Copyright and Reference Information: This material (preprint, accepted manuscript, or other author-distributable version) is provided to ensure timely dissemination of scholarly work. Copyright and all rights therein are retained by the author(s) and/or other copyright holders. All persons copying this work are expected to adhere to the terms and constraints invoked by these copyrights. This work is for personal use only and may not be redistributed without the explicit permission of the copyright holder. The definite version of this work is published as
[·] Richard Holzer. Modeling and control of complex and self organizing systems. Technical Report MIP-0916, Faculty of Computer Science and Mathematics (FIM), University of Passau, 2009. Essay Collection of Ph.D. Course.
See http://www.net.fim.uni-passau.de/papers/Holzer2009b for full reference details (BibTeX, XML).

Modeling and Control of Complex and Self-Organizing Systems

Richard Holzer[†] (Ed.)

[†] Department of Informatics and Mathematics, University of Passau
holzer@uni-passau.de

Technical Report, Number MIP-0916
Department of Informatics and Mathematics
University of Passau, Germany
July 2009

Modeling and Control of Complex and Self-Organizing Systems

Richard Holzer (Ed.)

Faculty of Informatics and Mathematics, University of Passau, Innstrasse 43, 94032 Passau, Germany

Abstract. At the University of Passau, a EuroNF PhD Course on Modeling and Control of Complex and Self-Organizing Systems was held from March 30th - April 3rd, 2009. This technical report contains the essays of the participants.

Preface

Not so long ago, computer systems consisted of a single CPU and only a few peripheral devices directly connected to the CPU. Each system was used by a single user running only a single application and there was no connection to other computers. All relevant states of the system could be foreseen and the desired behavior of the system could be ensured by considering all relevant states. Today, computer systems consist of many concurrent components and millions of computer systems are interconnected and influence each other. In the future, there will be networked systems (like sensors) in most buildings, cars, streets, and on most persons in a city acting as a single, coherent system dynamically reallocating resources and priorities. Due to exponential complexity and incomplete knowledge, the exact state of the overall system can neither be evaluated nor be controlled centrally. The correctness of the overall network of systems has to be ensured by local policies applied to each single system.

At the University of Passau, a EuroNF PhD Course on "Modeling and Control of Complex and Self-Organizing Systems" was held by Richard Holzer from March 30th - April 3rd, 2009. This PhD course provided insights into building tractable models for complex and self-organizing systems by covering the following topics (among others):

- Modeling techniques
- Phase Space
- Continuous time models
- Discrete time models
- Differential equations
- Recurrence equations
- Agent-based modeling
- Bifurcation
- Cellular Automata

– Model examples (e.g., Lotka-Volterra Model, Nagel-Schreckenberg Model)

The theoretical sessions were accompanied by practical hands-on tutorials to illustrate the applicability of the knowledge imparted earlier. This technical report contains the essays of the participants.

Emergence in Complex Systems

Essay

Gerhard Fuchs

University of Erlangen-Nuremberg
Dept. of Computer Science 7 (Computer Networks and Communication Systems)
Martensstr. 3, 91058 Erlangen, Germany
gerhard.fuchs@informatik.uni-erlangen.de
http://www7.informatik.uni-erlangen.de/~fuchs/

Abstract. The number of computers increases up to now. Systems are becoming more complex with regard to the number of sub systems that are integrated and interacting. Coming from the field of robot sensor networks (RSNW), this essay gives short definitions of "emergence" and "complex system" (CS) and points out, that the relation between emergence and self-organization (SO) is up to now unclear. Additionally it contains my bio-inspired thoughts (many control one, dualism system / signal, description of cycles analog to complex numbers) and my vision to find a relation between complex numbers / complex systems and a recursive definition of a system. A minimal system, that contains the essence of an emergent, complex systems would be a step to more understanding.

Introduction. In the frame of my PhD studies I'm engaged in RSNW, which can be seen as an example for a CS. Parallel to my research in this field, I have studied some mechanisms in the field of cell biology, for inspiration.

A Wireless Sensor Network (WSN) consists of several interacting sensor notes (I call it spots). According to Akyildiz [1], a spot has four main parts: a power unit, a transceiver unit, a processing unit and a sensing unit. At our chair we additionally use the spots to control a chassis or propellers to get robots. Many groups are engaged in finding a way, how to handle and control the huge amount (hundreds, thousands, ten thousands ...) of spots, that should interact in future. Thereby "emergence" is one buzzword mentioned in this context.

Also much work has been done in understanding and using emergence for the purpose of engineering, it still seems to me, that there is no common sens whether emergence is pane (there is something mystical happening we haven't expected) or boon (if we understand this phenomena we can find a new way of programming CS).

In this essay I give some informal definitions of "emergence" and "complex system", that I found in literature. I describe a gap between three descriptions of the relation between emergence and SO, that I have recognized during my studies. Afterwards I present some of my thoughts in the field of this topic and give a conclusion.

What is emergence? Wikipedia says:

> "In philosophy, systems theory and science, emergence is the way complex systems and patterns arise out of a multiplicity of relatively simple interactions." [2]

There exist many other definitions of emergence beside the verbal definition presented above. Deguet et al. have written a detailed survey paper [3] that summarizes and discusses important aspects of this phenomena. Further definitions based on entropy were given by Mnif and Müller-Schloer [4] and Holzer et al. [5]. The authors introduces a quantitative model for emergence.

What is a complex system? Wikipedia says:

> "A complex system is a system composed of interconnected parts that as a whole exhibit one or more properties (behavior among the possible properties) not obvious from the properties of the individual parts." [6]

Furthermore they mentioned the following features for CS: difficult to determine boundaries; dynamic network of multiplicity; CS may be open, may have memory, may be nested, may produce emergent phenomena; relationships are non-linear and contain feedback loops. Ant-colonies and cells are two examples for CS.

What is the relation between emergence and SO? One special relation, which is unclear up to now, seems to be between emergence and SO. In the field of computer science I have found three different points of view. De Wolf and Holvoet wrote, that emergence and SO are different concepts [7]. Mnif and Müller-Schloer defined "emergence as self-organized order" [4]. Finally Holzer et al. see emergence as one main property of SO [5].

Cell biology. A body of an animal consists of several interacting cells and can be seen as an example for a CS. Many proteins are used for singling between cells (many control one). A protein, seen for it self, is a system. At the same time a protein can be a signal for the cell. I have learnt from these studies, that a system can be a signal and a signal can be a system (dualism system / signal). Additionally there is something like a cycle during the development of a fly. Cells produce signals, that signals influences the cells to produce signals. Can this cycle be described using a construct like $se^{i\omega t}$ (s is a system, t a time) in analogy to complex numbers?

Is there a similarity between "complex" number and a "complex" system? About this aspect I am thinking since I have read an excursus form Görnitz about the impact of complex numbers to physics in his book [8]. It is obvious that syntactically "complex" is a common part in both expressions. Why

have mathematicians introduced complex numbers? There was a lack of expressiveness using the "traditional numbers". Could a real part and a imaginary part of a system, in analogy to complex numbers, extend the expressiveness of system descriptions, so that emergence can be covert?

What is a system? I have not found a transfer of Aristoteles sentence: "The hole is more than the sum its the parts" like my following recursive definition, covering two layers, in literature:

Definition 1 (System). *A system σ is more Δ than the set of all parts $\mathbb{S}$.*

$$\sigma := (\mathbb{S}, \Delta)$$

$\mathbb{S}$ *set of sub systems (also a system).*
Δ *additional properties beyond those of the subsystems, to be clarified.*

Definition 2 (elementary system). *An elementary system $\sigma^{\odot}$ is a system without sub systems. It is the end of the recursive definition of a system. The labeling is done using a superior $\odot$. $\odot$ is called the core the elementary system.*

Conclusions. I currently recognize a theory driven and an application driven stream in the field of emergence in CS. One group of researchers analyse and simulate existing systems in order find an abstract description. Another group of researchers build more and more complex systems and get confronted with phenomenas, that are new to them. RSNW and cell biology are two examples for CS where emergence can occur. But many aspects of these examples are still unknown. The optimum for further research would be a minimal system that contains the essence of an emergent, CS. Many activities are done by many interacting researchers. I am curious about what will emerge in future.

References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: a survey. Elsevier Computer Networks **38** (2002) 393–422
2. Wikipedia, The Free Encyclopedia: Emergence [web; 30.04.2009].
3. Deguet, J., Demazeau, Y., Magnin, L.: Elements about the Emergence Issue, A Survey of Emergence Definitions. ComPlexUs **3** (August 2006) 24–31
4. Moez Mnif and Christian Müller-Schloer: Quantitative emergence. In: Proc. of the IEEE Mountain Workshop on Adaptive and Learning Systems, IEEE (2006) (SMCals: Logan, US-WV; July 2006).
5. Holzer, R., de Meer, H., Bettstetter, C.: On Autonomy and Emergence in Self-Organizing Systems. In: Proc. of the 3rd International Workschop on Self-Organizing Systems. Vol. 5343 of LNCS., Springer (2008) 157–169 (IWSOS: Vienna, AT-9; December 2008).
6. Wikipedia, The Free Encyclopedia: Complex System [web; 29.04.2009].
7. Wolf, T.D., Holvoet, T.: Emergence Versus Self-Organisation: Different Concepts but Promising When Combined. Vol. 3464 of LNAI. Springer (May 2005) 1–15
8. Thomas Görnitz: Der kreative Kosmos. Elsevier GmbH, Spektrum Akademischer Verlag, Heidelberg, DE-BW (2007)

Autonomy in Self-Organizing Systems

Nafeesa Bohra

Chair of Computer Networks and Computer Communications, Faculty of Informatics and Mathematics, ITZ/IH, University of Passau, Innstr. 43, D - 94032, Passau, Germany. bohra@fmi.uni-passau.de

Abstract: Autonomy means self-governing, while a system is said to be autonomous if and only if the organization of internal aspect of the system processes are the dominant factor in the system's self-preservation, making both the system itself and the processes that constitute to autonomy functional. It means that autonomy is an organizational property constituted of processes with some degree of closure, though the closure to external forces need not be complete. Autonomic Computing is a challenging new paradigm in system development whose basic aim is to develop a system which on one hand self-protecting and self-healing while on the other hand it must be self-configuring and self-optimizing. The essay discusses the basic fundamentals of autonomy in self-organizing systems.

The basic design goal of future computing and networking paradigm is to minimize the administrative requirements not only for the users but also for the ISPs (Internet Service providers). In order to increase the functionality/usability it is required that the system should be developed in such a way that it must not only easy to use but also successfully configure itself according to the changing circumstances. Another requirement is that the system should detect and may be able to correct the failures automatically. All the above mentioned facts have lead towards the development of Self-Organizing System (SOS). Self-organization is a process of attraction and repulsion in which the internal organization of a system, normally an open system, increases in complexity without being guided or managed by an outside source. Self-organizing systems typically (but not always) display emergent properties.

A system is said to be self-organized when:

- It evaluates in an organized form without the presence of external pressure/forces.
- It exhibits emergent properties.
- It can be seen as the increase of coherence or decrease of statistical entropy.
- It possess the following main properties:

Emergence, Adaptability, Decentralization, and Autonomy,

Emergence: The appearance of a property or feature that have not previously observed as a functional characteristic of the system. Or properties/pattern appears in a system as a complete unit but disappear in the single component. For example: An automobile is an emergent property of its interconnected parts. That property disappears if the parts are disassembled and just placed in a heap.

Adaptability: It refers to a systems' ability to change itself in order to accommodate changes, especially the changes, in its environment that have very little influence upon the overall behaviour of the system.

Decentralization: A single entity or a group of entities are not responsible for the control of the entire system, in fact the system is controlled by all the entities of the system.

Autonomy: It means self-governing. i.e. Almost no external control is required for the system.

Autonomy means self-governing, and it comes from a Greek word which means independent. Functionality, intentionality, and meaning are the basic building blocks of Autonomy. A system is said to be autonomous if it uses its own information to modify itself and its environment in order to enhance its survival. An autonomous system accommodates itself through unexpected self-organizing processes, together with some constraints that maintain autonomy. OR

A system is said to be autonomous if and only if the organization of internal aspect of the system processes are the dominant factor in the system's self-preservation, making both the system itself and the processes that constitute to autonomy functional. Hence, it can be said that autonomy is an organizational property constituted of processes with some degree of closure, though the closure to external forces need not be complete. Hence, it can be said that autonomy in self-organizing systems has yield towards the development of Autonomic Computing (AC).

AC was first introduced by IBM in 2001. The basic aim behind this initiation was to develop a computing system capable of self-management, in order to overcome the rapidly growing complexity of the computing system management and also to reduce the barrier that complexity poses towards further growth. In other words AC can be defined as the self-managing characteristics of distributed computing resources, which must be able to adapt unpredictable changes while limiting the administrative requirements for the users and the operators.

AC aims towards the following main concepts:

- A system must be self-protected and self-healing so that the reliability can be increased while on the other hand;
- A system must be able to utilize self-optimizing and self-configuring mechanisms so that the autonomy and performance will be increased by enabling the system to adapt to changing circumstances.

From the above discussion it is concluded that an autonomic system is self-managing which means that it must possess the four functional properties defined by IBM, namely:

Self-healing:	Automatic discovery, and correction of faults/failures; This basically means that a system not only be capable of identifying the failures successfully but also be able to repair them with minimum disruption to the users while avoiding loss of data and delays.
Self-configuring:	Automatic configuration of components; Alternatively it is system's ability to accommodate itself according to the changing circumstances.
Self-protecting:	Proactive identification and protection from arbitrary attacks; System will be able to defend itself from accidental and malicious attacks, which means that the system must be aware of potential threats and must know how to handle such threats.
Self- optimizing:	Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements;

From the above discussion it is summarized that in order to achieve the above mentioned objectives it is necessary that the system must possess a self-awareness property and it must be aware about its current external operating conditions. A system must possess self-monitoring property in order to detect changes i.e. what's going on in its circumstances and then adapt it accordingly. Precisely, it can be said that a system must have knowledge about its resources; components, what are their desired performance characteristics, their current status, and what is the status of inter connection with other system.

References

[1] Roy Sterritt and Dave Bustard: Towards an Autonomic Computing Environment: *In Proc. of the 14th Int.*

Workshop on Database and Expert systems Applications (DEXA '03): IEEE Computer Society, 2003.

[2] Hermann de Meer: self-Organization in Peer-to-Peer Systems: *Dagsthule Seminar Proc 04411*: Service Management and Self-Organization in IP-based Networks: http://drops.dagsthule.de/opus/volltexte/2005/86

[3] Richard Holzer and Hermann de Meer: On modeling of Self-Organizing systems: *In Proc. Autonomics 2008*, Sep 23 – 25, 2008, Turin, Italy.

[4] Richard Holzer and Hermann de Meer and Christian Bettstetter: On Autonomy and Emergence in Self-Organizing Systems: *IWSOS 2008 - 3rd International Workshop on Self-Organizing Systems*: Vienna, Austria, December 10-12, 2008, Springer Verlag 2008.

[5] Autonomic Computing Concepts, IBM White Paper, 2001.

[6] Jeffery O. Kephart and David M. Chess: The Vision of Autonomic Computing: IBM TJ, Watson Research Center: *Published by IEEE Computer Society: 2003*: ISBN: 0018-9162/03.

[7] Roy Sterritt and Dave Bustard: Autonomic Computing- A Means of Achieving Dependability: *In Proc. of IEEE Int. Conf. on the Engineering of Computer Based systems (ECBS '03):* Huntsville, Alabama, USA, April 7 – 11 2003, pp 247 – 251.

Stability of Equilibria for Continuous Systems

Jennifer Mylosz

University of Hamburg, Department of Mathematics, Center of Mathematical Statistics and Stochastic Processes, Bundesstrasse 55, 20146 Hamburg, Germany

Abstract. To analyze a continuous system's behavior trajectories and equilibria of its differential equation can be investigated, then the equilibria can be checked for stability. The needed definitions will be given and ways to derive the equilibria and their stability type will be presented and discussed.

1 Continuous Systems

This essay about the analysis of dynamic systems with a continuous time variable is based on [1, pp. 1-65] and on the part of macro level modeling of continuous systems in [2]. Since this essay is supposed to be a summary, proofs are omitted throughout and the interested reader finds more information in the mentioned literature. At first, the continuous systems considered here will be introduced by

Definition 1. *A* continuous system *is a system which consists of a state space S (here $S = \mathbb{R}^n$), a time variable $t \in \mathbb{R}$ and an evolution rule which describes the change of the state during the time in terms of a differential equation*

$$\frac{dx}{dt} = \dot{x} = X(x)$$

where $x \in \mathbb{R}^n$ is a vector and $X : \mathbb{R}^n \to \mathbb{R}^n$ is a vector field.

To analyze the behavior of a continuous system defined above, one can have a look at the trajectories of the system and, moreover, derive equilibrium points of the according differential equation and analyze their stability.

1.1 Trajectories

Every differential equation modeling a real system should have a unique solution. We assume that the solution is uniquely determined by the initial condition $x(t_0) = x_0$. We shortly write $x(t, t_0, x_0)$ for the according *solution of the differential equation* at time t. Setting $t_0 = 0$, the solution $x(t, 0, x_0)$ describes the state of the system t time units after starting in state x_0.
The solution of the differential equation can be drawn as an orbit or trajectory in $\mathbb{R}^n$. The set of all maps $x(t, t_0, x_0) : \mathbb{R}^n \to \mathbb{R}^n, t \in \mathbb{R}$, forms the *trajectory (orbit)* for the initial condition $x(t_0) = x_0$. Consequently, we get an orbit for each initial condition. Considering all possible initial conditions the according orbits are either equal or disjoint. The plot of typical orbits is a helpful representation of the flow of the considered system.

1.2 Equilibria and their Stability

A more detailed analysis of a system can be done by searching for equilibria of its differential equation.

Definition 2. *A point $x^* \in S$ is called* equilibrium point *of the differential equation $\dot{x} = X(x)$, if $X(x^*) = 0$.*

If x^* is an equilibrium point, then $x(t, t_0, x^*) = x^*$ for all $t \in \mathbb{R}$ with $t \geq t_0$. Thus each equilibrium point is a fixed point of the flow and its orbit is the fixed point itself.

There are several classifications of equilibria. A distinction is made between a stable and an unstable equilibrium, or more precisely, an asymptotically stable, a neutrally stable, and an unstable one.

Definition 3. *Let x^* be an equilibrium of the differential equation $\dot{x} = X(x)$.*

- *x^* is* stable *if each trajectory starting near x^* will stay near x^*, i.e., if for any $\varepsilon > 0$ there exists $\delta(\varepsilon) > 0$ such that $\|x(t, 0, x_0) - x^*\| < \varepsilon$ holds for all $t > 0$ and for all $x_0 \in S$ with $\|x_0 - x^*\| < \delta(\varepsilon)$.*
  Even more, x^* is asymptotically stable *if each trajectory starting near x^* will converge to x^* (i.e., $\lim_{t\to\infty} x(t, 0, x_0) = x^*$), otherwise x^* is* neutrally stable*.*
- *x^* is* unstable *if it is not stable.*

To get a feel for the different stability characteristics of equilibria see Fig. 1.

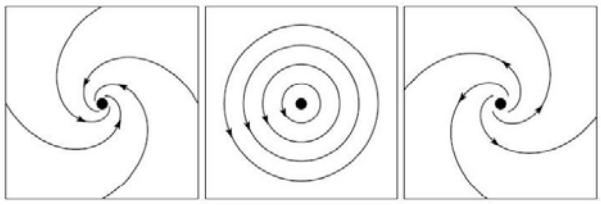

Fig. 1. [1, p. 55] Examples for the stability types of equilibria, from left to right: asymptotically stable, neutrally stable, unstable.

For Definition 3 it is necessary to solve the differential equation at first, in order to analyze the behavior of trajectories near the equilibrium point. To avoid the difficulty of finding a solution, in many cases it is possible to derive the behavior directly from the differential equations.

To find out directly from the differential equation whether an equilibrium is stable or unstable, we need to look at linear and nonlinear differential equations separately.

Theorem 4. *Consider a linear differential equation $\dot{x} = A \cdot x$. Let x^* be an equilibrium of the differential equation.*

– *x^* is stable if and only if all eigenvalues of A have no positive real part. More precisely, x^* is asymptotically stable if and only if all eigenvalues of A have a negative real part, otherwise x^* is neutrally stable.*
– *x^* is unstable if and only if at least one eigenvalue of A has a positive real part.*

So if the differential equation is linear, the stability can be derived from the eigenvalues of the matrix.

A nonlinear differential equation $\dot{x} = X(x)$ with equilibrium in x^* can be transformed by Taylor approximation into the linear differential equation $\dot{y} = DX(x^*) \cdot y$ where y substitutes $x - x^*$ and $DX = \left(\frac{dX_i}{dx_j}\right)_{i,j=1,\dots,n}$ is the Jacobi matrix. Therefore it seems obvious that the eigenvalues of the Jacobi matrix give the information of the stability of the equilibria. Indeed, this is possible for hyperbolic equilibria.

Definition 5. *Consider a nonlinear differential equation $\dot{x} = X(x)$. Let x^* be an equilibrium of the differential equation. x^* is* hyperbolic *if all eigenvalues of the Jacobi matrix $DX(x^*)$ have a nonzero real part.*

For a hyperbolic equilibrium, the transformation to the linear differential equation does not change the stability. This fact leads to

Theorem 6. *Consider a nonlinear differential equation $\dot{x} = X(x)$. Let x^* be a hyperbolic equilibrium of the differential equation.*

– *x^* is asymptotically stable if and only if all eigenvalues of $DX(x^*)$ have a negative real part.*
– *x^* is unstable if and only if at least one eigenvalue of $DX(x^*)$ has a positive real part.*

So if the considered equilibrium of a nonlinear differential equation is hyperbolic, the stability can be derived from the eigenvalues of the Jacobi matrix. But for non-hyperbolic equilibria, the theorem above does not provide any information about the analysis of their stability.

2 Discussion

The presented ways to analyze the stability of equilibria for continuous systems are sufficient to get results for systems whose differential equation is easy to solve as well as for systems whose differential equation is linear. For continuous systems which have a nonlinear differential equation and whose differential equation is difficult to solve the definitions and theorems presented above do not always turn out satisfactory. As long as hyperbolic equilibria are to be analyzed for stability Theorem 6 is applicable. If an equilibrium of a nonlinear differential equation is non-hyperbolic, one can search for (or guess) the *Lyapunov function* for the analysis of stability instead. More information about the Lyapunov function can be found in [1].

References

1. Boccara, N.: Modeling Complex Systems. Springer, New York et al., 2004.
2. Holzer, R.: Modeling and Control of Complex and Self-Organizing Systems. Presentation of the EuroNF PhD Course on "Modeling and Control of Complex and Self-Organizing Systems", University of Passau, Germany, 2009. http://www.net.fim.uni-passau.de/mcsos

Bifurcation

Edzard Höfig[1], Stefan-Liviu Taranu[2], and Stefan Neumann[3]

[1] Fraunhofer Institute, Berlin, Germany,
edzard.hoefig@fokus.frauenhofer.de
[2] Fraunhofer Institute, Berlin, Germany,
stefan-liviu.taranu@fokus.frauenhofer.de
[3] Hasso-Plattner-Institute, Potsdam, Germany
stefan.neumann@hpi.uni-potsdam.de

Abstract. Using models for describing complex systems facilitates the analysis concerning properties like stability and the convergence to equilibria. We discuss the influence of parameter modification regarding erratic changes in the overall model behavior, by applying bifurcation theory. Different kinds of bifurcation types, like Saddle-Node, Transcritical, and Pitchfork Bifurcation are discussed for the family of one parameter, one-dimensional vector fields by using examples and giving sufficient conditions, which need to be fulfilled.

1 Foundations

Models of complex systems are commonly described using differential equations. Often stability criteria are essential for the overall behavior of the considered system. Following we discuss the analysis of differential equations regarding their equilibria and stability properties.

1.1 Equilibria

Let the differential equation $\dot{x} = X(x, \mu)$ describe a complex system where x is the system variable and μ is a parameter. Intuitively, the system is at equilibrium when it is not changing over time. Therefore its derivation against the time should be zero: $X(x^*, \mu^*) = 0$, where (x^*, μ^*) give the points of equilibrium.

1.2 Stability

Some knowledge is necessary before we continue with the analysis of stability. We therefore define the trajectory of the system as the path in time a system's behavior is following; and the closed orbit as a cyclic trajectory, i.e. if we begin at a point of the trajectory, we will end up in the same point after a certain amount of time. If we start analyzing the model when it is not in equilibrium, then it might tend to evolve

1. towards the equilibrium (Figure 1.a), in which case the equilibrium is stable

2. away from the equilibrium (Figure 1.b) in which case the equilibrium is unstable
3. around the equilibrium point (Figure 1.c) in which case the equilibrium is neutrally stable or
4. towards a closed orbit around the equilibrium points (see Figure 2)

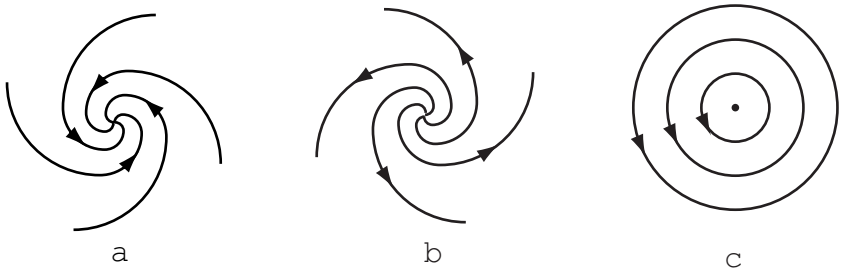

Fig. 1. Stability points

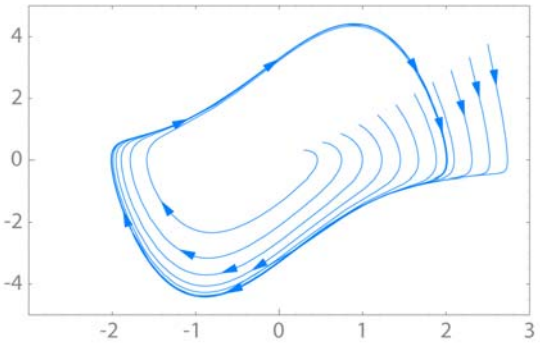

Fig. 2. Limit cycle

2 Bifurcation

The term "Bifurcation" refers to an erratic change in the qualitative behavior of a model in response to a small and smooth modification of its parameter values. Such a change might appear in continuous, as well as discrete systems. Being able to determine the exact parameter values at which a bifurcation appears within the phase space of a model is important for determining its overall stability.

The Logistics Map A popular example for demonstrating bifurcation is the "logistics map" (see equation 1 as found in [1]). In the field of biology the logistics map is used as a demographic model for capturing the effects of reproduction and starvation in animal populations. It also demonstrates that chaotic behavior can ensue from a simple, discrete, first-order differential equation due to bifurcation.

$$x_{n+1} = rx(1 - x_n) \tag{1}$$

The logistics map has a single parameter r, which is a positive number and specifies the combined rate of reproduction and starvation. The variable x is a number between zero and one and gives the size of the population for each step n (which represents years).

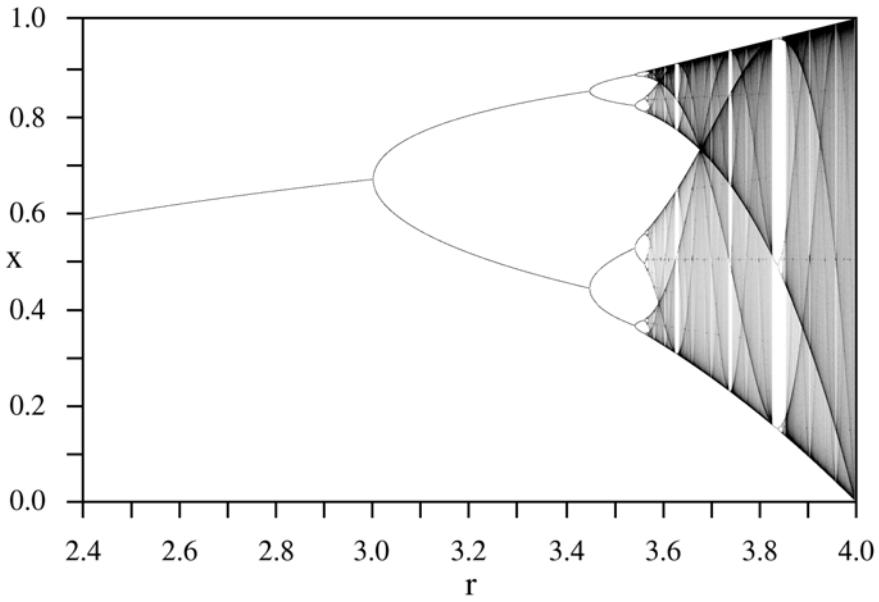

Fig. 3. Bifurcation Diagram of the Logistics Map

Figure 3 shows a bifurcation diagram[4] depicting the equilibria that this system can assume as a function of the bifurcation parameter r in the range between 2.4 and 4.0. It can be seen that for values of r smaller than 3, the population of x converges to a stable equilibrium. Values larger than 3 show bifurcation: The population x first oscillates between two possible stable values. The larger the value of r, the more different equilibria the population can assume, which finally leads to chaotic behavior for the population size. The location where

[4] Taken from Wikipedia

a model qualitatively changes its stability is referred to as a bifurcation point. The logistics map example is only shown here to give the reader an initial understanding on bifurcation theory, in the following discussion we will use different examples. Bifurcation points can be classified as several different kinds, and we subsequently discuss three types: Saddle-Node, Transcritical and Pitchfork Bifurcation. For a more in-depth discussion about other types of bifurcation points, please refer to [2, page 71ff.]

2.1 Saddle-Node Bifurcation

The term "Saddle-Node Bifurcation" refers to those bifurcation points in which two equilibria meet and extinguish each other. We are going to demonstrate how to analyze equilibrium points for their saddle-node bifurcation properties for the family of one-parameter, one-dimensional vector fields as specified by equation 2. For sake of simplicity, we suppose that the equilibrium point for analysis is supposed to be found at $x = 0$, $\mu = 0$.

$$\dot{x} = X(x, \mu) \tag{2}$$

Following [2, page 75–77] the equation 2 might have a saddle-node bifurcation point at (0,0), if it is a non-hyperbolic equilibrium point. This is the case if both equations 3 hold. Furthermore, if both equations 4 are fulfilled, a saddle-node bifurcation point has been found.

$$X(0,0) = 0 \qquad \frac{\partial X}{\partial x}(0,0) = 0 \tag{3}$$

$$\frac{\partial X}{\partial \mu}(0,0) \neq 0 \qquad \frac{\partial^2 X}{\partial^2 x}(0,0) \neq 0 \tag{4}$$

For an example consider equation 5

$$\dot{x} = \mu - x^2 \tag{5}$$

The equilibria of this equation can be found as

$$x^* = \pm\sqrt{\mu} \tag{6}$$

Following the process for determination of equilibria we find that there is no equilibrium for $\mu < 0$, one non-hyperbolic equilibrium for $\mu = 0$, and two equilibria for $\mu > 0$. Calculating the Eigenvalues for the two last results we get $x^* = \mp 2\sqrt{\mu}$, giving us a stable equilibrium for $\mu > 0$ and an unstable one for $\mu < 0$.

This is also depicted in Figure 4: case `a` corresponds to $\mu < 0$, case `b` to $\mu = 0$, and case `c` shows the two equilibria $x^* = -\sqrt{\mu}$ and $x^* = \sqrt{\mu}$ for $\mu > 0$. Arrows in the diagram indicate convergence, respectively divergence in the vicinity of the equilibrium points. These phase portraits match to the general bifurcation

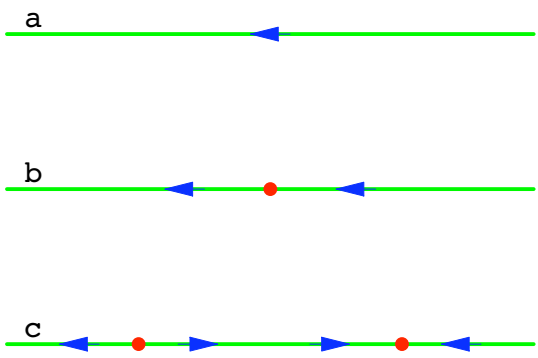

Fig. 4. Phase Portraits for a Saddle-Node Bifurcation

diagram in Figure 5, shows that a saddle-node bifurcation for a family $X(x, \mu)$ leads to a curve of fixed points tangent to $\mu = 0$ at $x = 0$ and lying entirely to one side of $\mu = 0$, as demanded by the saddle-node bifurcation conditions. As a convention, the solid line represents the collection of stable equilibrium points and the dashed line represents the collection of unstable equilibrium points. A larger dot represents the bifurcation point and smaller ones represent equilibrium points[5].

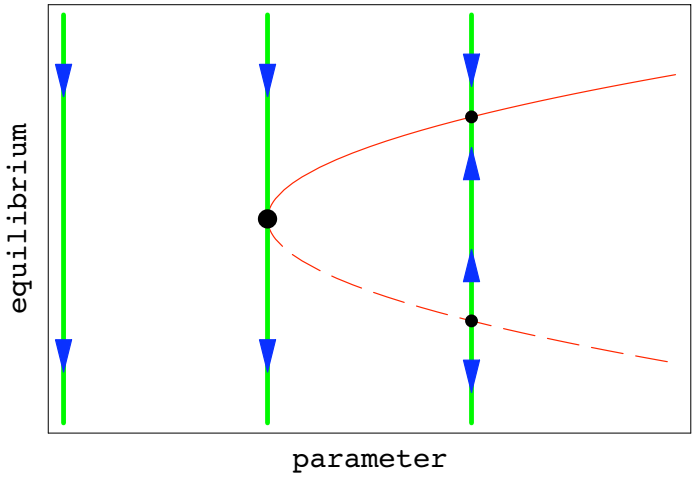

Fig. 5. Saddle-Node Bifurcation Diagram

[5] These conventions are valid for the entire document.

2.2 Transcritical Bifurcation

In a bifurcation the equilibria points change their stability, depending on the change of the parameter. The number of equilibrium points remains the same in the transcritical bifurcation, whereas in saddle-node bifurcation the number of these points seems to disappear after certain value of the parameter.

Necessary Conditions Let

$$\dot{x} = X(x, \mu) \quad (x \in \mathbb{R}, \mu \in \mathbb{R}) \tag{7}$$

be the differential equation that describes a one-dimensional system. To simplify the problem we will assume that the equilibrium point (x^*) and the parameter (μ^*) are both equal to 0. In order for the system to achieve a non-hyperbolic equilibrium in $(0, 0)$ the equation 3 should be met. According to [2, page 78], because there are two lines that intersect in $(0, 0)$ (cf. Figure 6) equation 8 should also be satisfied.

$$\frac{\partial X}{\partial \mu}(0, 0) = 0 \tag{8}$$

Additionally equations 9 need to be satisfied:

$$\frac{\partial^2 X}{\partial^x}(0, 0) \neq 0 \qquad \frac{\partial^2 X}{\partial x \partial \mu}(0, 0) \neq 0 \tag{9}$$

As seen in Figure 6 there are 2 (red) lines – described by the equations $x = \mu$

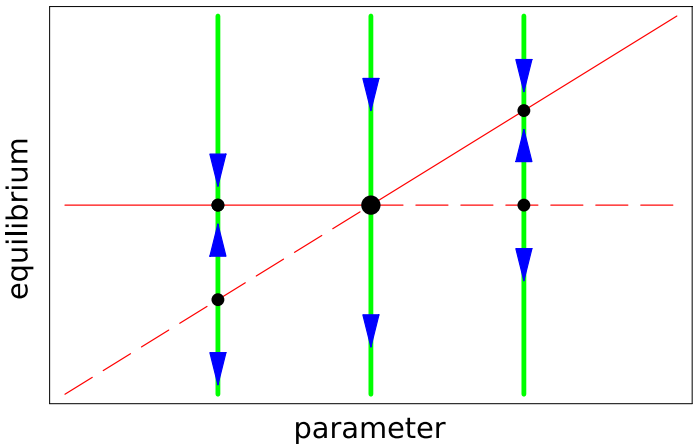

Fig. 6. Transcritical Bifurcation Diagram.

and $x = 0$ – that intersect in 0. Considering the convention, one can see that the equilibrium points change their stability once they pass the bifurcation point $(0, 0)$.

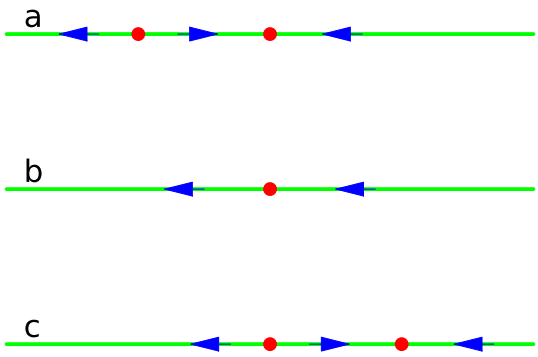

Fig. 7. Transcritical Bifurcation Phases

Example Let's take, for example the equation

$$\dot{x} = \mu x - x^2 \tag{10}$$

The solutions of this equation are $x_1 = 0$ and $x_2 = \mu$.If $\mu = 0$ then we will have only an equilibrium point; for the other cases ($\mu \neq 0$) there will be two equilibrium points.

The Jacobi matrix is $DX(x, \mu) = \mu - 2x$. We can see that when $\mu = 0$ we have only one equilibrium point $x^* = 0$ and that is not hyperbolic. If $\mu \neq 0$ we will have two solutions $x_1^* = 0$ and $x_2^* = \mu$. We then calculate the Jacobi matrix for each solution. We will have $DX(x_1^*, \mu) = \mu$ which is stable for $\mu < 0$ and unstable for $\mu > 0$; and $DX(x_2^*, \mu) = -\mu$ which is unstable for $\mu < 0$ and stable for $\mu > 0$. The points discussed above are represented with red in Figure 6.

In analyzing the behavior of the systems one needs to know first what are the points where the system is stable. Once these points and their nature are known, one can see what are the behaviors of the system around these equilibrium points. This way one can have a clear picture of evolution of the model and can say with precision its state at a particular moment in time.

An example in the real world is the predator-prey model, where the evolution of the two species is analyzed. Supposing that the model is defined as in the example above, where it has a dependency on parameter μ one can say that

- the population can be unstable and extinguish or increase to infinity for $\mu = 0$
- the population stabilizes at a high number of specimens
- the population stabilizes at a low number of specimens

2.3 Pitchfork Bifurcation

The necessary and sufficient conditions for analyzing "Pitchfork Bifurcation" for the family of one-parameter and one-dimensional vector fields are similar to

these, which exist to exhibit a transcritical bifurcation. We take into consideration an equation of the form like shown in equation 2 and again assume that an equilibrium point can be found at $x = 0$ and $\mu = 0$. Like in case of saddle-node bifurcation and transcritical bifurcation we need to show that at the point $(0, 0)$ exist a non-hyperbolic equilibrium and so the equations 3 need to be fulfilled. Additionally the equations 11 need to be fulfilled to exhibit pitchfork bifurcation for the point $(0, 0)$ in case we have a non-hyperbolic equilibrium.

$$\frac{\partial X}{\partial \mu}(0,0) = 0 \qquad \frac{\partial^2 X}{\partial^2 x}(0,0) = 0 \qquad \frac{\partial^2 X}{\partial \mu \partial x}(0,0) \neq 0 \qquad \frac{\partial^3 X}{\partial^3 x}(0,0) \neq 0 \tag{11}$$

An example equation, which fulfills all criteria for pitchfork-bifurcation is given in equation 12 (compare [2, page 74–77]).

$$\dot{x} = \mu x - x^3 \tag{12}$$

For this example, setting $\mu = 0$ the only possible equilibrium point is at $x^* = 0$. By deriving the Jacobi Matrix $DX(x, \mu) = \mu - 3x^2$ and the Eigenvalue we find out that this point is not hyperbolic because of $DX(0, 0) = 0$. For $\mu \leq 0$, $x^* = 0$ is the only equilibrium point and it is asymptotically stable. For $\mu > 0$ we can find three equilibrium points.

$$x^* = \pm\sqrt{\mu} \qquad\qquad x^* = 0 \tag{13}$$

while the points at $x^* = \pm\sqrt{\mu}$ are stable, the equilibrium point at $x^* = 0$ is

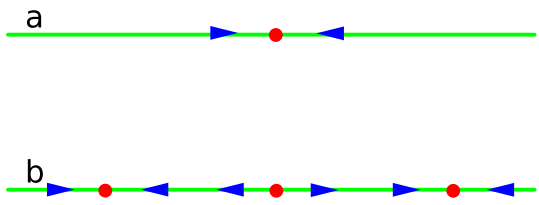

Fig. 8. Phase Portraits for a Pitchfork Bifurcation

not stable. Figure 8 shows the phase portrait for a pitchfork bifurcation where case `a` depicts the equilibrium point for $\mu < 0$ and case `b` depicts the three equilibrium points for $\mu > 0$. Figure 9 shows the bifurcation diagram for the pitchfork bifurcation including the information of the phase portrait shown in Figure 8 in combination with the bifurcation point (at the point $(0, 0)$). Generally a pitchfork bifurcation has the property that if a certain system is in an unique stable state with $\mu \leq 0$, if the value for μ changes to $\mu > 0$ it has two possibilites to reach again a stable state.

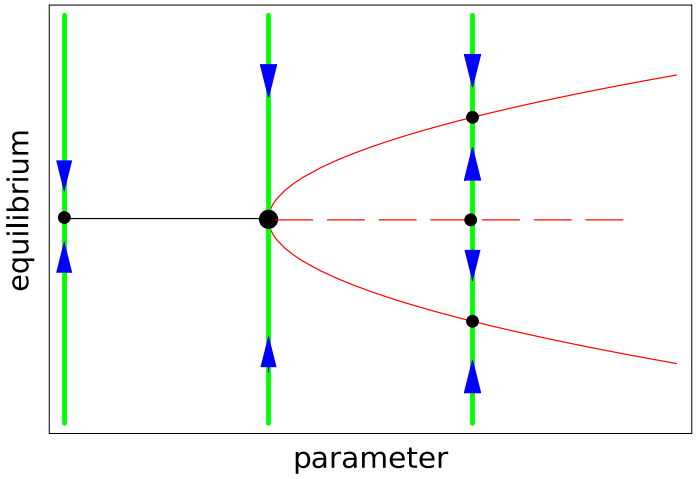

Fig. 9. Pitchfork Bifurcation Diagram

References

1. May, R.M.: Simple mathematical models with very complicated dynamics. Nature **261** (1976) 459–467
2. Boccara, N.: Modeling Complex Systems (Graduate Texts in Contemporary Physics). Springer (November 2003)

Equilibrium in size-based scheduling systems

Sebastien Soudan, Dinil Mon Divakaran

INRIA / Université de Lyon / ENS Lyon,
{Sebastien.Soudan,Dinil.Mon.Divakaran}@ens-lyon.fr

1 Introduction

Scheduling based on flow size (or flow age) has been gaining importance in the recent times. Researchers have proposed different ways of scheduling based on size, ranging from SRPT (Shortest Remaining Processing Time) to LAS (Least Attained Service) to MLPS (Multi-level Processor Sharing) scheduling mechanisms [1,2,3]. These scheduling strategies differ from the general model for flow scheduling in the Internet. The queues in the Internet nodes, though are served in an FCFS order at packet level, can be modeled using an M/G/1-PS (processor sharing) queue at flow level. The motivation to deviate from this norm, and schedule flows based on size, is to give better completion time to small flows. Strictly speaking, the aim has been to improve the conditional mean response time of small flows, at negligible cost to large flows. LAS, for example, always gives highest priority to the flow that has attained the least service. More details on size-based scheduling policies and the advantages they bring, can be found in [4] and [2]. Note that, researchers use *age-based scheduling* to refer to the scheduling schemes that are *blind*, in the sense that, they do not have information about the size of the flow when it arrives, and hence uses its age (the number of bytes/packets already scheduled) to make scheduling decision. Whereas, in this paper, we use the broader phrase *size-based scheduling* to include all the policies that use *age* or *size* to make scheduling decisions.

A user (an end-user or an application) sends a file as a single flow across the Internet. We take this as a normal behaviour. If size-based scheduling is deployed by an operator, there is a clear motivation for one or more users to deviate from the normal behaviour. Indeed, there is an incentive in splitting a flow (possibly large, but more precisely, one that is not small) into multiple small flows to exploit the advantage (say, priority in scheduling) given to small flows to improve the response time. If a considerable number of users deviate from the normal behaviour, then the operator's aim of giving shorter response time to small flows might well be deceived. More importantly, an operator would like to know if such user manipulations would lead to an unstable system behaviour. This poses an important problem in the context of size-based scheduling systems which, to the best of our knowledge, has not been addressed yet. This is the problem we address in this work. In the scenario where users do not misbehave, the stability issue (for network of queues) has been addressed in [5] recently.

The focus of this work is to study the equilibria in size-based scheduling system where users misbehave. We believe this would lead to better understanding

of the implication of deploying a size-based scheduling mechanism. More description of the problem is given in Section 2. The model is elaborated in Section 3. The existence of equilibria is studied in Section 4, for the case in which the service rates are fixed.

2 Problem statement and assumptions

We study the problem that arises when an operator deploys a size-based scheduling mechanism. Though there are different ways of scheduling based on size, our focus is on size-based scheduling using two queues. Here, flows are classified based on their sizes. Small flows are sent to one queue, and large flows to another[1]. Each queue is assigned a specific service rate, such that the total service rate equals the line capacity. The aim of operator in setting such a mechanism is to give to reduce the average response times of small flows.

To formulate the objective of the operator, we assume Poisson flow arrivals. Arrivals and service rates are in units of small flow. λ_x and λ_y are the arrival rates for small and large flows respectively. Each large is F times a small flow. The service rates at small and large queues are ϕ_x and ϕ_y respectively, such that if C denotes the line capacity, $\phi_x + \phi_y = C$. Each queue is served using the PS discipline; hence it is an $M/G/1 - PS$ queue.

We study the existence of equilibria under the scenario where users *cheat* by splitting a large flow into multiple small flows to improve their delay. This is explored in two cases: (i) where the service rates assigned are static, (ii) where the operators exhibits control by dynamically changing the service rates. In the latter case, we explore the existence of interesting equilibria, and state the conditions required for stability, under the assumption that the incentive for players to migrate is to minimize the delay the flow will incur. Note that, by 'players', we consider only the users who migrate.

3 Model description

The fluid model used in this work is inspired by the one used in [6], where the authors analyse dynamic bandwidth resource allocation and migration between *guaranteed performance* and *best effort* traffic classes.

The two-queues model is depicted in Fig. 1. The queue for small flows is called *small queue* and is referred to as Q_x. The other queue is called the *large queue* which is denoted by Q_y. The number of flows at Q_x is represented by x. At the large queue, this number (in number of small flows) is denoted by y. We assume infinite queues. The service rates, ϕ_x and ϕ_y, are also in number of small flows. They are both assumed to take non-zero values.

[1] A flow is called small if its size is less than a threshold, θ. In practice, θ bytes of every large flow also go to the small queue. But, we ignore this to keep the model simple. Besides, this affects neither the analysis nor the results given here.

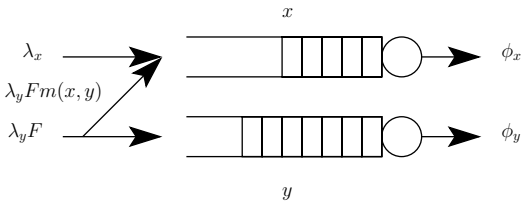

Fig. 1. Two-queues model.

The system parameters ϕ_x and ϕ_y are set by the operator. System state is modeled using averaged queue sizes: x and y. Depending on the measured delay values, a user might decide to split a large flow into multiple small flows. Therefore, a fraction of the flows arriving at the large queue might be *migrated* to the small queue. This migration function, which is a result of aggregate user behaviour, is represented as $m(x, y)$. It is linear in $\lambda_y F$ as a result of the integration of individual user that send $\mathrm{d}\lambda_y$ each: $\int m\mathrm{d}\lambda_y = \lambda_y m$. We take m to be a non-negative and continuous function of x and y. m represents the fraction of λ_y which goes to Q_x.

$$0 \leq m(x, y) \leq 1 \tag{1}$$

For every large flow that migrates, it adds an overhead of η (e.g. connection establishment cost, slow-start cost). The rate equations can now be written as:

$$\frac{dx}{dt} = \lambda_x - \phi_x + \lambda_y F m(x, y)(1 + \eta), \quad x > 0 \tag{2}$$

$$\frac{dy}{dt} = \lambda_y F - \phi_y - \lambda_y F m(x, y), \quad y > 0 \tag{3}$$

The rate equations are different at the borders. For $x = 0$,

$$\left.\frac{dx}{dt}\right|_{x=0} = [\lambda_x - \phi_x + \lambda_y F m(x, y)(1 + \eta)]^+ \tag{4}$$

and for $y = 0$,

$$\left.\frac{dy}{dt}\right|_{y=0} = [\lambda_y F - \phi_y - \lambda_y F m(x, y)]^+. \tag{5}$$

4 System analysis for static service rates

This section details the analysis of a system where the service rates at both the queues are fixed.

Proposition 4.1 *An interior point (x, y) is an equilibrium iff $\phi_x - \lambda_x = \lambda_y F - \phi_y$ and m is such that $m(x, y) = \frac{\phi_x - \lambda_x}{\lambda_y F}$ and $0 \leq m(x, y) \leq 1$.*

Proof (Proof of Prop. 4.1).
Let (x, y) be an interior point. It is an equilibrium if and only if :

$$\begin{cases} \frac{dx}{dt} & = 0 \\ \frac{dy}{dt} & = 0 \\ 0 \leq m(x,y) \leq 1 \end{cases} \iff \begin{cases} m(x,y) = \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ m(x,y) = \frac{\lambda_y F - \phi_y}{\lambda_y F} \\ 0 \leq m(x,y) \leq 1 \end{cases}$$

□

Remark 4.2 *Existence of interior equilibrium does not only depend on m function but also on the arrival rates and service rates. Meaning that they can only exist in very specific cases.*

Proposition 4.3 $(0,0)$ *is an equilibrium point if and only if:*

$$\begin{cases} m(0,0) \leq \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ \frac{\lambda_y F - \phi_y}{\lambda_y F} \leq m(0,0) \\ 0 \leq m(0,0) \leq 1 \end{cases}$$

Proof (Proof of Prop. 4.3). Using equations (4) and (5), we obtain that $(0,0)$ is an equilibrium point if and only if:

$$\begin{cases} \frac{dx}{dt}\Big|_{x=0} & = 0 \\ \frac{dy}{dt}\Big|_{y=0} & = 0 \\ 0 \leq m(0,0) \leq 1 \end{cases} \iff \begin{cases} \frac{\lambda_x - \phi_x}{1+\eta} + \lambda_y F m(0,0) \leq 0 \\ \lambda_y F - \phi_y - \lambda_y F m(0,0) \leq 0 \\ 0 \leq m(0,0) \leq 1 \end{cases}$$

$$\iff \begin{cases} m(0,0) \leq \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ \frac{\lambda_y F - \phi_y}{\lambda_y F} \leq m(0,0) \\ 0 \leq m(0,0) \leq 1 \end{cases}$$

□

Proposition 4.4 $(0,y)$ *with* $y > 0$ *is an equilibrium point if and only if:*

$$\begin{cases} m(0,y) \leq \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ m(0,y) = \frac{\lambda_y F - \phi_y}{\lambda_y F} \\ 0 \leq m(0,y) \leq 1 \end{cases}$$

Proof (Proof of Prop. 4.4). Using equations (4) and (3), we obtain that $(0,y)$ is an equilibrium point if and only if:

$$\begin{cases} \frac{dx}{dt}\big|_{x=0} & = 0 \\ \frac{dy}{dt} & = 0 \\ 0 \leq m(0,y) \leq 1 \end{cases} \iff \begin{cases} \frac{\lambda_x - \phi_x}{1+\eta} + \lambda_y F m(0,y) \leq 0 \\ \lambda_y F - \phi_y - \lambda_y F m(0,y) = 0 \\ 0 \leq m(0,y) \leq 1 \end{cases}$$

$$\iff \begin{cases} m(0,y) \leq \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ m(0,y) = \frac{\lambda_y F - \phi_y}{\lambda_y F} \\ 0 \leq m(0,y) \leq 1 \end{cases}$$

□

Proposition 4.5 *$(x, 0)$ with $x > 0$ is an equilibrium point if and only if:*

$$\begin{cases} m(x,0) = \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ \frac{\lambda_y F - \phi_y}{\lambda_y F} \leq m(x,0) \\ 0 \leq m(x,0) \leq 1 \end{cases}$$

Proof (Proof of Prop. 4.5). Using equations (2) and (5), we obtain that $(x, 0)$ is an equilibrium point if and only if:

$$\begin{cases} \frac{dx}{dt} & = 0 \\ \left.\frac{dy}{dt}\right|_{y=0} & = 0 \\ 0 \leq m(0,y) \leq 1 \end{cases} \iff \begin{cases} \frac{\lambda_x - \phi_x}{1+\eta} + \lambda_y F m(x,0) = 0 \\ \lambda_y F - \phi_y - \lambda_y F m(x,0) \leq 0 \\ 0 \leq m(x,0) \leq 1 \end{cases}$$

$$\iff \begin{cases} m(x,0) = \frac{\phi_x - \lambda_x}{\lambda_y F(1+\eta)} \\ \frac{\lambda_y F - \phi_y}{\lambda_y F} \leq m(x,0) \\ 0 \leq m(x,0) \leq 1 \end{cases}$$

□

4.1 Discussion

The aim of a network operator in deploying such a scheduling mechanism is to give shorter delays to small flows, at negligible cost to large flows. With this in mind, we can now evaluate which among the equilibrium points are interesting and useful (from the perspective of a network operator).

To start with, let us consider the equilibrium point $(0, 0)$. The inequalities of Prop. 4.3 give the shaded region of Fig. 2, where one m can exist to make $(0, 0)$ an equilibrium. This region is dominated by the line $\lambda_x + \lambda_y F = C$, which defines the region where a single queue system would have empty queue equilibrium. Thus, this equilibrium (in the two queue system) is not of great interest for the network operator.

The lines $(x, 0)$ and $(0, y)$ constitute the remaining border point equilibria. $(x, 0)$ is the set of those points where there is queueing in the small queue, but not at the large queue. For this reason, these are not desirable equilibria from operator's point of view. Similarly existence of $(0, y)$ means, there is nothing queueing at Q_x. So, there is incentive for users to migrate to Q_x. Hence $(0, y)$ will not be stable.

5 Conclusions and future work

As seen in previous section, interior point equilibrium are only possible in limiting cases where the surplus rate at the large queue is exactly equal to the surplus of service of x, with the additional constraint that m transfers exactly this. This situation is too constrained to happen in a real scenario. To introduce more flexibility, the operator can control the service rate. But this requires the

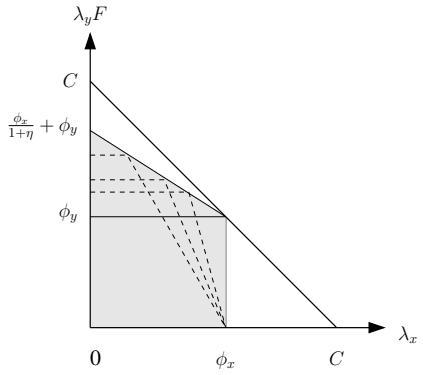

Fig. 2. Existence region of equilibrium $(0, 0)$ under static service rate.

use of some observable parameters of the system. In this system, the only observable parameters are x and y as arrival rates λ_x and λ_y are not separable at the queues.

References

1. L. Kleinrock, *Queueing Systems, Volume II: Computer Applications*. Wiley Interscience, 1976.
2. K. Avrachenkov, U. Ayesta, P. Brown, and E. Nyberg, "Differentiation Between Short and Long TCP Flows: Predictability of the Response Time," in *Proc. IEEE INFOCOM*, 2004.
3. C. Sun, L. Shi, C. Hu, and B. Liu, "DRR-SFF: A Practical Scheduling Algorithm to Improve the Performance of Short Flows," in *ICNS '07: Proceedings of the Third International Conference on Networking and Services*, 2007, p. 13.
4. M. Nuyens and A. Wierman, "The foreground-background queue: A survey," *Perform. Eval.*, vol. 65, no. 3-4, pp. 286–307, 2008.
5. P. Brown, "Stability of networks with age-based scheduling," in *Proc. IEEE INFOCOM*, 2007, pp. 901–909.
6. E. Altman, A. Orda, and N. Shimkin, "Bandwidth allocation for guaranteed versus best effort service categories," *Queueing Syst. Theory Appl.*, vol. 36, no. 1-3, pp. 89–105, 2000.