satellite Communication Network", IEEE Journal on Selected Areas in Communications, Vol. 10, No 2, February 1992, pp. 465-477.

9. Chang Jin-Fu, Lin Sung-Hui
"Delay Performance of VSAT-Based Satellite Wide Area Networks", International Journal of Satellite Communications, Vol. 11, No.1, January-February 1993, pp.1-12.

10. Tasaka Shuji, Ishibashi Yutaka
"A Reservation Protocol for Satellite Packet Communication. - A Performance Analysis and Stability Considerations", IEEE Transactions On Communications, Vol. Com-32, N. 8, August 1984, pp. 920-927.

11. Delli Priscoli F., Listanti M., Roveri A., Vernucci A.

"A Distributed Access Protocol For An ATM User Oriented Satellite System", ICC '89, Boston (USA), June 11-14, Vol. 2, pp. 690-694.

12. Baiocchi A., Carosi M., Listanti M., Roveri A.
"Modelling of a Distributed Access Protocol for an ATM Satellite System: An Algorithmic Approach", IEEE Journal on Selected Areas in Communications, Vol. 9, No 1, January 1991, pp. 65-75.

13. Lam S.S.
"Satellite Packet Communication. Multiple Access protocol and Performance", IEEE Transactions on Communications, Vol. COM-28, No. 4, April 1980, pp. 468-488.

14. Will E. Leland, Murad S. Taqqu, Walter Willinger, Daniel V. Wilson
"On the self-similar nature of Ethernet Traffic (extended version)", IEEE/ACM Transactions on Networking, Vol. 2, No. 1, February 1994.

15. Vern Paxon, Sally Floyd
"Wide Area Traffic: the failure of Poisson Modelling", IEEE/ACM Transactions on Networking, Vol. 3, No. 3, June 1995.

16. Wing-Cheong Lau, Ashok Erramilli, Jonathan L.Wang, Walter Willinger
“Self-Similar Traffic Generation: The Random Midpoint dispalcement Algorithm and Its Properties”, ICC ’95, Seattle, Washington USA, June 18-22 1995, pp. 466-472.

17. Samples of external Ethernet traffic measured at Bellcore during October 1989 are available with anonymous FTP from flash.bellcore.com, directory pub/lan_traffic/OctExt.TL.

18. Ilkka Norros
“The management of Large Flows of Connectionless Traffic on the Basis of Self-Similar Modelling”, ICC ’95, Seattle, Washington USA, June 18-22 1995, pp. 451-455.

19. Tarif Zein, Gerard Maral
"Performance of the Combined/Fixed Reservation Assignment (CFRA) scheme for aggregated traffic", COST-226 (Integrated Space/Terrestrial Networks), Final