Mitigating Systemic Risks in Future Networks

Antonio Manzalini, *Telecom Italia, Strategy - Future Centre / Innovative Architecture*
Noel Crespi, *Institut Mines-Telecom, Telecom SudParis*

***Abstract*— This paper elaborates about the potential risks of systemic instabilities in future networks and proposes a methodology to control and mitigate them. The starting concept is modeling the network as a complex environment (e.g. ecosystem) of resources and associated controllers in a continuous and dynamic game of cooperation and competition. Key observation is that Internet might be viewed as "self-organizing" and that this is achieved through "constrained optimizations" (through protocols methods, algorithms, etc). The methodology foresees defining and associating utility functions to these controllers and elaborating a global utility function (as an aggregated function of local utilities) for the overall network. It is conjectured that the optimization of proper global utility functions ensures network optimization and stability at the same time.**

***Index Terms*—Stability, Network of Networks, Cloud Computing, Self-Organization, Self-Governance**

I. INTRODUCTION

TECHNOLOGY trends and socio-economic drivers are steering the evolution of networks towards a connectivity fabric capable of interconnecting huge numbers of interacting heterogeneous nodes. One can easily imagine a scenario in the near future where virtual links are dynamically created and destroyed by applications and services to produce a very dense, interconnected environment of processing and storage resources, sensors, actuators, machines, etc.

This scenario will be the socio-economic arena of multiple Players (e.g. Network and Service Providers, Over-The –Top, Enterprises, etc.) interacting with each other as in a natural ecosystem, providing any sort of services and data.

This evolution will require that an intelligence (managing and controlling network resources and their services) to be exploited through sets of protocols and controllers (interacting each other and properly orchestrated), performing certain levels of automation in Operations (to ease human operation and mitigate mistakes). These controllers could implement, for example, methods algorithms and control-loops.

In this evolutionary scenario, in order to achieve acceptable performance, hardware is not the critical point (e.g. line-cards with very high speed interconnected general purpose CPUs provide already a sound basis), but Industrial Mathematics tools and software are the real challenges. Software Defined Networking (SDN) can be seen as a step in the direction of this network transformation.

In SDN architecture, control and data planes are decoupled, and so network infrastructure is abstracted from business applications. This is expected to bring about greater programmability and the flexibility to build multiple networks (on the same physical infrastructure) offering multiple network services. Network services, for example, will include routing, security, access control, bandwidth management, traffic engineering, processor and storage optimization and all forms of policy management, custom tailored to meet business objectives.

Future networks will therefore rely more and more on software, which will accelerate the pace of innovation as it has done in the computing and storage domains. A network will look like a complex system consisting of many diverse and autonomous units, but with interrelated software and hardware components. As known, complex systems cannot be easily described by rules and their characteristics are not reducible to just one level of description. In fact, complex systems exhibit properties (e.g. self-organization) that emerge from the interaction of their parts and which cannot be predicted from the properties of the single parts.

This means that the increasing level of complexity in future networks will bring with it new unexpected management challenges and systemic risks. In particular, it is argued that the level of complexity will be soon comparable with the one experienced today in the financial trading market (whose dynamics come from the intertwining of human operations and automated trading systems [1]).

Let's consider the example of the banking ecosystem: [2] takes the metaphor of a natural ecosystem as an assembly of species: each of which has feedback mechanisms that could ensure the population's stability if acting alone. However, the assembly as a whole may show sharps transition from overall stability to instability as the number and strength of interactions among species increase. Sharps transitions and instabilities in a financial ecosystems could have the same cause: intertwining of large numbers of feedback mechanisms related to the interactions of several Players with automatic trading machineries. [1] elaborates on the abrupt system-wide transitions and crashes that may occur out of the spontaneous mix of human and rapid control machine interactions.

Coming back to networks, it is widely recognized that they are strategic assets; in this sense, it is of paramount importance to mitigate the risk of these stability transitions, whose primary effects might be jeopardizing performance or, in the worst case, of creating a meltdown of a portion of the network.

Today Internet might be viewed as "self-organizing" [3]: practically this is achieved through "constrained optimizations" (through protocols methods, algorithms, etc).

This paper proposes modeling a future network as a complex ensemble (e.g. an ecosystem) of resources and controllers in a continuous and dynamic game of cooperation

and competition pursuing "constrained optimizations". Utility functions can be associated with these controllers and a global utility function (in turn a function of the controllers' utility functions) can be associated to a network. Then considering said the network as a system with a specific topology, the question arises as to whether one can define an optimal control problem whose solution leads to a distributed controller solving the utility maximization problem end ensuring stability at the same time. This paper conjectures that this is possible. A first step is defining what is meant by network stability. In the prior-art there are several definition, but all of them are specific to the context (e.g. session, routing, congestion control, etc).

An abstract definition of network stability (of larger validity) requires introducing the concept of network state, which is defined by a vector of data (which are relevant network parameters, e.g. QoS, etc) characterizing the state of the network upon a certain set of configurations. Imagine a phase space (with dimensions of said vector) which represents the network behavior in terms of states trajectories changing over time: this phase space has areas where network states have to stay, and other areas where network states don't have to reach. A network is stable when its states stay in the designed areas of said space. Ensuring stability means avoiding abrupt changes of the network states, specifically when moving states into "forbidden areas". This an abstract definition which will be useful in the following discussions.

II. Example of Network Instabilities And Prior-Art

The risks of instabilities are already present in today's network and cloud infrastructures. This section presents a brief overview, including some examples and a brief literature review.

In a generic communication network, the instability of an end-to-end path is a cross-layer issue; in fact, it might depend on the unwanted combination of diverse control mechanisms acting on either the underlying transport network or on the higher layers' components (e.g. flow admission control, TCP congestion control and dynamic routing).

The main arguments for introducing and enhancing flow admission control are essentially derived from the observation that a network otherwise behaves in an inefficient and potentially unstable manner. In fact, even with resource over-provisioning, a network without an efficient flow admission control has instability regions that can even lead to congestion collapse in certain configurations.

Congestion control is another area with undesired instability. Currently available mechanisms (like TCP Reno and Vegas) are examples of large distributed control loops designed to ensure stable congestion control of resources. On the other hand, these mechanisms will be ill-suited, from a stability viewpoint, for future dynamic networks where transients and capacity will potentially be much larger.

A further example is the instability risk in any dynamically adaptive routing system. Routing instability, which can be (informally) defined as the quick change of network reachability and topology information, has a number of possible origins, including problems with connections, router failures, high levels of congestion, software configuration errors, transient physical and data link problems, and software bugs.

In [4] a simple model of traditional network traffic dynamics is presented. It shows that a phase transition point appears, separating the low-traffic phase (with no congestion) from the congestion phase as the packet creation rate increases. In [5], the previous model has been improved by relaxing the network topology using a random location of routers. This enhanced model has exhibited nontrivial scaling properties close to the critical point, which reproduce some of the observed real Internet features. The authors in [6] discuss the possibility of phase transitions and meta-stability in various types of complex communication networks as well as the implication of these phenomena for network performance evaluation and control. Specific cases include connection-oriented networks with dynamic routing, TCP/IP networks under random flow arrivals/departures, and multiservice wireless cellular networks. Reference [7] presents an investigation of the dynamics of traffic over scale-free networks. A series of routing of data packets are proposed, including the local routing strategy, the next-nearest-neighbor routing strategy, and the mixed routing strategy based on local static and dynamic information. The results have indicated the existence of the bi-stable state in traffic dynamics; specifically, the capacity of the network has been quantified by the phase transition from a free flow state to a congestion state.

Paper [8] has addressed the risk of instabilities in Cloud Computing infrastructures. That study points out some analogies of Cloud Computing infrastructures and complex systems and elaborates on the emergence of instabilities due to the unwanted coupling of several reactive controllers.

As prior-art (in the use of utility functions in network design and Operations), we've surveyed the work about questions, results, and methodologies in the emerging theory of achieving stability in network utility maximization.

In [9] and [10], Kelly et al. presented an innovative idea of formulating a network optimization problem in terms of maximizing an utility function where the variables are the source rates constrained by link capacities and the objective function captures design goals.

Since then many research activities have been carried out on distributed network resource allocation using the language of Network Utility Maximization (NUM). For example, also cross-layer interactions can be characterized by viewing the process of "layering as decomposition of a given NUM problem into many sub-problems. These sub-problems are "combined together" by certain functions of the primal and dual variables. This framework of "layering as optimization decomposition" is well described in [11].

Utility functions can be constructed based on user behavior model, operator cost model, or traffic elasticity model. They can also shape the fairness of resource allocation [12].

Next section proposes a methodology to optimize and, at the same time, control stabilities in future networks.

III. A METHODOLOGY ADDRESSING NETWORK STABILITY

The approach starts by modeling the network as a complex ensemble (e.g. an ecosystem) of resources and controllers (partly centralized, partly distributed) in cooperation and competition. The controllers provide complete visibility and control over the network, ensuring access control, traffic engineering, quality of service, security, and other policies.

As an example, from an implementation viewpoint, these controllers can be seen as s/w components (performing distributed computations) pluggable in a lightweight middleware running on top of network equipment [13] (in the future, for certain applications, these controllers should migrate into h/w components to speed up the system).

In the direction of developing functional controllers, we can consider the approach reported in [14].

As an example, TCP/IP protocol can be seen as an example of an optimizer: its objective is to maximize the sum of source utilities (as functions of rates) with constraints on resources. In fact, each variant of congestion control protocol can be seen as a distributed algorithm maximizing a particular utility function. The exact shape of the utility function can be reverse engineered from the given protocol. Similarly, other recent results also show how to reverse engineer Border Gateway Protocols (BGPs) as a solution to the Stable Path Problem, and contention-based Medium Access Control (MAC) protocols as a game-theoretic selfish utility maximization [11].

This is in line with the NUM approach where a problem is decomposed and distributed algorithms can be developed where each of the controllers controls local variables, based on local observables, such as link load or path price. It should be noted that by techniques such as Lyapunov function or the descent lemma, global or local asymptotic convergence towards the optimum can be proved for these distributed algorithms.

Let us assume that we will exploit a number of functional controllers (i.e. proactive, reactive feedback control loops, methods, etc.) in a network, performing network features, or solving certain problems. The problem we wish to highlight is that instability can occur from the unintended coupling of independently developed controllers (like those in a complex system). We wish defining a methodology, based on utility functions, for solving this problem.

It is well-known that a utility can be seen as a value that represents the desirability of a particular state or set of configurations of its associated system. This is akin to saying that a utility function can be seen as a function mapping of the consequences of certain governance decisions into utility values. Therefore, to maximize a utility function $U(\cdot)$, means finding that configuration, Y_i , for which we get the maximum utility value: $u_i = U(Y_i)$.

The proposed methodology is based on a three- step approach:

- Decompose network problems: this is required to develop and exploit the required set of controllers in charge of handling network services (e.g. Congestion control, Dynamic routing, Scheduling, Load balancing, Resource Optimization, etc);
- Derive the controllers' utility functions: used to derive utility functions to be associated with the above controllers; each controller is seen as an optimizer whose objective is to maximize its utility, with the associated constraints [10]; and
- Define the network utility aggregated function: to derive the utility function to be associated with the network, which is an appropriate aggregation of the controllers' utility functions.

The task is to develop an optimization procedure to maximize the network utility function, and even better, to find those network utility values (corresponding to the controllers' configurations) that can achieve an overall network utility value above a certain threshold.

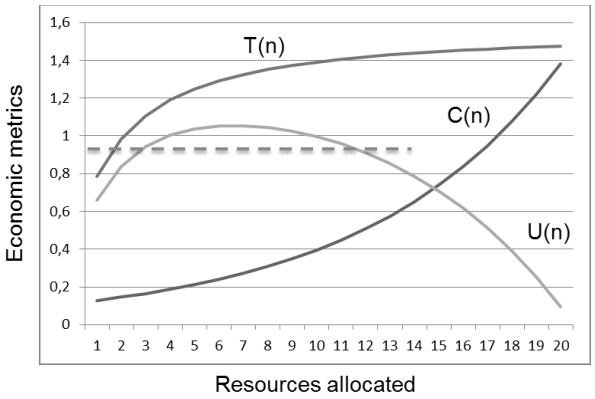

Fig. 1. Example: to achieve a stable trade-off for network performance

In many situations, practically, the scope is to achieve a stable trade-off for network performance; figure 1 shows a simple example where economic metrics (throughput/cost/utility) are functions of the resources allocated.

Let's consider two controllers: one in charge of optimizing the throughput of the network and the other that takes care of cost optimization. On one side, the higher the amount of allocated resources, the higher the network throughput, and so the utility function of the controller is T=T(n). On the other hand, the cost C=C(n) of the network is a function that monotonically increases with the amount of resources allocated. In this very simple example, the overall utility function can be written as U(n) = T(n) – C(n). The final task is keeping U(n) above a certain threshold (fixed, for example by the SLA).

A. Block Diagrams descriptions

Imagine a network with M controllers: each controller has a utility function $U_i(\cdot)$, in relation to certain performance metrics. The global utility function of the network is a function F of the utility functions of each controller

$$U_g(\cdot) = F(U_1(\cdot), \ldots, U_M(\cdot)) \quad (1)$$

This is equivalent to saying that the network has a global controller (whose utility function is $U_g(\cdot)$); a sort of orchestrator, which is in charge of configuring the M controllers to optimize the global utility function.

Maximizing a weighted sum of all utility values is one possible formulation. Other approaches may consider the maximization of an aggregation function in multiplicative

form, or multi-objective optimization to characterize the Pareto-optimal tradeoff between the controllers' objectives, or even game-theory.

Figures 2 and 3 show the blocks diagrams of a controller and a network controller, respectively. The controller in figure 2 has three main blocks: a monitoring function, a performance model and a utility function evaluator. In particular, the performance model allows specific performance metrics to be adopted (e.g. throughput as a function of load, traffic and number of resources allocated to a network). In figure 3, the combinatorial search block looks over the space of the possible configurations of the controller parameters. This could be done at regular intervals, upon reaching a trigger or in reaction to changes in the global utility function.

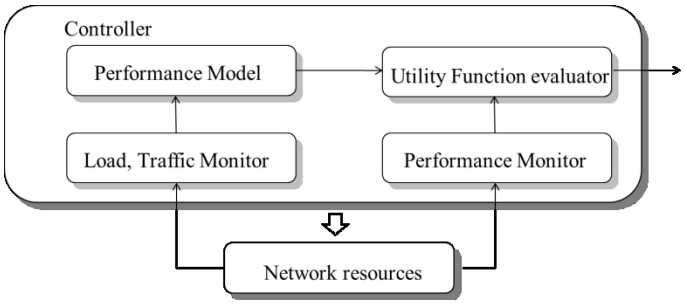

Fig. 2. Block diagram of a Controller

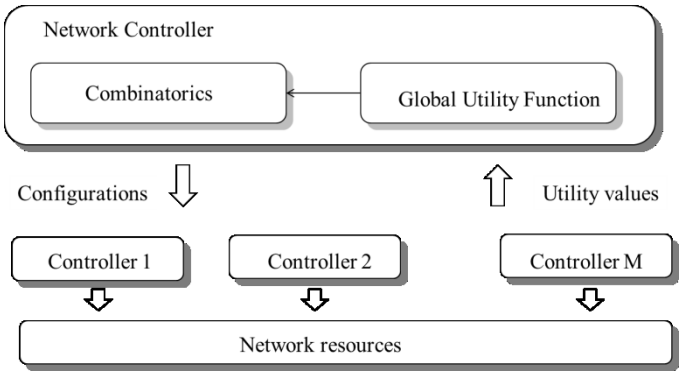

Fig. 3. Block diagram of a Network Controller

For each set of configurations, the controllers return the values of their utility functions U_i, which are used to compute the network utility function. When the network controller finds a better configuration vector, it re-starts the controllers' configuration process.

B. Algebric Representation of the Global Utility Function

The global utility function of a network can be seen as a multi-attribute utility function: general expressions of this aggregation can take additive or multiplicative forms.

Interestingly, using an algebraic representation of the multiplicative form (2), the global utility function includes terms involving linear interaction between the overall utility and the single-controller utility functions, as well as multiplicative interaction terms, taking account of the multiple interactions of controllers.

In equation (2), K is a normalization constant, ensuring that the utility values are scaled over the range space between 0 and 1.

$$U(x_1,...x_n) = \frac{\prod_i [1 + Kk_i U_i(x_i)] - 1}{K} \quad (2)$$

This algebraic representation in multiplicative form is a valid instrument for representing compensatory and complementary interactions between components. In fact, different controllers can either complement or compensate each other. Combining over both scenarios, this leads to a tree structure for the evaluation function: a series of AND and OR conditions that measure the overall possibility of optimizing the network resources. The multi-attribute utility framework can be used as a generalization for a tree analysis. The additive form, a special case in which each of the components can be treated separately.

IV. USE CASE

Let's consider an infrastructure encompassing IT resources (for virtual processing and storage) and network resources (virtual routers). Let's assume that Virtual Machine (VM) and Virtual Router (VR) can be moved from one physical node to another (the physical node merely serve as the carrier substrate on which the actual virtual node operate).

Dynamic provisioning of virtual resources (VMs and VRs) will allow load and traffic engineering in order to improve performance (e.g. limiting hotspots in the IT resources) and to reduce power consumption in the routers network. In other works, the size of the physical network can expand and contract according to load and traffic demand, by idling or powering down node not needed.

In case of hotspots in the IT resources, operators can change the allocation or migrate VMs to improve performance (e.g. response time). At the same time, as the network traffic volume decreases, operators can migrate VR to a smaller set of physical routers and shutdown or hibernate unneeded physical routers to save power. When the traffic starts to increase, physical routers can be brought up again and virtual routers can be migrated back accordingly).

In summary, in this use case we can see the interaction of two main control loops: the former is in charge of allocating VM across multiple networks for performance optimization; the latter is in charge of migrating VR a smaller set of physical routers for saving power (by shutting down or hibernating unneeded physical routers).

Although both control loops would be stable if operating alone, the combination of the two control loops may risk a positive feedback loop.

The utility function of the control-loop in charge of allocating or migrating VM across multiple networks is based on optimizing certain performance parameter (or set of parameters); for example, we may consider a function that indicates a decreasing utility as the response time increases (but any other functions could be considered depending on the required metrics).

The utility function of the control-loop in charge of migrating VR a smaller set of physical routers can be based on saving electrical power. The two utility functions cannot be used independently otherwise instabilities may occur.

Suppose during one minute a hot spot in a node 1 happens. VM control loop notices this and shifts some VM away from node 1 to a node 2 in the next minute, while at the same time VR control loop notices traffic surge in node 1 and moves more VR to that node and reduce the number of VR in node 2 (saving energy in node 2). While either of these actions alone would lead toward convergence, the two in combination cause overcompensation.

A global utility function should be considered as a combination of the above two functions. Overall the network controller will have the task of optimize the global utility function.

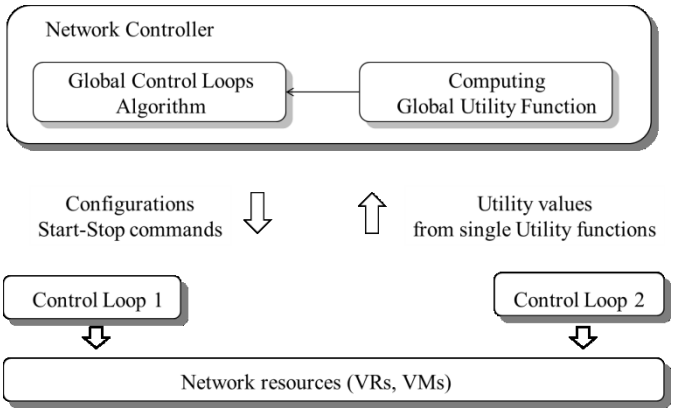

Fig. 4. Block diagram of a Network Controller

In the ongoing simulation activities, the global control loops algorithm (figure 4) searches the space of control loops configurations using (for example) a beam-search algorithm. This could be done at regular intervals, upon reaching a trigger or in reaction to changes in the global utility function.

Other methods are under consideration for automating the creation of utility functions: for example evolving them through genetic programming. To construct the utility function, the genome consists of a predicate grammar. On the other hand, such utility function can only provide boolean values, where one of the main attractions of utility is to provide comparable values for optimization.

V. Conclusions and Future Work

The increasing level of complexity in future networks will bring new management challenges and systemic risks. Such levels of complexity will soon be comparable with that of today's financial markets (whose dynamics come from the intertwining of humans operations and automated trading systems). There will almost certainly be risks of the network experiencing sharp transitions from overall stability to instability as the complexity increases. Networks are strategic assets, and so it is of paramount importance to mitigate the risk of these stability transitions, whose primary effects might not only jeopardize performance.

In order to begin to truly address this problem, we have proposed a methodology for controlling and taming instabilities in complex networks. The starting concept is modeling the network as a complex ensemble (e.g. ecosystem) of resources and controllers. Utility functions are associated to these controllers and a proper global utility function (elaborated as an aggregation of controllers' utilities) is associated to the network. This paper argues that the maximization of a proper network utility function ensures optimization and stability at the same time.

Some next steps are already being investigated, including the development of a concrete use case to test and demonstrate (with simulations and emulations) the feasibility of the proposed methodology. Theoretically, the methodology will be completed by the addition of theorems, which are still under study.

Acknowledgment

This work is partially supported by the European Commission through the UniverSELF project of the 7th FP.

References

[1] N. Johnson, G. Zhao, "Financial black swans driven by ultrafast machine ecology", available at arXiv:1202.1448v1;
[2] A.G. Haldane, R.M. Day ,"Systemic risk in banking ecosystems", Nature, 2011, Volume: 469, Pages: 351–355;
[3] S. A. Kauffman, "The Origins of Order: Self-organization and Selection in Evolution", Oxford University Press, 1993;
[4] T. Ohira and R. Sawatari, Phys. Rev. E 58, 193 (1998);
[5] R. V. Sole, "Information Transfer and Phase Transitions in a Model of Internet Traffic", Physica A (2001), Volume: 289, Issue: 3—4: Publisher: Elsevier BV, Pages: 595-605;
[6] V. Marbukh, "Towards Understanding of Complex Communication Networks: Performance, Phase Transitions & Control", Sigmetrics Performance Evaluation Review, 2007;
[7] B. H. Wang, "Routing strategies in traffic network and phase transition in network traffic flow", Pramana, Journal of Physics, Vol. 71, n.2, August 2008;
[8] B. Ford, "Icebergs in the Clouds: the Other Risks of Cloud Computing", available at arXiv:1203.1979v1;
[9] F.P. Kelly, " Charging and rate control for elastic traffic". European Transactions on Telecommunications 1997; 8:33–37;
[10] F.P. Kelly, A. Maulloo, D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability". Journal of the Operational Research Society 1998; 49:237–252;
[11] M. Chiang, S. H. Low, A. R. Calderbank, J. C. Doyle, "Layering As Optimization Decomposition: A Mathematical Theory of Network Architectures", Proceedings of the IEEE, Vol. 95, No. 1. (05 January 2007), pp. 255-312, doi:10.1109/JPROC.2006.887322;
[12] Mo J,Walrand J. "Fair end-to-end window-based congestion control". IEEE/ACM Transactions on Networking 2000; 8(5):556–567;
[13] A. Manzalini, P.H. Deussen, S. Nechifor et alii "Self-optimized Cognitive Network of Networks", in Oxford Journals "The Computer Journal"; 2010, Volume 54, Issue 2, pp 189-196;
[14] F.P Kelly, "The Mathematics of Traffic in Networks", In "The Princeton Companion to Mathematics" Princeton University Press, 2008. 862-870.