Modelling computer networks

G Max
Budapest University of Technology and Economics,
Department of Automation and Applied Informatics
H-1521, Budapest, Po. Box. 91. Hungary

E-mail: max@aut.bme.hu

Abstract. Traffic models in computer networks can be described as a complicated system. These systems show non-linear features and to simulate behaviours of these systems are also difficult. Before implementing network equipments users wants to know capability of their computer network. They do not want the servers to be overloaded during temporary traffic peaks when more requests arrive than the server is designed for. As a starting point for our study a non-linear system model of network traffic is established to exam behaviour of the network planned. The paper presents setting up a non-linear simulation model that helps us to observe dataflow problems of the networks. This simple model captures the relationship between the competing traffic and the input and output dataflow. In this paper, we also focus on measuring the bottleneck of the network, which was defined as the difference between the link capacity and the competing traffic volume on the link that limits end-to-end throughput. We validate the model using measurements on a working network. The results show that the initial model estimates well main behaviours and critical parameters of the network. Based on this study, we propose to develop a new algorithm, which experimentally determines and predict the available parameters of the network modelled.

1. Introduction
In a large-scale computer network data packets may choose among many routes to transfer their destination address. These selections are based on knowledge of recent nodes of the computer networks. In most cases parts of the computer networks do not know all levels of the networks. Though almost all nodes take part in the process of forwarding packets, nodes of the computer network know more on neighbours than far ones. When users want to follow data transfers between starting and destination nodes, they need more knowledge on data transfers. To get know characteristics of computer networks, the simulation is a widely used method. After the introduction the existing works are summarized. In third session our non-linear model and its behaviours are established. The third session shows main elements of the computer networks. Next session explains the estimation process of model parameters and criteria. The fifth session presents main elements of the communication function which were implemented in our model, while in the sixth session a case study is given. The study compares an existing network to our model and observes system parameters. These system parameters and criteria are used to analyze quantitatively behaviour of the system as

---

1 This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Hungary Development Plan (Project ID: TAMOP-4.2.1/B-09/1/KMR-2010-0002).
well as to investigate the impact of system parameters, such as data density, load level, link capacity on system stability. The congestion problems are followed with special attention. Last session shows tendency of future works.

2. Related works
To study network behaviour, traffic parameters are needed to measure and analyse, and to find their statistical laws, such as was done by Bhole and Popescu [1]. After having known the statistical rules, some traffic models are built, such as the establishment of a model for Novel network traffic by Jiang and Papavassiliou in [2] or we try to understand why buffers are overflowed like Sidiroglou et al. did in [3].

If the time-scale of network traffic is considered, the network traffic behaviour will be different in different time-scale. Paxson and Floyd [4] showed that the traffic behaviour of millisecond-time scale is not self-similar by the influence of network protocol. Due to the influence of environment, the traffic behaviour whose time-scale is larger than ten minutes is not also self-similar and is a non-linear time-series. Only the traffic behaviour in second-time scale is self-similar. This problem becomes known when transactions of the computer networks have to be described.

Insufficient operations of the network trace back to congestion problems. Model of Haifeng et al. [5] performs an effective network congestion control method for multilayer network.

If operational cost is important read Chang’s article [6] on Pricing Multicast Communication.

3. Network elements
We may divide the elements of a real computer network into two big groups. On the one hand, the nodes imply the storage units; on the other hand, the transfer mediums connect the nodes to each other.

Forming a computer network several geometry features set conditions which are not implemented in our model:

- Geometrical conditions:
  - Distance of nodes.
  - Materials of the transfer medium.
  - Number of users.

- Seasonal effects:
  - The number of the users using the network actually moves on a very wide scale.

- Diverse external factors:
  - Weather.
  - Electromagnetic storm.

3.1. Nodes
That statement appears unambiguous, that in the network actually the nodes, the active elements of the network communicate with each other, and the nodes form the peaks of the network graph in the graph theory model of the computer network.

3.2. Transfer mediums
The process of the communication is implemented through the transfer medium. The communication happens with almost speed of light. This hilltop speed makes the transfer of a big amount of data possible already under short time. In order to avoid data collisions in the computer networks, the one-way communication is forced always along the network. Naturally, two-way communication may
happen when dataflow is separated in time on the same medium. This means when a message splits into the packets to transfer them through the medium, in the same time only one data bit of the split message is crossed the transfer medium. For example, the message may represent a very long train crossing on an extraordinary short railway rail. Therefore, while the backside of the train still stays on the starting railway station and the engine has already arrived to the destination station only one railway carriage stays always on the examined railway rail. Similarly, although the network model's graph may imply one-way directed edges, in most case these mediums are used as two-ways channels since their communications are divided in time.

3.3. The communication
Communication between the nodes of the computer network is always dynamic. Peak hours are usual between the nodes, but no broadcast periods also exist. The temporal changes of the dynamic message transmissions are modelled with the help of a communicational matrix. The task of the communicational matrix is to take into consideration all elements of the network as well as their features, which regulate the data traffic between the nodes. These rules prescribe the conditions for the usage of the data transfer between the nodes and the data transfer medium. During the communication of the network's active elements, the model examines full connection system of the network. To the faultless simulation of the communication, the model has to imply some important parameters yet namely, the lengths and numbers of data transfer sections, the physical parameters of the medium, the sizes of storage units in the nodes, the degree of the utilisation, etc.

3.4. Some special characteristics of the computer network in case of faultless data transfer

- In the course of data transfer, although transfer velocity of one data bit is constant in time, transmission times of packets with the same longitude may be different.
- In the course of data transfer, the data transfer sections running in parallel with each other do not affect each other directly, but in the nodes, for example the appearance of multiplied packets makes disturbing effects.
- Two-way traffic does not exist.
- The intensity of inner communication changes in time between the nodes directly connected to each other. In case of wrongly chosen parameters for example this inner communication capable to create peak-load on the examined network without traffic arriving from the external network.
- To control affecting message transmission, an inner communicational system works between the nodes connected to each other. For example, the receiver can receive a message vainly if the transmitter does not have a message to be sent on.

4. Traffic simulation model
Transactions in computer networks show non-linear features. For example, sometimes no data transfer happens between two nodes, because buffer of receiver node is full, though sender node is able to transfer data. It means no tools to describe transactions by mathematics of linear control systems. In this paper, a non-linear traffic behaviour model is set up. By the aid of this model, the non-linear features of traffic can be described. We also present a method to recognise congestions.

The computer network models known by the literature trace the data traffic back to the description of a communication happening between edges and nodes of network. The models handle the nodes as important elements in this description method. The outcome of a model is a communication graph that faithfully imitates the physical arrangement of the computer network, where the nodes, the active elements of the system are the peaks of the graph, which are connected to each other by the transfer mediums called edges. This statement is important, because the nodes, the active elements of the network communicate with each other, and the nodes form the peaks of the network graph in the graph
theory model of the computer network. Figure 1 shows a plain example how nodes communicate to
each other.

This description gives back that view on a natural manner. In this model the central place are
occupied by nodes as the peaks of the network graph and the edges of the network graph show
transactions of the traffic in which the peaks communicate with each other along the data lines
connecting them.

This process separates our internal nodes, where switches or bridges are located with a closed curve
from their input and output workstation nodes, as it is shown in figure 1. In the following the internal
nodes are numbered, input nodes are marked by small letters and output nodes are marked by capitals.
Consider the network is known in a time \( t \) and numbers of bits stored in the nodes are marked with
\( N(t) \). Each node has some buffer for their messages. This capacity is measured by data density (1) that
shows rate of recent data quantity in time and the maximal data quantity in node \( i \).

\[
b_i(t) = \frac{\text{number of recent databits in buffer of node } i}{\text{buffer size of node } i} = \frac{N_i(t)}{A_i} \quad (1)
\]

In our model the data density of node \( i \) is a number without dimension between \( 0 \leq b_i \leq 1 \). Traditionally, the numeration of data density shows the difference of outgoing and incoming bits in
time \( t \) or in other words it can show the maximum number of data bits that can be transferred in the
next time unit. Now data density is introduced as a rate of the length of messages stored in node \( i \) and
the maximal message length. This property shows non-linearity, because no chance is to receive more
bits than the maximal size of the data buffer or node \( i \) can not receive any bit, if node \( j \) has no data to
send. In our network data density of node \( i \) is marked with \( b_i \). In our model \( N(t) \) and \( b(t) \) are vectors of
\( n \) elements (2).

\[
N_i(t) = A_i \cdot b_i(t) \quad (2)
\]

shows the amount of data stored in node \( i \), where \( A_i \) represents the buffer size of node \( i \). Since no
buffer is between node \( i \) and node \( j \), \( A \) is a diagonal matrix (3), that is

![Figure 1. Internal and external elements of a network measured and simulated.](image)
\[ \mathbf{N}(t) = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_n \end{bmatrix} \begin{bmatrix} b_1(t) \\ b_2(t) \\ \vdots \\ b_n(t) \end{bmatrix} = \mathbf{A} \cdot \mathbf{b}(t) \] (3)

If elements of \( \mathbf{N}(t) \) are known, total number of databits stored in the observed area is calculated (4) as a vector scalar product:

\[ N(t) = \sum_{i=1}^{n} A_i \cdot b_i(t) = \text{diag}(\mathbf{A}) \cdot \mathbf{b}(t) \] (4)

Now let us examine status of the network in time \( t+\Delta t \). The amount of data stored in node \( i \) changes as (5):

\[ N_i(t+\Delta t) = N_i(t) + \Delta N_i^\text{internal} + \Delta N_i^\text{input} - \Delta N_i^\text{output} \] (5)

In the network the transmission speed means how many bits can be transported in one second between two nodes. Transmission speed may be different in some parts of the network, but from node \( j \) to node \( i \) the same value is supposed and marked with \( v_{ij} \). If transfer speeds are known between the nodes, data changes can be expressed as (6)

\[ \Delta N_i^\text{internal} = \sum_{j=1 \atop j \neq i}^{n} c_{ij} \cdot v_{ij} \cdot \Delta t, \quad \Delta N_i^\text{input} = \sum_{l=1}^{k} c_{il} \cdot v_{il} \cdot \Delta t, \quad \Delta N_i^\text{output} = \sum_{o=1}^{M} c_{oi} \cdot v_{oi} \cdot \Delta t. \] (6)

where \( c \) is communication function of the node. The communication function describes the quality of transmissions. Indexes of the communication function show the direction of the data transfer. The first index shows the receiver node and the second one represents the sender node. Input and output nodes are marked with I and O respectively. After summarizing data changes in all nodes, we get total data change (7) in our network:

\[
\begin{bmatrix}
N_1(t+\Delta t) \\
N_2(t+\Delta t) \\
\vdots \\
N_n(t+\Delta t)
\end{bmatrix} =
\begin{bmatrix}
N_1(t) \\
N_2(t) \\
\vdots \\
N_n(t)
\end{bmatrix} +
\begin{bmatrix}
\sum_{j=1 \atop j \neq i}^{n} c_{ij} \cdot v_{ij} + \sum_{l=1}^{k} c_{il} \cdot v_{il} - \sum_{o=1}^{M} c_{oi} \cdot v_{oi} \\
\sum_{j=1 \atop j \neq i}^{n} c_{2j} \cdot v_{2j} + \sum_{l=1}^{k} c_{2l} \cdot v_{2l} - \sum_{o=1}^{M} c_{2o} \cdot v_{2o} \\
\vdots \\
\sum_{j=1 \atop j \neq i}^{n} c_{nj} \cdot v_{nj} + \sum_{l=1}^{k} c_{nl} \cdot v_{nl} - \sum_{o=1}^{M} c_{no} \cdot v_{no}
\end{bmatrix} \cdot \Delta t
\] (7)

Forming (7) and do \( \Delta t \to 0 \), the result is shown by (8):

\[
\lim_{\Delta t \to 0} \frac{N(t+\Delta t) - N(t)}{\Delta t} = N'(t) = \sum_{i=1}^{n} \left( \sum_{j=1 \atop j \neq i}^{n} c_{ij} \cdot v_{ij} + \sum_{l=1}^{k} c_{il} \cdot v_{il} - \sum_{o=1}^{M} c_{oi} \cdot v_{oi} \right)
\] (8)
(8) says to add up all bits in transactions. In the observed area between two nodes a transaction transports the same amount of data. Only their signs are different, that is the total amount of data transported is equal to zero. (8) becomes more simple (9).

\[
N'(t) = \sum_{i=1}^{n} \left( \sum_{l=1}^{k} c_{il} \cdot v_{il} - \sum_{O=1}^{m} c_{Oi} \cdot v_{Oi} \right) = \sum_{i=1}^{n} \sum_{l=1}^{k} c_{il} \cdot v_{il} - \sum_{O=1}^{m} \sum_{i=1}^{n} c_{Oi} \cdot v_{Oi} \tag{9}
\]

(9) represents that the change of numbers of databits independents of change of internal dataflow in the observed area. (9) shows the difference of numbers of incoming and outgoing databits in a time unit or in other words (10)

\[
\int_{t}^{t+\Delta t} N'(t) \cdot dt = \left( \sum_{i=1}^{n} \sum_{l=1}^{k} c_{il} \cdot v_{il} - \sum_{O=1}^{m} \sum_{i=1}^{n} c_{Oi} \cdot v_{Oi} \right) \cdot \Delta t = \text{Inputs} - \text{Outputs} \tag{10}
\]

5. Communication function

The communication function plays big role in (6) – (10). There are some subfunctions implemented in \(c_{ij}\). As we could see before \(c_{ij} = c_{ij}(b_i, b_j, t)\). Let us examine parts of this function. \(C_{ij}(b_i, b_j, t)\) must contain at least the following:

- Internal connections of the network.
- Environmental connections of the network.
- Size of internal buffers in nodes.
- Capabilities of nodes to send and receive data.

The subfunction of internal or environmental connection, marked with \(k\) grants the connection when node \(j\) can communicate with node \(i\). The connection subfunction is product of two components. This function is created node by node. Each node is examined and if a physical connection exists between nodes \(j\) and node \(i\), \(f_{ij}\) is set to 1, otherwise it is zero (11). Elements of \(f_{ij}\) creates a symmetrical matrix, because if node \(i\) is connected to node \(j\), than node \(j\) is connected to node \(i\), too. If there is the opportunity of the permanent communication between two nodes, and the node \(j\) works for node \(i\), then \(k_{ij}(t) = f_{ij} \cdot p_{ji}(t)\), where \(p_{ji}(t)\) is the probability of data transmission from node \(j\) to node \(i\). The probability of data transmission, \(p_{ji}(t)\) presents the distribution proportion of given routes belonging to a node with relative weighting, where \(0 < p_{ji}(t) < 1\), if node \(j\) works for more than one node and \(p_{ji}(t)\) may differ from \(p_{ji}(t)\). The relative weightings of node \(j\) are shown in column \(j\) of the matrix \(k\) and \(\Sigma p_{ji}(t) = 1\) for each column. If there is no physical connection between these nodes, then \(k_{ij} = 0\).

\[
k_{ij} = \begin{cases} 
  f_{ij} \cdot p_{ji}, & \text{if connection exists between nodes } j \text{ and node } i \\
  0, & \text{if no connection is between nodes } j \text{ and node } i
\end{cases} \tag{11}
\]

where \(i \neq j\) and \(1 \leq ij \leq n\).

As we could see before the size of the buffers is included in data density of the node.

The third subfunction presents the properties of the connection. We know that data link mechanisms depend on density of the traffic. In our model, two internal subfunctions are used. \(S(t)\) defines properties of sender node, while \(R(t)\) declares features of the receiver.
$S_j(t)$ is an internal subfunction (12) of transmitter node with values of 1 or 0. It shows whether node $j$ has any message to send or not. The connection is disable if data density of node $j$ ($b_j(t)$) is equal to 0, anyway 1.

$$S_j(t) = \begin{cases} 1, & b_j(t) > 0 \\ 0, & b_j(t) = 0 \end{cases}$$

(12)

$R_i(t)$ is another internal subfunction (13) of the receiver node with values of 1 or 0. The connection is enable if $b_i(t)$, the data density of node $i$ smaller, than 1, anyway it is 0. Zero means that buffer of node $i$ has been overloaded, so node $i$ closes its communication port to direction of node $j$, therefore node $i$ does not receive any message from node $j$.

$$R_i(t) = \begin{cases} 1, & b_i(t) < 1 \\ 0, & b_i(t) = 1 \end{cases}$$

(13)

Finally, the communication function between node $j$ and node $i$ is a product of three subfunctions (14): $$c_{ij}(t) = k_{ij}(t) \cdot S_j(b_j(t), t) \cdot R_i(b_i(t), t)$$

(14)

Suppose that dataflow is working normally. Input nodes can always send data to the internal nodes, no any trouble is among the internal nodes and output nodes can receive all data from our network. In this case both $S_j(b_j,t)$ and $R_i(b_n,t)$ are set to 1, that is $v_{ij} \Delta t \leq (1-b_i(t)) \cdot A_i$ as well as $v_{ij} \Delta t \leq b_j(t) \cdot A_j$ and the average transfer speed for the total time period is $e_{ij}(t) \cdot v_{ij}$, where $e_{ij}(t)$ is the effectiveness of the transmission channel that shows the time rate of real data transfers and the total working time.

6. Simulation and results

6.1. Test environment

An existing network which can be seen in Figure 1 was measured and simulated at transfer speed of 100 Mb/sec. The first file of 89 MB was transported from Inpa and the second one of 91 MB started from Inpb. First channel used TCP protocol, while second file was transported by FTP protocol. The second file was started in 8 seconds after the first one. The third input channel named Inpc did not work. The result was measured at channel of OutA. Measured figures were observed by Wireshark. (Wireshark is registered trademarks of the Wireshark Foundation.) The transfer was repeated 60 times. The calculated efficiency rate was 72% in the 100 Mb/sec channels, because an average transfer took 20 sec.

6.2. Simulation unit

The simulation takes too much time if dataflow is simulated by bit by bit. We wanted to make a unit for our simulation. First the lengths of the frames were measured. Data transfers were made in different time period (early morning, afternoon, midnight), with different data lengths (from some hundred bytes to hundred megabytes) and different transmission speed (from 24 Kbits/s up to 100 Mbits/s). Measurements were made by the Wireshark Network Protocol Analyzer. After having collected more hundred thousands messages, the incoming and the outgoing frames were analyzed. The result showed two significant figures among the values. One of them represented the length of confirmation messages, while the other one was the typical length of data messages. The most frequented values are shown in Table 1. Taking into consideration figures of Table 1, we chose 55 bytes as unit of frame length. Using this frame length an average data length is 26 units.
6.3. Simulation

The simulation was written in MatLab. The same files were transferred through our model like we used in the real network. We are mainly interested in transfer systems of large number of messages. Our work is motivated by the following questions:

- How to implement update rules?
- How to implement update rules in different systems (i.e.: mono or multiplied channels)?
- How to reduce or avoid collision?
- How to measure effectiveness?

Main features of our update rules in our system are the following:

- No multichannel mode is implemented in nodes.
- The effects of collisions take into consideration in efficiency rate.
- Messages always move forwards or stay in nodes.

Our system uses parallel update rule. The internal nodes are numbered from 1 to \( n \) while input nodes are numbered from 1 to \( m \). At each time step, each node is updated at least once. Each time step is divided into three stages. Update is performed as follows:

- **Stage 1**: If there is at least message in a node to transfer its move is based on communication function (transfer probabilities as well as own and neighbour status).
- **Stage 2**: The conflicts of transfers are resolved. The conflicts arise because more than one messages want to move to the same node at the same time step. If at least two messages want to move to the same node, one message is randomly chosen to move to that node and the other conflicting messages remain at their current node during the current time step.
- **Stage 3**: the actual message transfer bases on the transfer decisions and conflict resolutions of the first two stages. These three stages are used to perform parallel update at each time step.

Initially, the messages are located in their nodes with density \( h_i \) where \( 0 \leq h_i \leq 1 \) and \( 1 \leq i \leq n \) or \( a \leq i \leq m \). Each transfer medium can be occupied by at most one message at any time. The data density of the internal nodes is measured and calculated because the messages are either added or removed after the initialization of the network, so the densities of the nodes change in time. The update rules for nodes and messages are the following:

- The transfer probabilities depend on the probability of data transmission from node \( j \) to node \( i \) (\( p_{ij}(t) \)).
- Messages can move forward but not backwards.
- Using different internal buffers, bottleneck effects can also be tested.

### Table 1. Number of pieces of the most frequented frame length [in bytes].

| Length | Pieces  | Total length |
|--------|---------|--------------|
| 56     | 165 535 | 8 938 890    |
| 66     | 11 773  | 777 018      |
| 1082   | 10 730  | 11 609 860   |
| 1434   | 271 383 | 389 163 222  |
| 1514   | 43 127  | 65 294 278   |
6.4. Results

Figure 2 and figure 3 show two simulations with different buffer sizes. In figure 2 5 MByte internal buffer was used, while in figure 3 only 1 MByte buffer was allocated. In Figure 2 inputs work linearly, while figure 3 shows non-linear elements. Figure 3 helps us to find bottlenecks of the network. Inpb was blocked by node Int1, because there were no enough room for incoming frames.

![Figure 2. Data transfer using 5 MB internal buffers.](image1)

![Figure 3. Data transfer using 1 MB internal buffers.](image2)
7. Conclusion and future work
This paper looks for possibilities to describe dataflow model of large-scale computer networks. A model was presented that was applicable to simulation, planning and regulation of computer network’s traffic. At the time of the model’s establishment, the partial differential equations were avoided in the mathematical model because of the specially chosen state variables. The nodes have honoured roles in this non-linear model because storage capacities of the transmission medium are practically zero. Nodes either communicate to each other or not. In our model, the mean of the data density is the proportion of the size of data stored in the single node and the data quantity, which can be stored maximally. Our model examines change of data density occurred by data flow among the nodes in a region demarcated by close curve. Input and output data densities are regarded as known. At first sight, these processes are the inputs and outputs of the model. Effectively, these processes together form the actual inputs of the mathematical model. State variables present data densities arising in the internal nodes of the system. Our system applies a data traffic model involving n internal and m external nodes. To create the mathematical model the communication matrices defining the network has fundamental importance. Our model applies four communication matrices. During simulation we supposed that speed of transmission equals between nodes. It must be changed in the future. We have to work out a quick test to calculate possible congestion points. Finally, a simply example was presented to demonstrate effectiveness of the model and how this model is applicable to the simulation, planning or regulation of computer networks.

References
[1] Bhole Y and Popescu A 2005 Measurement and Analysis of HTTP Traffic Journal of Network and Systems Management 13 357-371
[2] Jiang J and Papavassiliou S 2004 Detecting Network Attacks in the Internet via Statistical Network Traffic Normality Prediction Journal of Network and Systems Management 12 51-72
[3] Sidiropoulos S, Giovanidis G and Keromytis A 2005 A Dynamic Mechanism for Recovering from Buffer Overflow Attacks ISC 2005, LNCS 3650 (J. Zhou et al.) 1–15
[4] Paxson V and Floyd S 1995 Wide area traffic: The failure of Poisson modelling IEEE/ACM Transactions on Networking 3 226–244
[5] Haifeng D, Yang X and Lingyun L 2008 An Effective Network Congestion Control Method For Multilayer Network Journal Of Electronics (China) 25
[6] Change J and Sirbu M 2001 Pricing Multicast Communication: A Cost-Based Approach, Telecommunication Systems 17 281–297