

Improving the Spatial Distribution Algorithm of the Traffic-Flow Analysis

L. Muka and G. Lencse

Elassys Consulting Ltd., H-1026 Budapest, Törökvész lejtő 10/a.
Phone: + 36 1 346-8814, fax: + 361 346-8814
e-mail: muka.laszlo@elassys.hu, lencse@sze.hu

Abstract: This paper investigates and improves an important algorithm that is used in the Traffic-Flow Analysis (TFA). TFA can be used for the fast and approximate (performance) analysis of Information and Communication Technology (ICT) systems. The method contains an algorithm for the spatial distribution of the traffic in the system. It is shown how the error of the spatial distribution can be measured, and the effect of the so called *size of routing unit* parameter (S_{RU}) of the algorithm is investigated. In order to define S_{RU} , the Aggregated Traffic Tree-Model (ATTM) is introduced. Using step-by-step Discrete-Event Simulation (DES) and appropriate statistical algorithms in ATTM a method for determination of S_{RU} is described. A method for the refinement of the S_{RU} – based on the metrics defined for the measurement of TFA results – is introduced. A method taking into account Quality of Service (QoS) requirements in S_{RU} refinement is also described in an example.

Keywords: discrete-event simulation, traffic-flow analysis, information and communication systems

1. Introduction

1.1. Performance analysis methods

Discrete-Event Simulation (DES) is a widely used method for the performance analysis [3] of ICT and BP systems. The simulation of large and complex systems requires a large amount of memory and computing power that is often available only on a supercomputer. Efforts are made to use multiprocessor systems or clusters of workstations. The conventional synchronisation methods for parallel simulation (e.g. conservative, optimistic) [1] use event-by-event synchronisation and they are unfortunately not applicable to all cases, or do not provide the desirable speedup. The Statistical Synchronisation Method proposed by Pongor [11] does not exchange individual messages between the segments but rather the statistical characteristics of the message flow. The method can produce excellent speed-up [4] but has a limited area of application [5].

The Traffic-Flow Analysis [6] was proposed for the rapid performance estimation of ICT systems. Its spatial distribution algorithm was studied first in [9]. The applicability

of Traffic-Flow Analysis for large networks and for ICT and BP systems was examined in [7] and [10].

1.2. The Traffic-Flow Analysis

The Traffic-Flow Analysis (TFA) is a combination of simulation and analytical and/or numerical methods. While the traditional discrete-event simulation models the travelling of each packet through the network, TFA uses statistics to model the networking load of applications [6], [8]. TFA works in two stages:

In the *first stage*, the method *distributes the traffic* (the statistics) in the network, using routing rules and routing units.

In the *second stage*, the influence of the *finite capacities* (line and switching-node capacities) is calculated.

The important features of TFA:

- The results are approximate but the absence and the place of *bottlenecks* is shown by the method.
- The *execution time* of TFA is expected to be significantly less than the execution time of the detailed simulation of the system.
- TFA describes the *steady state* behaviour of the network (there is no need for warm-up time definition).

2. Stating the problem of spatial distribution

TFA is a general method, and can be used with any traffic model that satisfies the requirements of TFA for the traffic model. In [6], there were proposed *bit-throughput distribution* and *packet-throughput distribution* (practically histograms) as traffic models to model the traffic on the lines and in the nodes, respectively.

The traffic model is always an *aggregated traffic model*, that is, it represents the *complete traffic of a given type of applications* that are *connected to the given node*. For example it represents the full traffic (*in both directions*) of 35 *FTP* applications that are connected to a router (by switches). If *static* routing is used, we can handle the complete traffic of the before mentioned 35 *FTP* applications (or 100 *web browsers* or any other type of applications) together: we must route only one statistics package through the network (containing the two types of histograms). However, if we have *adaptive* routing, then the traffic of a given type of application should not be handled together, rather it must be routed in multiple packets (routing units), each of which *represent a given portion of the traffic of the given type of application connected to the given node*. According to the previous definition, the number of routing units may be a number between one and the number of the units of traffic generated during the detailed simulation of the system.

When determining the *size of the routing unit* (S_{RU}), we must consider the following issues:

The larger S_{RU} we choose the fewer statistics packages are to be routed in the first phase and the less traffic model addition operations are to be performed in the second phase of TFA. However, if S_{RU} is too large, the spatial distribution of the traffic may considerably differ from the one that is formed in the detailed simulation of the system (and from the one in the real system). If S_{RU} is small, the spatial distribution of the traffic may be quite precise, but the larger amount of messages to be routed and traffic models to be added slow down the analysis. *The choice of S_{RU} ¹ must be a reasonable compromise (between the contradicting requirements) that is made in the knowledge of the whole system modelled.*

3. Defining the Aggregated Traffic Tree-Model

First, for the analysis in order to define S_{RU} , let us introduce the tree model of the aggregated traffic. The Aggregated Traffic Tree-Model (ATTM, shown in Figure 1) is a *fragment* of the whole examined network model which consists of *nodes* and *lines* with *application models*. The ATTM has the following features:

- The application model a_y^x generates the aggregated traffic of all the applications of the y -th class (e.g. VoIP, FTP, web browsing, etc.) connected to node h_x , that is a_y^x contains all the application models of the y -th class for node h_x and a_y^x generates the full traffic between the *source and destination node pairs* ($h_x - h_j, h_x - h_{j+1}, \dots, h_x - h_{j+d}$) in *both directions*. (In Figure 1, the application model a_y^x is in a rectangle drawened with thick line, the logical connection of a_y^x to node h_x is represented by thick line, nodes are shown with thick-line circles and lines of the ATTM model are drawened by thick lines between circles.)

¹ Examples for the influence of the selection of different values for S_{RU} are described in [6].

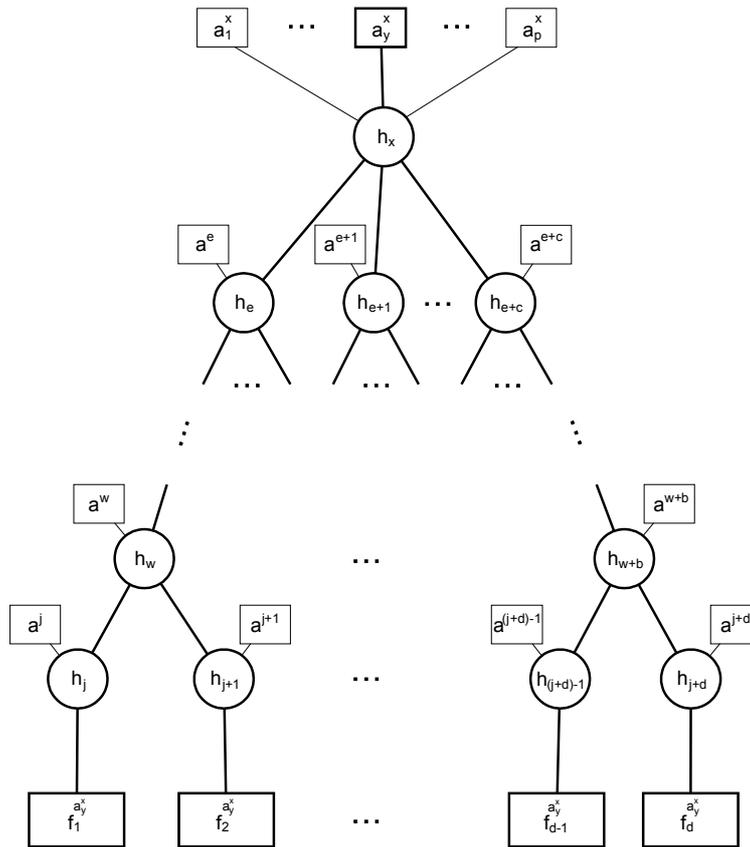


Figure 1. The Aggregated Traffic Tree-Model of an Application

- Other application models generate the traffic *independently* from a_y^x : the application models connected to h_x (a_1^x, \dots, a_p^x) represent the traffic of other classes; application models connected to other nodes (for example a^e, a^w , and a^j connected to nodes h_e, h_w and h_j respectively) represent the application models of all classes. (In Figure 1, all these applications are shown with thin-line rectangles and their logical connections to nodes are also represented by thin lines.)
- Any other application model connected to any other node in the model of the *examined network* can generate traffic for the nodes and lines of ATTM.
- ATTM contains only the nodes and lines used by the traffic generated by a_y^x .
- The *routing* in the *examined network* is an *adaptive* routing with unknown features.

- In ATTM there is only *one route* between the source node and a destination node. (A route is a sequence of *nodes* and *lines* between the source node and a destination node related to a_y^x). If there are more routes between the source and the same destination used by the traffic generated by a_y^x then each route is handled separately as if it has a different destination. For these “*split destinations*”, the traffic (for each unit of the traffic) is traced and recorded separately for all the elements of the route. Thus, it also means, that more split destination of ATTM will represent one original, non-split destination. (Splitting of destinations is performed according to the traffic generated by event-by-event simulation in the examined network.)
- In the ATTM, the traffic generated by a_y^x is observed at destination nodes $h_j, h_{j+1}, \dots, h_{j+d}$. The frequency distribution of the traffic observed at destination node h_{j+d} is denoted by $f_d^{a_y^x}$ and shown in a rectangle drawn by thick line, and the logical connection of which to node h_{j+1} is also represented by a thick line. (The exact description of the measurement of this frequency distribution is given in point 4. of this paper.)

4. Statistical analysis of the aggregated traffic

Now, using the ATTM, let us examine the *behaviour of the aggregated traffic* generated by the application model a_y connected to the node h_x (that is by a_y^x) between the *source* and *destination* node pairs ($h_x - h_j, h_x - h_{j+1}, \dots, h_x - h_{j+d}$) in both directions at destinations. The traffic between the node pairs is *influenced* by node and line capacities, by the traffic generated of other application models and by the routing (which is *adaptive routing* for now) in the examined network. All the application models are detailed DES (Discrete-Event Simulation) models and generate the traffic according to the event-by-event simulation method. The aggregated traffic at destinations is examined in time intervals of T length. The T^2 interval is selected of the same length as the throughput collection interval in TFA. For the observation, in ATTM, the aggregated traffic is *recorded* for each destination of a_y in the subsequent T long time intervals.

To *observe* the aggregated traffic at destinations of a_y in ATTM, the *statistical sampling* approach will be applied: the statistical sampling method is used to define the *mean aggregated traffic* at a destination in T long time intervals. There are observed frequency distributions generated by a_y : $f_i^{a_y^x}$ denotes the frequency distribution at destination i

$$(f_i^{a_y^x} \in \{f_1^{a_y^x}, f_2^{a_y^x}, \dots, f_d^{a_y^x}\}).$$

The mean \bar{k}_i of the observed frequency distribution $f_i^{a_y^x}$ is calculated by the formula:

² The influence of the selection of the value for the T parameter is analysed in [6].

$$\bar{k}_i = \frac{\sum_j f_{ij} k_{ij}}{\sum_j f_{ij}} \quad (1)$$

where k_{ij} denotes the number *units of the aggregated traffic* observed in T time interval at destination i and f_{ij} is the observed frequency of k_{ij} .

In using the statistical sampling method, both *infinite* and *finite population* approach may be applied. (In case of simulation of the traffic for long-run, the observation is characterised by a large number of T intervals and may be approximated by an *infinite population*, but the observation of simulation of one day probably may be modelled by using the *finite population* approach.)

The mean aggregated traffic, m_i , at a destination i can be calculated using the formula:

$$m_i = \bar{k}_i \pm B_i, m_i \in \{m_1, m_2, \dots, m_d\} \quad (2)$$

where \bar{k}_i is the *mean aggregated traffic of the sample* (frequency distribution), B_i is the allowable error for determination of the mean m_i at destination i .

The sample size which is necessary to define the m_i with the precision B_i depends on the *method of sampling* too.

Let us examine the application of the *simple random sampling (SRS)* method which has the advantages that requires minimum knowledge of *population* and free of possible classification errors. The *sample size* for *SRS* samples *without replacement from an infinite population* may be calculated according to the formula:

$$n = \frac{z^2 \sigma^2}{B^2} \quad (3)$$

where n is the required sample size, σ denotes population standard deviation, z is the z score determined for a specific confidence level and B is the allowable error.

The *sample size for SRS* samples without replacement from a *finite population* with *finite population correction factor* may be determined as:

$$n = \frac{\sigma^2 N}{\frac{(N-1)B^2}{z^2} + \sigma^2} \quad (4)$$

where N is the population size.

Let us see an *example* of sampling analysis of the aggregated traffic at a destination: the aggregated traffic is measured in packets, the length of T interval is 5sec, the required confidence level is 95% ($z_{0.05} = 1.96$), the allowable error $B = 10$ packets, the finite population sizes are $N=17280$ and $N=120960$ (according to the 1-day and 7-day long observation intervals).

The value of the population standard deviation is determined using an *estimate*: using the *GAM* (General Application Model) *transaction parameters of TFA* [6] for a_y , connected to the node h_x the *range* of the aggregated traffic may be determined. (The *GAM* parameters which can be used for the determination of the range are: $N_p^{\{s/r\}}(tt)$)

the number of packets sent or received by the application during a transaction of the given type, $N_{TrPD}(tt, day)$ the number of transactions performed by an application from the given transaction type on a given day of the week, $I_{Tr}(tt, day, time)$ the probability density function describing the time distribution of the transactions of the given type on the given day of the week.)

For the example, the range of the aggregated traffic determined this way is 290 packets for 5 sec, thus – according to the *estimation* based on *Chebyshev's theorem* – s (sample standard deviation) $\cong range/4 \cong 72.5$ packets $\cong \sigma$.

Thus, the sample sizes for SRS sampling from finite populations are:

$n_{N=17280} = 199.60 \cong 200$ and $n_{N=120960} = 201.59 \cong 202$ (after rounding it is the same size as the sample size for SRS sampling from *infinite* population: $n = 201.92 \cong 202$). After performing the sampling according to the determined sample size, it can be asserted with 95% confidence that the mean aggregated traffic m at the examined destination can be calculated as $m = \bar{k} \pm 10$ packets, where \bar{k} is the mean aggregated traffic of the sample.

The sample sizes should be determined for each destination of a_y , according to the method described in the previous example and the *largest sample size* should be used for sampling:

$$n = \sup \{n_1, n_2, \dots, n_a\} \quad (5)$$

The use of *other sampling methods* may also be considered:

If there is more available preliminary knowledge about the behaviour of the aggregated traffic generated by the application model, then, for example, the *stratified sampling* method may be used. Applying this method the *variability* of data within strata, as well as their *size* may be taken into account.

The approach of *sampling based on the expert judgment* and on *convenience* – that is on non-probability considerations – may simplify the execution of sampling. (Using the expert judgement approach, for example, one or more *typical* or *critical* intervals for observation are selected and the sampling is executed taking into account the necessary warm-up intervals for the simulation. The approach using the convenience considerations may mean, for example, using only the intervals for which data have been collected.) The disadvantage of this approach is the occurrence of uncontrolled *bias* in sampling.

5. Determining the size of the routing unit

Now, let us formulate the requirements for S_{RU} . Let us use *TFA application models* in the *ATTM*: the aggregated traffic in the examined network is generated by the *GAM* of TFA. Let S_{RU} be the *size of the unit of the traffic* that can be handled together for the given *type of applications* connected to the *given node*. If the S_{RU} is increased then the amount of the necessary computations will be decreased but the resolution of the traffic is decreased thus the *uncertainty* will be increased and the reliability of the results will be decreased.

How to find a compromise? Let us call for help the *chi-squared test* for “goodness of fit”.

The calculation for the χ^2 -test is expressed by the formula:

$$\chi^2 = \sum_{j=1}^r \frac{(k_j - n^* p_j)^2}{n p_j} \quad (6)$$

where k_j is the observed frequency in class j , p_j is the expected hypothetical probability for class j , n is the number of observations, r is the number of classes, the grade of freedom is $r-1$ and of course $\sum_{j=1}^r p_j = 1$ and $\sum_{j=1}^r k_j = n^*$.

According to the logic of the test, a hypothesis is evaluated: for example if the value of the test statistic is large enough (at a determined level of significance) to lie in the critical region of the test the null hypothesis is rejected and it is concluded that the theoretical expected distribution is not a good fit to the observed distribution.

The necessary sample size – in order to have a good approximation to a chi-squared distribution – may be determined according to the inequality $n p_j \geq 10$ (which is well known from statistics) where p_j is the minimum of the expected theoretical probability.

For further consideration, let us use the *frequency distributions observed* at destinations during the *ATTM DES work sampling* described in the previous point.

The observed *DES-sampling* frequency distribution at destination i is $f_i^{a_y^x}$ ($f_i^{a_y^x} \in \{f_1^{a_y^x}, f_2^{a_y^x}, \dots, f_d^{a_y^x}\}$), the sampling mean aggregated traffic at destination i is m_i ($m_i \in \{m_1, m_2, \dots, m_d\}$) Let us calculate the expected probabilities using the observed *DES-sampling* frequency distributions taking into account the number of *classes* for the expected hypothetical distribution. The probability of class j calculated this way at destination i is p_{ij} , $p_{ij} \in \{p_{i1}, p_{i2}, \dots, p_{ir}\}$. Now, let us use these probabilities to determine the necessary sample sizes. The minimum probability at the destination i may be expressed as:

$$\inf \{p_{i1}, p_{i2}, \dots, p_{ij}, \dots, p_{ir}\} \quad (7)$$

Thus – using the inequality $n^* p_j \geq 10$ (that is $n^* p_{ij} \geq 10$) – the *necessary* sample size at destination i may be expressed as follows:

$$n_i^* \geq \frac{10}{\inf \{p_{i1}, p_{i2}, \dots, p_{ij}, \dots, p_{ir}\}}, n_i^* \in \{n_1^*, n_2^*, \dots, n_d^*\}, (j = 1, 2, \dots, r) \quad (8)$$

Now, taking into account the observed means of the aggregated traffic at destinations of a_y^x , the number of required *RUs* for a_y^x may be determined using the following formula:

$$N_{RU}^{a_y^x} \geq \sup_i \left\{ \frac{1}{\inf \{p_{i1}, p_{i2}, \dots, p_{ij}, \dots, p_{ir}\}^{\sum_{s=1}^d m_s}} \right\} \quad (9)$$

where $m_i \in \{m_1, m_2, \dots, m_d\}$, ($i = 1, 2, \dots, d$).

In $N_{RU}^{a_y^x}$ is taken into account – because of the definition of ATTMM – the weight (m_i) and the shape (n_{i^*}) of every route for a_y^x destinations.

The required S_{RU} for the given type of application a_y connected to the node h_x in ATTMM may be calculated according to the *inequality*:

$$S_{RU}^{a_y^x} \leq \frac{N_u^{a_y^x}}{N_{RU}^{a_y^x}} \quad (10)$$

where $N_u^{a_y^x}$ is the amount of traffic (measured in *packets* or in *bits*) that has to be transferred in transactions generated by a_y .

6. Refining the routing unit size determination

The traffic model is always an aggregated traffic model, that is, it represents the complete traffic of a given type of applications that are connected to the given node. In the aggregated traffic model of TFA, the traffic of all application models of the same class – that is the *traffic of the same class* – connected to the *given node* is modelled together.

In order to *refine* the determination of S_{RU} , let us examine the determination of S_{RU} *together with the classes of traffic* of TFA.

The traffic in TFA may be expressed using the following formula:

$$\mathcal{T} = \bigcup_{q=1}^Q \bigcup_{v=1}^V \mathcal{T}_{qv} \quad (11)$$

where V is the number of nodes, Q is the number of traffic classes, \mathcal{T}_{qv} is the traffic generated by the application model a_q^v , and \mathcal{T} is the whole traffic generated by the *GAM* of TFA. To increase the number of RUs for traffic \mathcal{T}_{qv} ($N_{RU}^{T_{qv}}$) means the change of the S_{RU} parameter of the application model a_q^v generating the traffic of class q for node v ($N_{RU}^{T_{qv}} = N_{RU}^{a_q^v}$).

6.1. Introducing metrics for evaluation

To be able to determine good enough values for S_{RU} that take into account the traffic classes too it is necessary to introduce tools to measure the features of a given spatial distribution of the traffic in TFA.

The *capacity matrix* $\underline{K} = [k_{ij}]$ describes the capacity of *nodes* and *lines* in the *examined network*. The capacity matrix is a $V \times V$ matrix, where V is the number of nodes in the examined network. Matrix element k_{ii} is the routing capacity of node i (measured in packets per second), and matrix element k_{ij} , where $i \neq j$ is the transmission capacity of the line from node i to node j , measured in Mbit/s. If there is no transmission line from node i to node j then $k_{ij} = 0$.

The results of the TFA are described by the *traffic matrix* $\underline{T} = [t_{ij}]$. Matrix element t_{ii} describes the result of TFA for node i : both the packet-throughput distribution (P_{ii}) and the delay distribution (D_{ii}) of TFA resulted in node i , and matrix element t_{ij} describes the result of TFA for the line from node i to node j : both the bit-throughput distribution (P_{ij}) and the delay distribution (D_{ij}) of the line from node i to node j .

To evaluate *TFA distribution procedure* from the point of view of utilisation of the network elements, let us determine the *utilisation matrix* $\underline{U} = [u_{ij}]$ that gives us the evaluation of the load of every node and link:

u_{ii} = (the average number of packets - defined by the arithmetic mean of the load in the steady state - for the node i) / k_{ii} ,

u_{ij} = (the average number of Mbits - defined by the arithmetic mean of the load in the steady state - for the line from node i to node j) / k_{ij} .

For the evaluation of the *uncertainty* of results of the given spatial distribution, the *sample evaluation matrix* $\underline{S} = [s_{ij}]$ is introduced. Matrix elements may have the following values:

$s_{ii}, s_{ij} = 3$, *small sample*, if the number of RUs through a given or line $< S_S$

$s_{ii}, s_{ij} = 2$, *medium sample*, if the number of RUs through a given node or line is between S_S and S_L , $S_S \leq$ the number of RUs $\leq S_L$

$s_{ii}, s_{ij} = 1$, *large sample*, if the number of RUs through a given node or line $> S_L$

The matrix element $s_{ij} = 0$ for an application if there was no RU sent through line from node i to node j and $s_{ij} = 0$ for all applications if there is no line from node i to node j as it is sure that there is no traffic on a non-existing line.

For the \underline{S} matrix to be used, the *limits* S_S and S_L should be determined for applications: S_L may be got from the RU defining procedure described in the previous points, the value S_S may for example be determined according to the sample size to be a *small sample* for sampling (30).

6.2. Increasing the number of routing units

Let us examine some typical *conditions* where it is necessary to *increase* the number of RUs for traffic \mathbb{T}_{qv} ($N_{RU}^{\mathbb{T}_{qv}}$).

Condition of *high uncertainty (HU)* of traffic distribution: after evaluation of the \underline{S} matrices, it may be found that:

$$\exists_{ii}(s_{ii}(\mathbb{T}_{qv}) = 3) \vee \exists_{ij(i \neq j)}(s_{ij}(\mathbb{T}_{qv}) = 3) \quad (12)$$

which expresses that there has been measured *small sample* in the distribution of traffic \mathbb{T}_{qv} .

Condition of *uncertain utilisation (UU)*: after evaluation of the \underline{U} matrix and the \underline{S} matrices, it may be found that:

$$\exists_{ii}(u_{ii} > U_{node} \wedge (\exists_{\mathbb{T}_{qv}}(s_{ii}(\mathbb{T}_{qv}) = 2) \vee \exists_{\mathbb{T}_{qv}}(s_{ln(l \neq n)}(\mathbb{T}_{qv}) = 2)) \vee$$

$$\forall \exists_{ij(i \neq j)} (u_{ij} > U_{line} \wedge (\exists_{\mathcal{T}_{qv}} (s_{ll}(\mathcal{T}_{qv}) = 2) \vee \exists_{\mathcal{T}_{qv}} (s_{ln(l \neq n)}(\mathcal{T}_{qv}) = 2)) \quad (13)$$

which means that the *limit* defined for utilisation of the network elements has been exceeded (limit for nodes U_{node} or limit for lines U_{line}) together with a *medium sample* of \mathcal{T}_{qv} in the network.

Condition of *uncertain delay (UD)*: after evaluation of the \underline{T} matrix and the \underline{S} matrices, it may be found that:

$$\exists_{\mathcal{T}_{qv}} (D_{\mathcal{T}_{qv}} > D_{limit(\mathcal{T}_{qv})} \wedge (\exists_{\mathcal{T}_{gh}} (s_{ii}(\mathcal{T}_{gh}) = 2) \vee \exists_{\mathcal{T}_{gh}} (s_{ij(i \neq j)}(\mathcal{T}_{gh}) = 2)) \quad (14)$$

which expresses that the delay *limit* defined for \mathcal{T}_{qv} ($D_{limit(\mathcal{T}_{qv})}$) has been exceeded together with a *medium sample* of \mathcal{T}_{gh} in the examined network.

6.3. Routing unit determination with the analysis of traffic classes

Now, let us examine an example of the routing unit determination based on the analysis of traffic classes. In the example, the capacities of a network should be examined from the point of view of capacity needs of different classes of traffic generated by IP communication services on the network, taking into account the *QoS* (Quality of Service) requirements for IP communication services.

The traffic classification used in the example is similar to the classification described in IP Capacity Planning (IPCP) Framework [2] – the traffic is classified depending on the *degree of delay tolerance* – with the difference that the *class for real-time traffic has been divided into two classes* for the examination, according to the *traffic volume* level that has to be propagated:

Traffic class \mathcal{T}_1 – high-volume real-time traffic (e.g. video services)

Traffic class \mathcal{T}_2 – low-volume real-time traffic (e.g. voice)

Traffic class \mathcal{T}_3 – low-delay tolerant traffic (e.g. web browsing)

Traffic class \mathcal{T}_4 – high-delay tolerant traffic (e.g. FTP)

In the TFA modelling, the traffic classes \mathcal{T}_1 and \mathcal{T}_2 are examined with the condition of high network availability (because of their strict delay requirements) and traffic classes \mathcal{T}_3 and \mathcal{T}_4 are examined with *decreased* available network capacities (providing the network is capable to manage the situation). The fulfilment of QoS requirements (specified by the ITU (International Telecommunication Union) for each service class) is examined through the allowable delay for the traffic class using the resulting delay distributions of TFA.

The examination *starts* with the *statistical* determination of S_{RUS} , decision about S_S and S_L values and determination of U and D limitations (the capacity of the network is defined by the \underline{K} matrix). The following steps of the *TFA examination (Step-1 - Step-4)* are not precise descriptions of the examination steps but they tend to be an illustration for the use of routing unit number-decisions in the process of examination:

Step-1

TFA examination of the network with traffic load $\mathcal{T}_{Step-1} = \bigcup_{v=1}^V \mathcal{T}_{1v}$ (i.e. the high-volume real-time traffic)

Execution of the spatial distribution: S_{RU} decisions with the examination of HU , UU and UD conditions

Delay evaluation according to QoS requirements

Step-2

TFA examination of the network with traffic load $\mathcal{T}_{Step-2} = \bigcup_{q=1}^2 \bigcup_{v=1}^V \mathcal{T}_{qv}$ (i.e. the high-volume and the low-volume real-time traffic together)

Execution of the spatial distribution: S_{RU} decisions with the examination of HU , UU and UD conditions

Delay evaluation according to QoS requirements

Step-3

TFA examination of the network with traffic load $\mathcal{T}_{Step-3} = \bigcup_{v=1}^V \mathcal{T}_{3v}$ (i.e. the low-delay tolerant traffic)

The *decreased available network capacity* for the traffic class \mathcal{T}_3 is calculated as follows:

$$k_{ii}(\mathcal{T}_{Step-3}) = k_{ii} (1 - u_{ii}(\mathcal{T}_{Step-2})), k_{ij}(\mathcal{T}_{Step-3}) = k_{ij} (1 - u_{ij}(\mathcal{T}_{Step-2})) \quad (15)$$

Execution of the spatial distribution: S_{RU} decisions with the examination of HU , UU and UD conditions

Delay evaluation according to QoS requirements

Step-4

TFA examination of the network with traffic load $\mathcal{T}_{Step-4} = \bigcup_{v=1}^V \mathcal{T}_{4v}$ (i.e. the high-delay tolerant traffic)

The *decreased available network capacity* for the traffic class \mathcal{T}_4 is calculated as follows:

$$k_{ii}(\mathcal{T}_{Step-4}) = k_{ii}(\mathcal{T}_{Step-3}) (1 - u_{ii}(\mathcal{T}_{Step-3})), k_{ij}(\mathcal{T}_{Step-4}) = k_{ij}(\mathcal{T}_{Step-3}) (1 - u_{ij}(\mathcal{T}_{Step-3})) \quad (16)$$

Execution of the spatial distribution: S_{RU} decisions with the examination of HU , UU and UD conditions

Delay evaluation according to QoS requirements

Remark: Of course, a better utilisation of the network may be achieved with the same QoS parameters if the traffic in the network (with all the traffic classes) is handled together.

7. Conclusions

We have defined a tree model of the aggregated traffic (ATTM) in the examined network and based on this model, a statistical algorithm – which serves for the determination of the size of the routing unit of TFA – has been introduced.

We have introduced formal description for the networks and traffic conditions such as: capacity, cost, traffic and utilization matrices as well as metrics for the difference of the traffic and utilization matrices.

On the basis of the statistical constraints on sample size, we have introduced the sample evaluation matrix, the elements of which express if the number of RUs are high enough for a given node or line.

We have shown how the S_{RU} can be dynamically controlled during the spatial distribution phase of TFA.

We have also shown on an example how the QoS parameters of traffic classes can be taken into account in determining S_{RU} .

We conclude that with our results on the appropriate choice of the S_{RU} of TFA, these methods have been matured for implementation.

References

- [1] Fujimoto, R. M.: *Parallel Discrete Event Simulation*, Communications of the ACM, Vol. 33, No. 10, (1990), pp. 31-53.
- [2] Gareth, D., Hardt, M., Kelly, F.: *Come the Revolution – Network Dimensioning, Service Costing and Pricing in a Packet Switched Environment*, Telecommunications Policy, Vol. 28, No. 5-6, (2004), pp. 391-412
- [3] Jain, R.: *The Art of Computer Systems Performance Analysis*, John Wiley & Sons, New York (1991).
- [4] Lencse, G.: *Efficient Parallel Simulation with the Statistical Synchronization Method*, in Communication Networks and Distributed Systems Conference, San Diego, (1998), pp. 3-8
- [5] Lencse, G.: *Applicability Criteria of the Statistical Synchronization Method*, in Communication Networks and Distributed Systems Conference, San Francisco, (1999), pp. 159-164
- [6] Lencse, G.: *Traffic-Flow Analysis for Fast Performance Estimation of Communication Systems*, Journal of Computing and Information Technology, Vol. 9, No. 1, (2001), pp. 15-27.
- [7] Lencse, G.: *Speeding up the Performance Analysis of Communication Systems*, in 2005 European Simulation and Modelling Conference, Porto, Portugal, (2005) pp. 329-333
- [8] Lencse, G., Muka, L.: *Convergence of the Key Algorithm of Traffic-Flow Analysis*, Journal of Computing and Information Technology, Vol. 14, No 2, (2006), pp. 133-139

- [9] Lencse, G., Muka, L.: *Investigation of the Spatial Distribution Algorithm of the Traffic Flow Analysis and of the Entity Flow-Phase Analysis*, in 2007 European Simulation and Modelling Conference, St. Julians, Malta, (2007) pp. 574-581
- [10] Muka, L., Lencse, G.: *Cooperating Modelling Methods for Performance Evaluation of Interconnected Infocommunication and Business Process Systems*, in 2008 European Simulation and Modelling Conference, Le Havre, France, (2008) pp. 404-411
- [11] Pongor, Gy.: *Statistical Synchronisation: a Different Approach to Parallel Discrete Event Simulation*, in 1992 European Simulation Symposium, Dresden, (1992), pp. 125-129.