Computer Networks as Complex Systems in Nonextensive Approach

Franciszek Grabowski, Andrzej Paszkiewicz, Marek Bolanowski

Rzeszow University of Technology,
Department of Distributed Systems
al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland
fgrab@prz.edu.pl
andrzejp@prz.edu.pl
marekb@prz.edu.pl

Abstract. Contemporary computer networks fit into the category of complex systems. However, many models still are based on an idealistic paradigm of simple systems based on thermodynamic equilibrium. In order to understand and optimize the design of real networks it is necessary to understand and describe their nonextensive operation aspects. Therefore, it is necessary to explain such concepts as self-organization, self-adaptation and matching load to finite resources.

Keywords: complex systems, computer networks, nonextensive.

1. Introduction

The approach to the concept of complex systems is varied. However, at the beginning we should distinguish natural systems from artificial systems. Natural systems are complex systems by definition, and a human just tries to understand inner processes, phenomena and mechanisms. Artificial systems, man-made are based solely on the model adopted by him. In many cases, these models more or less imitate natural systems. Natural systems = complex systems are characterized
by the fact that three operating states are permitted: underload, state of equilibrium and overload. In the state of underload the system is not making full use of its resources, and thus their potential. At equilibrium, the system is pushed to its limits, and is a state of optimal use of resources. However, very often systems are also in overload condition, e.g. as a result of infection, excessive load, etc. This condition is normal for natural systems. Of course, depending on the predisposition of the system and the amount of the overload state is longer or shorter acceptable. It should be also noted that natural systems can run in this condition adaptive mechanisms in order to adapt to the situation and return to equilibrium. Artificial systems, including computer networks are treated as simple systems, assuming that there are only two acceptable states: underload and equilibrium. The adoption of such assumptions results in that such systems cannot work effectively in the overload condition, which causes the degradation of the system.

The result of reflection in recent years is the assertion that real systems, including artificial systems such as computer networks that inscribe in a formula of three operation states and overload state should be treated as a normal operating state. In this paper, we assume that each of the three states of the work is acceptable and we are trying to take this into account in the adopted model. It should be also noted that computer networks are operating in a thermodynamic disequilibrium. We describe the state drawing on the tools nonextensive thermodynamics.

Previous research on computer networks have mainly focused on technical consideration especially in underload conditions. Computer networks can be regarded as distributed systems in the idealistic approach. George Coulouris says that the distributed system is a system whose hardware and software components are allocated in network nodes, they communicate and coordinate their actions only by sending messages [1]. The immanent features of this system are concurrency, the lack of global synchronization and independence of the failure of an individual system’s components. This approach also fits in with the concept of computer networks as fractal structures only in geometrical terms [2] (Figure 1a.).

This definition of the distributed system allows to regard it as a simple system of the following characteristics: algorithmic processing, linearity characteristics as a function of load, thermodynamic equilibrium, stationarity of processes, ergodicity, laminar deterministic flow, lack of overload and collapse of the system, static capacity planning, system resources independence processes, the normal distribution in time and space, additive statistics, guaranteed quality of service, etc. Regarding individual components as simple elements in the case of distributed systems is a great simplification.
Computer networks are often regarded as complex systems consisted of many components. However, these elements carry out specific tasks that are defined and described by simple relationships. Example of such relationships is the process of routing, local QoS, load balancing, etc. In case of appearing of any problems resulting from the number of elements (e.g. nodes), the number of parameters describing relationships (sometimes characterized by dynamic changes), then we can see that the common practice was and still is the archaic activity where the task is splitting into smaller ones. In this way a simple system is obtained and well-known laws and existing computational models can be applied. Such actions are commonly used for problems of designing of large computer networks such as elements allocation, creating topology or modeling mechanisms within theme, for example the selection of communication channels.

Modern computer networks fit into a category of social-economic-technological systems where processing is interactive and self-organizing, and the main features are: non-linearity, thermodynamic disequilibrium, not extensiveness, non-stationarity of processes, sensitivity to congestion and infection, emergence, dynamic load balancing, scale free processes, fractional Brownian motion, exponential law, small worlds, preferential attach, nonextensive statistics, the lack of guaranteed quality of service, etc. These features enables us to treat real computer networks as complex systems. Furthermore, it appears that the working area of computer networks oscillates between a laminar and a turbulent flow, that is on the border of chaos [3, 4]. The complex system, totally different from the simple system, is an emergent structure of mutually interacting elements. Processes taking
place in this system are scale free, form short-term to long-term ones.

So, if we want to talk about fractal terms of computer networks, it should be considered from the perspective of fractal nonextensiveness processes occurring in it. Such nonextensiveness is directly related to the ratio of the feedback gain \([5]\).

### 2. Computer networks as a complex system

During analyzing networks as complex systems, we should refer to the basis of these systems. The precursor of this theory was Ludwig von Bertalanffy \([6]\), which is considered the father of systems theory. Originally this theory was related to biology. However, these works were soon developed by different researchers in various fields such as cybernetics and engineering (hence the engineering system). Over time, various strands of science began to penetrate and complement each other. The social sciences such as sociology and economics had active participation in this \([7]\). At the same time new theories occurred and slowly formed foundations for complex systems. At that moment, the fractal geometry and chaos theory should be mentioned \([8]\), that were introduced and disseminated by Benoit Mandelbrot and Edward Lorenz. These papers were one of the pillars of dynamic systems theory. On the other hand, that papers concerning self-organization \([9]\) emergence \([10]\), autopoiesis systems \([12]\) occurred. They led consequently to the development of dynamic systems both technical and human resources. On that foundation the New Science of Networks appeared. The basis of this science includes the theory of small worlds \([11]\), scale-free networks, power law, etc.

A characteristic feature of the small worlds theory is that the number of intermediaries between any pair of nodes in the network is relatively small. Typically, it corresponds to the sixth degree of a separation. Despite the fact that the value characterizing the distance between any two vertices is small, these networks are characterized by a high degree of clustering. Watts and Strogats proposed a minimal model of creating small worlds in networks. The model proposed by them can be used in designing computer networks and network communities. The location of small worlds in the context of a relationship randomness between nodes is shown in Figure 1b., where \(p\) is the probability of connections randomness. In fact, regular networks are unheard and hence their importance for modeling of computer networks is small, or even negligible. On the other hand, fully random network shows no structural arrangement which also does not correspond to real
relationships between nodes. Therefore, the intermediate state corresponding to small worlds refers to real network structures.

Barabási and others have shown that many real networks have scale-free degree distribution characterized by a Poisson distribution whose tail declines according to the power law [12]. The degree of distribution is often called the connectivity, and means the number of connections of one node to other nodes. On this basis, the probability of connection occurrence between the new node and each node belonging to the network depends a lot on the number of edges held by these nodes \( k \) and is:

\[
P(k) \sim k^{-\gamma}
\]  

While the exponent \( \gamma \) depends on the type of considered network. The consequence of these assumptions was the statement that most new nodes connect to nodes that already have most neighbors. These connections is called preferential connections (preferential attachment).

Another concept that is inherently associated with complex systems is **self-organization**. This term was introduced by Ashby in 1947 [13]. On this basis it can be concluded that the current state of the system is dependent on the previous state and the current input. However, it should be noted that natural self-organizing systems require a number of sophisticated information. While in the case of artificial systems such as computer networks, information is reduced intentionally or unintentionally to simplify the model. Immanent feature of self-organizing systems is the possibility of moving away from the desired state and thereby worsening their condition. Especially it can be shown in the case of absence of the adequate input information in complex systems. An example well illustrates this type of situation is both the process of routing and QoS in computer networks. In the case of routing, making decisions to change the routing path based on the current situation and only based on local information may contribute to the instability of logical topology or to improper use of available transmission resources. The same applies to use of QoS mechanisms. Typically, flow control mechanisms are only applied locally beyond statically defined threshold values and slightly take the previous situation into account. However, the processes occurring in computer networks are long-term. The adoption of a simplified approach described above can cause bottlenecks and actions are only short-term.

Additionally, until now processing in computer systems, including computer networks also have been implemented as in a simple system with almost unlimited resources. This assumption resulted from the adoption of the reductionist paradigm.
appropriate for these systems. Thus, the adoption of ideal processing space-time conditions resulting from the thermodynamic equilibrium meant the perfect processing. Such a system model is described by the equation proposed by Malthus [14] which determines system performance $X_M$:

$$X_M = \frac{dN}{dt} = rN,$$

(2)

where $r$ denotes increase parameter and $N$ the number of tasks in the system. Real computer networks cannot be considered in the context of the reductionist approach, where the system has infinite resources. Adoption of incorrect assumptions leads to the creation of inefficient interconnection structures, and implemented network mechanisms often do not cope with congestion. In the case of natural systems, both an under load condition and an overload condition is real and even acceptable.

However, it can be seen that Malthus equation is an integral member of logistic equation proposed by Verhulst. Logistic equation imposes condition of limited resources (i.e. the external conditions) on Malthus equation. Therefore, the system performance description as a whole it can be used as proposed by [4] general logistic equation for the family of complex systems including computer networks:

$$X = rN \left(1 - \frac{N}{K}\right)^f = X_M(1 - u)^f,$$

(3)

where: $K$ – means available resources, $N$ – refers to a set of current data transmission tasks, $u = N/K$, $r$ – is a fixed gain factor and $f$ is a parameter heterogeneous complex system of self-organization, that describes a trajectory $X(N)$ which are associated with internal sensitivity of the system.

Taking above statements into consideration, computer networks as self-organizing systems, but based on the principle of limitless resources may consequently lead to the degradation of such systems. The solution to this problem would be to move from self-organizing systems to self-adaptive systems. Figure 3. shows the dependence the level adaptation on an exploration level. In most cases, current computer systems, including computer networks based only on strictly defined data as input for algorithms. Therefore, the classical algorithmization of processes, tasks, etc. is associated with the low level of system adaptation to the existing conditions. The introduction of additional information helps to better understand and describe phenomena and processes, and thus increases adaptability of the system which is reflected in its appropriate level of self-organization.
Though, achieving the appropriate level of knowledge, and first of all intelligence allows to create a self-adaptive system. The level of knowledge and intelligence may be associated with the appropriate information processing in the context of micro-and macroscopic, maintaining appropriate short and long term relationships as well as. This approach will allow to draw appropriate conclusions, and thus enable the creation of mechanisms such as self-adaptive routing that will replace the current deterministic algorithmic approach (with short-term nature) in the future. Therefore computer networks should be considered as self-adaptive complex systems.

Figure 3 is a graphical representation of logistic equation of an arbitrary order (3), on which an ideal case was indicated determined by Malthus equation named I and characteristic work areas N and C, where N is accordingly the system work area defined as not critical and C as a critical area. In addition, U is a sub-area of under-load system, O is a sub-area of overload system, and M match points between limited resources of the system and realized task. Of course, the match point is mobile and moves on the green line. From this perspective, there is not only one the best solution, but it depends on the volatility of the operating conditions. Thus, self-adaptive system should/must frequently analyze the current situation and revise assumptions, parameters, thresholds and other factors to be able come back to
the match point M whenever deviation occurred.

The challenge for modern engineering of computer networks is to equip them with self-adaptation mechanisms, as is the case of natural systems, which leads to state where the finite resources and the load are balanced.

3. Nonextensive perspective for computer networks

Analyzing the computer network it cannot be assumed that parameters describing it are intensive or extensive. For example, network bandwidth is not usually equal in each of its elements, and is not the total capacity of the individual components. Additionally, computer network is a dynamic system in which changes occurred in various locations (also locally) have often a significant impact on the operation macroscopic scale (Figure 4a.). An example of this work is the dynamic routing.

However, modern computer science (including computer networks), like many other systems created by human, are based on the paradigm of thermodynamic
Figure 4. a. Macroscopic network approach; b. 3-element nonextensive structure.

equilibrium, which mean respectively that:

1. Only finite number of allowed states can occurred in the system and in its components.
2. All permitted states are equally probable.
3. System and its component efficiency is unlimited and its structure is closed.
4. Entropy is extensive,
5. Interactions among states have only a short-term character.

Thermodynamic equilibrium ignores the space-time constraints of real systems, which leads to the thermodynamic disequilibrium [4]. Many current system models, including computer networks, do not take into account basic conditions for the real system, i.e. unlimited states number, limited performance of components and the system as a whole, etc.

However, the thermodynamic disequilibrium leads to a nonextensive Tsallis entropy [15, 16], on which q-algebra is based on. One of the commonly used operation in this algebra is a q-sum operation (Figure 4b.), which gives the following expression in the case of a simplified example consisting of two components:

\[ S_{d1} = a \oplus_d b = a + b + (1 - q_d) ab, \quad (4) \]

where \( \oplus_d \) is a non-extensive sum operator, \(-\infty < q_d < +\infty\) determines the dynamics of the interaction between \( a \) and \( b \) any scale - from short-term to long-term. When
$q_d = 1$, then the structure is in a thermodynamic equilibrium (short processes) and thus contains only the data, so $a \oplus_d b = a + b$. When $q_d \neq 1$, then the structure is non-equilibrium and in addition to data contains also information specified by the free-scale processes $(1 - q_d)ab$. Effect of $q$-operation on processing in computer systems is presented by Grabowski [4].

Computer networks efficiency is described by many parameters and dependencies such as nodes capacity, the bandwidth of communication means, convergence time, performance of routing and forwarding, etc. However, it is always dependent on one or more of the following elements: hardware = executive system, protocols/algorithms and the structure of the transmitted data. Reducing view only to the hardware or just algorithms performance is part of the reductionist trend relating to simple systems. To consider the functioning of the network as a complex system any of these three elements cannot be skipped. Moreover, the influence of the various elements together should be also taken into account.

Using the $q$-sum operation, based on non-extensive operator $\oplus_d$, the sum of the real processing occurring in computer networks can be define as:

\[
\text{Processing} = \text{Executive system} \oplus_q \text{Protocols/Algorithms} \oplus_q \text{Data structure}
\]

This formula allows to assume that the interaction of all components gives you something more than expected from ordinary algebra. As a result, it can be said that all cooperating elements have influence on the process occurring in computer networks. Each of them individually, and all together as well as have influence on efficiency of its operation. For example, the data structure (thick or small granular) have a direct impact on the result of the specific algorithm, which is implemented in a given protocol. The computational complexity of the algorithm and its structure is important, but only together with the obtained input is a certain whole. In this way, you can just refer to the protocol both the transmission and control. From a wider perspective, the hardware structure cannot be omitted, which operation and cooperation of individual elements exposes capabilities of the executive system. Possessing a specific executive system with the protocol/algorithm (usually with variable data structure) we are talking about the real processing. Combining all above elements it may be only referred to the efficiency of the network and the real possibility of control mechanisms that occur in computer networks.

However it should be noted, that each of constituent elements (executive system, protocol, data structure) should not be treated as a simple system in itself.
Therefore, as a further consequence the real processing is obtained as expressed through:

\[
\text{Executive system (with limited resources)} \oplus_q \\text{Protocols/Algorithms (incomplete)} \oplus_q \text{Data structure (nonequilibrium)} = \text{Processing (self-deforming)}
\]

Resources limitations of the executive system was clearly explained before, and seems to be an obvious fact (e.g., limited bandwidth). Therefore, the processing in executive systems cannot be performed in accordance with adopted rules e.g. for Turing machine. Other important assumption is that the protocols/algorithms are incomplete. This statement is true in most cases because as a result of their action is not possible to achieve an exact solution. Additionally, obtained solution often not dispel doubts if the search direction is correct. Incompleteness of protocols/algorithms can be the result of limited resources, simplified models (e.g. limited number of decision criteria), incorrect input data (e.g. local perception of load balancing only from the perspective of a node), etc. The structure of data sent over the network is characterized by dynamic changes. Additionally, micro- and macroscopic relationships have influence on its statistical parameters. From this point of view, they are non-equilibrium nature. Resultant of adopted above assumption gives us the processing with self-deforming character in an ever-distortion. Such processing is a natural phenomenon. From this perspective, the state of complete balance and total chaos can be considered as special cases, like the appropriate probability value equal to 0 and 1.

Network performance, just like of any other complex systems is dependent on a micro- and macroscopic relationship and directly depends on the efficiency of the executive system, implemented protocols/algorithms and the data structure. At the same time, the protocol/algorithm can be treated as a software performance and data structure has an influence on both the hardware and software performance. Referring to the definition of complex systems and their nested properties the network performance can be represented as (Figure 5.):

Defining system performance as \( X = \{X_S, X_H\} \), where \( X_S \) denotes software performance and \( X_H \) is hardware performance we can assume that the sum of the
4. Conclusion

The basic to understand the self-adaptive systems is to understand the self-organization mechanism. Without this understanding, it is not possible to design self-adaptation mechanisms properly. The convergence and scalability of computer networks coupled with the load mismatch to the available resources, leads to the loss of control, which is usually revealed in degradation of response time and, ultimately, degradation of the quality of services.

This paper proposes a departure from modelling of real computer networks as simple systems. To understand the functioning of computer networks as nonextensive complex systems it is necessary to clarify such concepts as self-organization, self-adaptation and matching the load to the finite resources. This paper presents the explanation of these issues.

Operation of the network was presented in nonextensive approach in the further part of the paper. This approach seems to close reality. Of course, the proposed
approach is only a prelude to further works. Probably the complete nonextensive network model cannot be immediately implement. However, any efforts should be made that will set us closer to this goal.

References


