DISCRETE AND CONTINUOUS
DYNAMICAL SYSTEMS
Supplement 2009

Website: www.aimSciences.org

pp. **404–415**

INTELLIGENT TRAFFIC CONTROL ON INTERNET-LIKE TOPOLOGIES - INTEGRATION OF GRAPH PRINCIPLES TO THE CLASSIC RUNGE–KUTTA METHOD

Antonia Katzouraki

Electrical & Electronic Engineering, Imperial College London, UK

Tania Stathaki

Electrical & Electronic Engineering, Imperial College London
SW7 2AZ, UK

Abstract. 'No man is an island' [John Donne]. Human and technological networks play a vital part in our lives, and their failures have often caused severe adverse consequences. In this paper we address this crucial issue by presenting a model to prevent not only network failures but also their propagation to the remaining network elements. Our model forecasts the number of packets each node is able to service without becoming overloaded, by determining the transition probabilities assigned to each link. Thus, our model ensures that nodes receive as many packets as their network resources prescribe. The model is portable to any type of topology and is based on Ordinary Differential Equations (ODEs), which are numerically solved as a multivariable, coupled system, over a variety of topologies. Our numerical algorithm is based on the classic Runge–Kutta 4th order, which is adjusted to integrate graph principles.

1. **Introduction.** Real world events have shown that network load can be affected, either due to a physical catastrophe or unexpected increase of the number of clients requesting service. A characteristic example is that of the 10th of August 1996, when a voltage-based failure occurred in two power lines. Major power disturbances were propagated through a series of cascading failures, ultimately leading to blackouts in 11 US states and 2 Canadian provinces, leaving 7 million people without power for up to 16 hours [20]. In such cases, a network provider must be able to quickly dynamically balance the load (network's demand and supply) according to the available network resources and prevent a cascading failure.

To date there have been continuous advances in Internet research along with its connectivity structures [1]. In particular, recently there has been much interest in studying cascading failures in complex networks. Initially Motter and Lai suggested a capacity model (ML) to describe cascade failures [15]. According to this model, the capacity is assigned in proportion to the load. Later, an analytical calculation of the capacity parameter was investigated [21]. Additionally, a costless strategy of defence was introduced and investigated to control and reduce the size of the cascade failures [14]. Recently, an improvement of the existing ML capacity model was suggested, in which the model is further generalized so that the proportionality

2000 *Mathematics Subject Classification.* Primary: 58F15, 58F17; Secondary: 53C35.
Key words and phrases. Ordinary Differential Equations, Equilibrium Model.
The authors are supported by DTC grant.

constant is now changed to an increasing function of load [19]. In other research, excitable networks have been modeled by Hodgkin–Huxley [12], Bonhoeffer–van der Pol-FitzHugh–Nagumo (BPFN) [4, 3] and Kuramoto oscillators [16, 17], to investigate performance in terms of activity and the subsequent synchronization of the coupled excitable elements. In a recent work, the performance of BPFN oscillators coupled on Small–World networks [8] was investigated to find that there is no simple dependence on the network topology, instead, it is affected by the complex pattern of interactions throughout the entire network [18]. Furtheremore, Frank Kelly derived a mathematical model based on ODEs to analyze the stability and fairness of a simple rate control algorithm defining a dynamical system [10, 11]. D'apice et al investigate a macroscopic fluid model on telecommunication networks with sources and destinations [2], whereas Marigo investigates equilibrium solutions for data flows on networks [13].

In this paper we present a capacity model, to balance network load according to the available resources. A set of ODEs relates the capacity to the load. Our model finds the best possible operating points for both traffic within each node and transition probabilities assigned to each link. In particular, we assign two ODEs to each network node to describe the change of both the number of packets within each node and the transition probabilities assigned to each link (the probability that a packet will travel across the link). Therefore, the number of ODEs we need to numerically solve is at least twice the size of our network. The numerical solution provides the number of packets each node is able to comfortably accommodate and we call this the equilibrium point; the point at which the load is balanced amongst the nodes. We find that network elements are exchanging packets in both a coherent and synchronized fashion according to their primitive (network resources) characteristics and their connectivity within the network. We experimentally find the relation between the network resources, such as buffer capacity and service ability and the equilibrium point that each node achieves. It is vital to properly assign network resources without wasting them, in order to design practical stable networks. Through these findings we aim to detect network failures and automatically seek the equilibrium points, which have a central role in preventing cascading network failures. In this paper we explain our numerical algorithm through which we integrate graph principles (network's connectivity) to classic Runge–Kutta 4th order, in order to synchronize the coupled network elements and relate network resources to each node's equilibrium point.

2. **The Model's Formulation.**

In the following paragraphs, we outline our model and the parameters that it depends on, along with the probabilistic and deterministic elements from which it is built. Consider single nodes $i = 1, 2, \cdots, N$ and let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a finite directed graph, where $\mathcal{V}$ is the set of network nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, the set of unidirectional links. Let $A = [a_{ij}]$ be an ad-hoc adjacency matrix, which is completely specified by an $N \times N$ matrix. It consists of zeros and ones, such that each entry $a_{ij} = 1$ represents a connection between the nodes i and j, whereas 0 implies there is no connection between the respective nodes. The adjacency matrix is continuously updated, so that network connectivity is changed.

The main parameters of our model are illustrated in Table 1. In particular, a set of N variables, $x_1, x_2, ..., x_N$ describe the traffic (node occupation e.g. bytes). Consider m the number of incoming links towards a single node $i \in \mathcal{V}$. A set of

$m \in \mathcal{E}$ variables $c_1, c_2, ..., c_m$ describe the cost on the $m \in \mathcal{E}$ incoming unidirectional links. Both the traffic, x_j, and link–cost, c_{ij} parameters depend on time. Each node and link have a number of initial–packets $[x_i]^0$ and $[c_{ij}]^0$ respectively, given for the time-step $t_n = 0$. A set of $m \in \mathcal{E}$ variables $\lambda_1, \lambda_2, \cdots, \lambda_m$ determine the fraction of traffic relayed through m unidirectional links, which constitute the coupling elements of our model. In this way the coupled network elements (links and nodes) are synchronized through these time dependent transition probabilities, λ_i, $\forall\ i \in [1, \cdots, m]$. These probabilities ensure that nodes receive as many packets as they can comfortably process. Thus transition probabilities toward almost overloaded nodes are decreased towards zero, to correspondingly decrease the load of the almost overloaded node. Every node is assigned a service power, $s_i\ \forall\ i \in \mathcal{V}$, which describes the ability each node has to process packets. The buffer capacity variable, b_i, $\forall i \in \mathcal{V}$ describes the maximum number of packets, within a node. Both the buffer capacity and service power parameters are independent of time. These built-in features allocated to each node operate as control parameters since their values affect both the solution curves and equilibrium point. In particular through the term $s_i \cdot (1 - \frac{x_i}{b_i})$ in the traffic Equation (2), which is introduced in a following paragraph both the buffer capacity and service power determine the pool from which packet rates take their values. A node without outgoing links $(i, j) \in \mathcal{E}$ should not attract more packets, and thus its service power should eventually become zero. With regard to the probability distribution of traffic arriving at a particular incoming link $(i, j) \in \mathcal{E}$, the following two conditions must hold: (I) The total mass of its elements is unity. (II) The probability of traffic arriving at an incoming link is inversely proportional to its current traffic load. The second condition on the transition probabilities ensures that each of the network elements is neither idle nor overloaded. Thus, transition probabilities follow the principle that the higher the cost, the fewer packets are sent through the link. This is mathematically implemented by setting each transition probability inversely proportional to the respective link cost. The transition probability for each link is denoted as an element of the $N \times N$ matrix, $\Lambda = [\lambda_{ij}]$, which is defined such that $\lambda_{ij} \in [0, 1]$ with: $\lambda_{ij} := \sum_{j \in \mathcal{V}} \lambda_{ij} = 1, \forall i \in \mathcal{V}$, and is determined by the relation,

$$\lambda_{ij} = \frac{\frac{1}{c_{ij}}}{\sum_{k \in \mathcal{V}} \left[\frac{1}{c_{ik}}\right]},\ \forall\ (i,\ j) \in \mathcal{E} \text{ such that } c_{ik} \neq 0\ \forall\ k \in \mathcal{V}. \tag{1}$$

The aforementioned relation not only depends upon time t, but also on both cost, $c_{ij}, \forall$ link $(i, j) \in \mathcal{E}$ and node occupation (traffic), $x_i, \forall\ i \in \mathcal{V}$. This is because c_{ij} depends on x_i, through the following Equations (2) and (3). The non–negative real–valued traffic variable, $x_j \in \Re^{+*}$, $\forall$ node $j \in \mathcal{V}$, describes the change in the number of packets within each node $j \in \mathcal{V}$. The traffic rate is given by

$$\frac{dx_j}{dt} \quad = \quad \sum_{i=1}^{N} \lambda_{ij} s_i x_i \left(1 - \frac{x_i}{b_i}\right) - s_j x_j \left(1 - \frac{x_j}{b_j}\right)\ . \tag{2}$$

Through the non–negative link–cost variable, $c_{ij} \in \Re^{+*}\ \forall$ link $(i, j) \in \mathcal{E}$, we calculate the number of packets that occupy each link, $(i, j) \in \mathcal{E}$. Each link–cost variable is modeled by a nonlinear function, in order to deliver the least possible

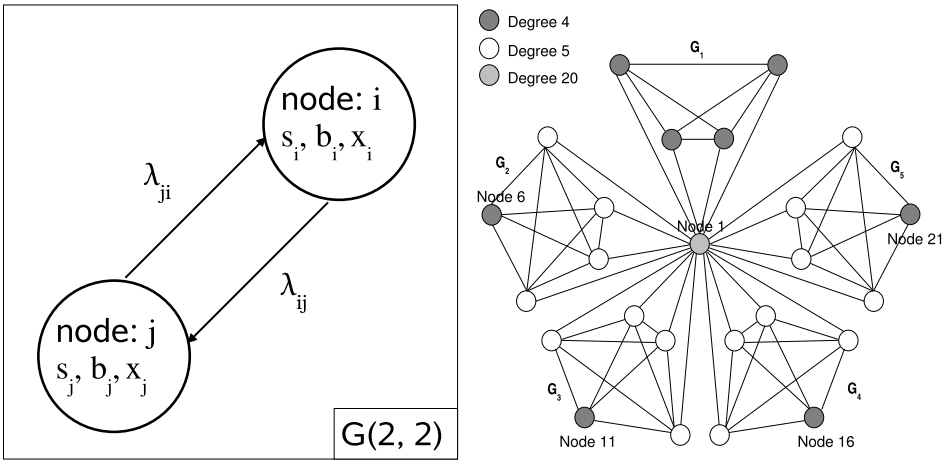

FIGURE 1. **(a)**. Illustration of a 2–node graph and **(b)**. a 25–node Ravsz-Barabasi graph.

congested service. The link–cost rate is given by

$$\frac{dc_{ij}}{dt} = \lambda_{ij} \cdot \left[\frac{s_i \cdot (x_j)^2}{(1+(x_j)^2)} - \frac{s_j \cdot (x_i)^2}{(1+(x_i)^2)} \right] , \tag{3}$$

and describes the cost $\forall$ incoming link $(i,j) \in \mathcal{E}$, that is transferring traffic from node $i \in \mathcal{V}$ to node $j \in \mathcal{V}$.

TABLE 1. Model's Parameters

Time dependent parameters	**Control parameters (independent of time)**
Number of packets within node i: x_i $\forall i \in \mathcal{V}$ Traffic — $1 \times N$ matrix	Buffer capacity for node i: b_i, $i \in \mathcal{V}$ $1 \times N$ matrix
Number of packets within link (i,j): c_{ij} $\forall (i,j) \in \mathcal{E}$ Link–cost — $N \times N$ matrix	Power ability of node i: s_i, $i \in \mathcal{V}$ $1 \times N$ matrix
Transition Probability — Coupling element $\Lambda = [\lambda_{ij}], \forall (i,j) \in \mathcal{E}$: $N \times N$ matrix	

The concept behind the introduced link-cost function is that 'Internet traces come into bursts of bytes' [6], whilst in social networks 'people bring more people'. This is mathematically described by the sigmoid curve. This ensures the network responds rapidly to large influxes of packets, without overreacting to small changes in activity. In the link-cost function described in Equation (3), small increases of traffic cause a very small increase in the cost (few people/bytes would not change the cost), whereas slightly higher volumes of traffic (large crowd of people/burst of bytes) would initiate a more dramatic increase in cost, in order to accommodate the number of people/bytes.

3. **Integrating Graph Principles to Classic Runge–Kutta method.** Here, the traffic and link–cost ODEs (2) and (3), are coupled to lead to a large system and further generalized through their application to multiple nodes. We modify the classic Runge–Kutta method to include network connectivity and synchronise the

coupled network elements (nodes and links). An implicit solution is more efficient, but also computationally more expensive as it evaluates more functions per time step. We consider it to be beyond the scope of this paper to apply an implicit method. Instead, an explicit solution for x_{n+1} and c_{n+1} in terms of known values at t_n is provided. The application of a lower order method, such as Euler or its improved version, also known as Runge-Kutta second order, would be less accurate compared with Runge-Kutta forth order. A more accurate approximation, through the application of Euler, would require the decrease of the time step. This decrease would increase the amount of steps required to reach the desired accuracy. Furthermore, short step sizes may lead to an accumulation of errors due to arithmetic truncation error through finite precision arithmetic. In the introduced algorithm we apply Runge-Kutta forth order and the time step is of the size 10^2. The introduced algorithm provides the same accuracy in results with the smaller time step, such as 10^6, but without the cost of taking increased time and using more computational power.

Our model is applied to directed graphs with bidirectional links. However, throughout the numerical analysis we choose the cheapest direction on each link, which is capable of transferring more traffic and has the least potential to create link–bottlenecks during the traffic distribution on the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. For example, for the case of a link (i, j) through which we transfer packets either from node i to j or node j to i, we choose the cheapest direction. Thus, node i either forwards packets to node j or receives packets from node j but not both. After preventing bottlenecks within links, we search for nodes without service power to prevent bottlenecks within nodes. A node $i \in \mathcal{V}$ without outgoing links is either deactivated due to malfunction or has no available service power for new packets. Thus, in order to stop this node from receiving more packets we set service power $[s_i]$, $i \in \mathcal{V}$ to zero. Our model relates the buffer capacity and service power of each node as a form of the traffic load (number of packets). It is applied to a static network (with respect to the number of packets injected), in which nodes start exchanging packets until they reach equilibrium. Without loss of generality, this version of our model lacks sink (more incoming flow) and source (more outgoing flow) nodes.

We simultaneously numerically solve the coupled traffic and link–cost differential equations. Restrictions are set on the time dependent (traffic and link–cost) variables, to ensure non–negative values for both the traffic and link–cost variables. We then calculate the current number of incoming links (m) for each node, to find the number of coupled ODEs that describe the state of the studied node. In particular, the number of ODEs required to be solved is at least twice the size of our network. In other words, it is m link-cost equations and one traffic equation, through which the state of a single node is described. This large system is numerically solved through the classic Runge-Kutta formula i.e. explicit form of the 4th order [7], which is adjusted in order to include the transition probability matrix $[\lambda_{ij}]$. Thus, after the calculation of each traffic and link-cost slope approximation we readjust the link-cost values and recalculate the transition probabilities to ensure transfer of the appropriate quantity of packets in the most desirable direction.

In the following section, we present numerical solutions from both the original version of the introduced *Dynamically Adjusted Traffic Rate Alterations* (DATRA) model [9] and its second version, DATRAv2, which is presented in this paper. The significant difference is that however dynamics are described by the same set of ODEs, those ODEs are solved as two single variable problems in the first version,

whereas in the second one as one multivariable problem. According to the first version of the introduced model, Runge-Kutta is deployed to solve the traffic ODE while the link cost is always inversely proportional to the traffic. Furthermore, Runge–Kutta is deployed for second time in order to solve the link–cost ODE. Thus in the first version of the introduced model traffic and link cost ODEs are solved as two single–variable problems, whereas in the second version are solved as one multivariable problem.

According to the first version of the introduced algorithm, the first single–variable problem is summarised through the following recursion formulas:

$$\begin{aligned} c_{ij}^{n} &= \frac{1}{x_j^n}, \\ \lambda_{ij}^{n} &= \frac{1}{c_{ij}^{n}}, \quad \sum \lambda_{ij}^{n} = 1, \\ c_{ij}^{n+1} &= c_{ij}^{n} + h, \\ x_j^{n+1} &= x_j^n + \frac{1}{6} \cdot (s_1^n + 2 \cdot s_2^n + 2 \cdot s_3^n + s_4^n), \end{aligned}$$

whereas the second single–variable problem is summarised through the following recursion formulas:

$$\begin{aligned} x_j^{n+1} &= x_j^n + h, \\ c_{ij}^{n+1} &= c_{ij}^{n} + \frac{1}{6} \cdot (k_1^n + 2 \cdot k_2^n + 2 \cdot k_3^n + k_4^n) \ . \end{aligned}$$

According to the second version, the model is solved as one multivariable problem, which is described by the following recursion formulas:

$$\begin{aligned} x_j^{n+1} &= x_j^n + h/6 \cdot (s_1^n + 2 \cdot s_2^n + 2 \cdot s_3^n + s_4^n), \\ \lambda_{ij}^{n+1} &= \frac{1}{c_{ij}^{n}}, \quad \sum \lambda_{ij}^{n+1} = 1, \\ c_{ij}^{n+1} &= c_{ij}^{n} + h/6 \cdot (k_1^n + 2 \cdot k_2^n + 2 \cdot k_3^n + k_4^n), \end{aligned}$$

In the following sections without loss of generality and for reasons of simplicity, we illustrate numerical results from our model applied to graphs of two nodes, Figure 1.a and the well known Ravsz–Barabasi graph of twenty–five nodes [5] in Figure 1.b. We analyze the experimental results to demonstrate that: (I) The number of equilibrium–packets at each node depends on every node's buffer capacity and service power; (II) Both the solution curves and the equilibrium points are synchronized due to the coupling element between nodes which depends on their connectivity; (III) The nature of the solution curves depends on both the buffer capacity and the service power assigned to each node.

4. **Numerical Results for a 2–Node Graph, $\mathcal{G}(2,2)$.** Consider the simplest graph $\mathcal{G}(2,\,2)$ depicted in Figure 1.a, that consists of two nodes and two unidirectional links. We apply DATRAv2 to illustrate the behaviour of the traffic and its

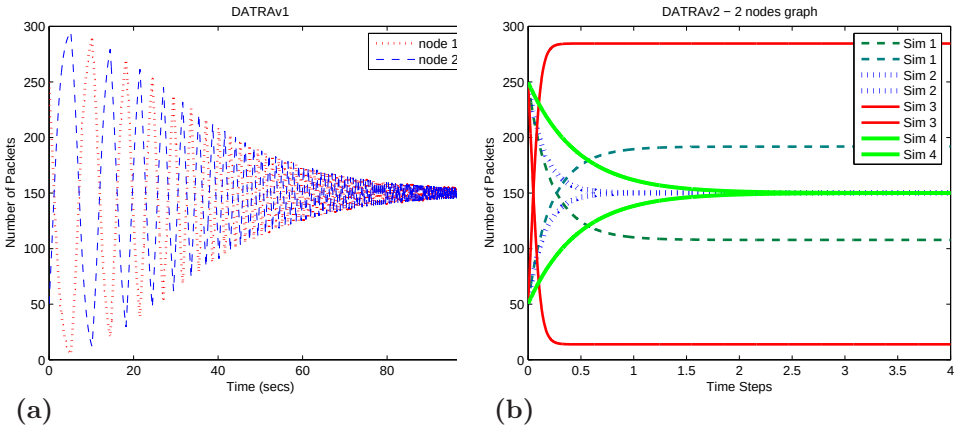

FIGURE 2. **(a)** DATRAv1 model applied to a 2–node graph — solution curves for each node separately. **(b)** DATRAv2 runs on a 2–node graph $\mathcal{G}(2,\,4)$. Five pairs of data solutions are illustrated.

rate over time for each node $i \in \mathcal{G}(2,\,2)$. The control parameters are listed as follows. Link–cost capacities: $[c_{11}]^0 = 0$, $[c_{12}]^0 = 250$, $[c_{21}]^0 = 50$, $[c_{22}]^0 = 0$. Number of initial packets $[x_1]^0 = 250$, $[x_2]^0 = 50$. Each time we numerically solve our model, we vary the control parameters accordingly to reach different equilibrium points. In Figure 2.b, we present four pairs of the numerical solutions of the Equations (2) and (3) applied to the graph $\mathcal{G}(2,\,2)$. We explain which control parameter is changed for each example, in order for each node to reach a different equilibrium point.

For the first two examples we keep the same network characteristics, only changing the buffer capacities. For the first example we set buffer capacities $[b_i] = [350, 350]$ and service power, $[s_i] = [10, 10]$, equal for each node $i \in \mathcal{G}$. The numerical results for this example are illustrated by the dotted lines in Figure 2.b. We observe that both nodes reach the same equilibrium point, at which they maintain the same number of equilibrium–packets. In the second example we set different buffer capacities for each node, $[b_i] = [550, 350]$. Results from this example are illustrated by the solid lines in Figure 2.b. Each of these solid curves reaches different equilibrium points, which are inversely proportional to their buffer capacity.

In the third example dash-dot lines illustrate numerical results, through which we observe that by increasing the service power to $[s_i] = [30, 30]$ the system reaches the same equilibrium point as the dotted solution curves, but this time more quickly. In the last example buffer capacities are set to be the same, whereas the service rates are unequal, $[s_i] = [40, 10]$. This time (dashed solution curves) we observe a significantly faster response time with introduced overshoot, which is triggered by the difference between each node's service power. In Figure 2.b we observe that according to the dashed lines the number of packets within node 1 is dramatically decreased, jumping from 250 packets to 100 packets within the first 0.25 seconds.

Application of the original version of our numerical algorithm [9]. In this example we consider the same graph, $\mathcal{G}(2, 2)$ with homogeneously distributed control parameters, similar with the example in Section 4. We provide experimental evidence that DATRAv2 achieves equilibrium 100 times more quickly than the original DATRAv1 algorithm. In Figure 2.a, solution curves for nodes one and two are produced by our original model DATRAv1, symmetrically oscillate around

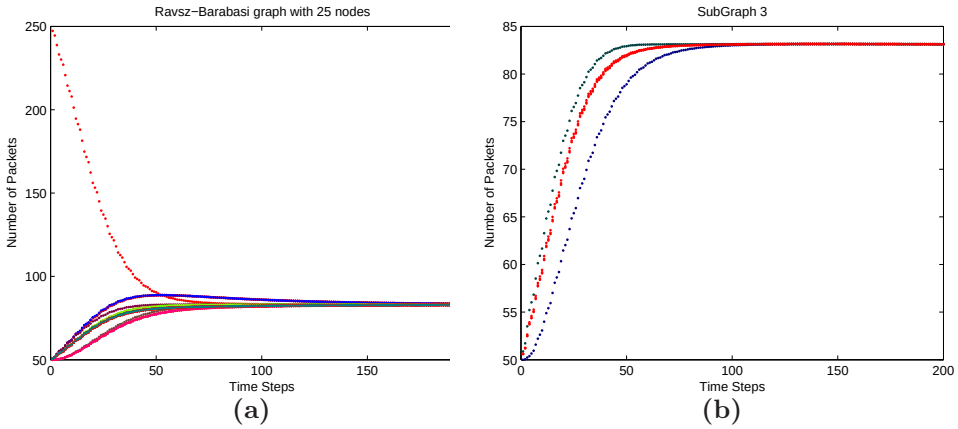

FIGURE 3. Example 1: **(a)** 25 node Ravsz-Barabasi graph with homogeneously distributed built-in attributes and initial-packets. **(a)** Node 11 is the only node within G_3 which is not connected to the *core node*, and thus its solution curve demonstrates a phase shift, and it receives fewer packets, resulting in a lower amplitude.

the equilibrium point, whilst their amplitude is exponentially reduced. According to this model the traffic and link-cost equations are solved as two single variable problems using the classic Runge–Kutta in the explicit form with forth order [7]. Therefore, we first numerically analyze the traffic equation always keeping the link cost value inversely proportional to the traffic value and respectively calculating the transition probabilities. The output of the traffic ODE is the input of the link–cost. Therefore, each time we numerically solve the link–cost ODE, we have a dramatic increase or decrease in the number of packets within each node. These dramatic changes produce oscillations around the equilibrium point. In Figures 2a. and 2.b we also observe that both nodes achieve the same equilibrium point at 150 packets. Finally, we observed that in both examples, independent of the algorithm deployed, the equilibrium point contains the same quantity of packets, 150.

5. **Numerical Results on a 25–node Ravasz–Barabasi Graph.** Consider the well known Ravasz–Barabasi graph of 25 labeled nodes with two hierarchical levels and a scale–free degree distribution [5], which is depicted in Figure 1.b. We split the 25–node Ravasz–Barabasi graph into five subgraphs, $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4, \mathcal{G}_5$. In particular, $\mathcal{G}_1$ consists of four nodes with degree four, whereas each of the other subgraphs, $\mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4, \mathcal{G}_5$ consist of five nodes, out of which three of them have degree four and two of them have degree five.

We present numerical results from three examples, in which we vary the control parameters accordingly. For the first example, we follow a homogeneous distribution of the buffer capacity and service power. We demonstrate that networks with homogeneously distributed attributes and the same connectivity, achieve equilibrium with the same number of equilibrium–packets. In Figure 3.a we split the solution curves into four categories according to the amplitude of the solution curve. The first category consists of the curve where the number of packets is decreased almost exponentially. In this category, outgoing link–costs are cheaper compared with incoming link–costs. This curve is produced by node 1 due to the large number of

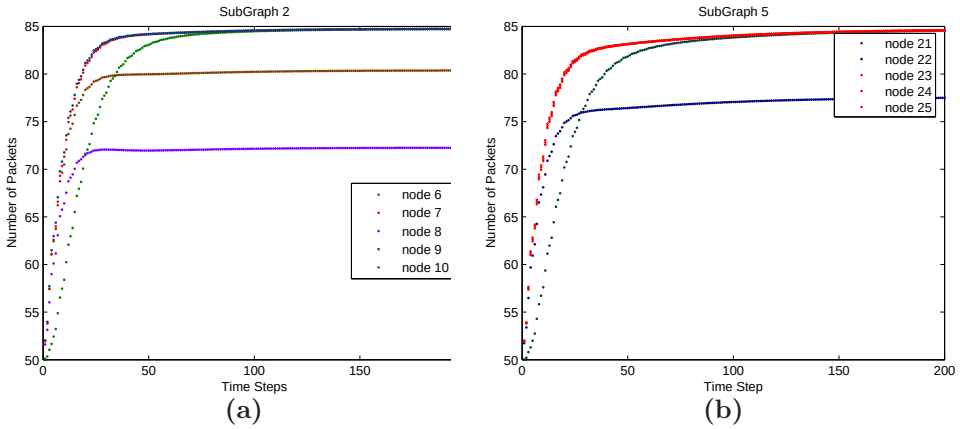

FIGURE 4. Example 2: Ravsz-Barabasi graph, $\mathcal{G}_2$, $\mathcal{G}_5$: non-homogeneously distributed buffer capacities.

initial–packets. The second category consists of the solutions, which are produced by $\mathcal{G}_1$. As this graph consists of four homogeneous nodes (same connectivity and control parameters) their curves are also homogeneous. We observe that the $\mathcal{G}_1$ subgraph receives a larger number of packets compared with $\mathcal{G}_2$, $\mathcal{G}_3$, $\mathcal{G}_4$ and $\mathcal{G}_5$. This larger amount of traffic is created due to the fact that $\mathcal{G}_1$ has better connectivity with node 1, as all nodes $\in \mathcal{G}_1$ are connected directly to node 1. The third category consists of curves for nodes $\in \{\mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_3, \mathcal{G}_5\}$, which are connected to node 1. The final category consists of solutions that oscillate with the minimum amplitude. These solution curves are produced from nodes $\in \{\mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_3, \mathcal{G}_5\}$ which are not connected to node 1. In Figure 3.b curves from nodes that belong to the second subgraph $\mathcal{G}_2$ are illustrated. Node 6 is the only node within $\mathcal{G}_2$, which is not connected to node 1. Its curve demonstrates a phase shift, as it receives fewer packets.

In the second example we distribute the buffer capacities of each node in a non–homogeneous fashion. Through this simulation we present the relationship between the buffer capacity and the equilibrium that each node achieves. The higher the buffer capacity assigned, the lower the equilibrium point achieved. As a result of the non–homogeneous distribution of the buffer capacities, each of the nodes achieves a different equilibrium, which depends on its buffer capacity. Figure 4.a demonstrates the solution curves for each of the nodes that belong to the second subgraph, $\mathcal{G}_2$. Nodes labeled 8 and 10 are assigned larger buffer capacities — $[b_8] = 650$, $[b_{10}] = 400$ — to achieve equilibrium with smaller number of packets, compared with the equilibrium that the other three nodes achieve, within $\mathcal{G}_2$. We also observe that node 8 achieves equilibrium with fewer packets (as its buffer capacity is $[b_8] = 650 > [b_{10}] = 400$) compared with the equilibrium–packets for node 10. For the same reason, node 8 is assigned a larger buffer capacity ($[b_8] = 650 > [b_1] = 560$) than node 1, and thus the number of equilibrium–packets for node 8 is less than for node 1. Figure 4.b shows the synchronised solution curves for the nodes that belong to the fifth subgraph, $\mathcal{G}_5$. Node 22 is assigned a relatively large buffer capacity ($[b_{22}] = 450$) and therefore is the one that achieves the equilibrium point with the least number of equilibrium–packets compared with the rest of the nodes $\in \mathcal{G}_5$.

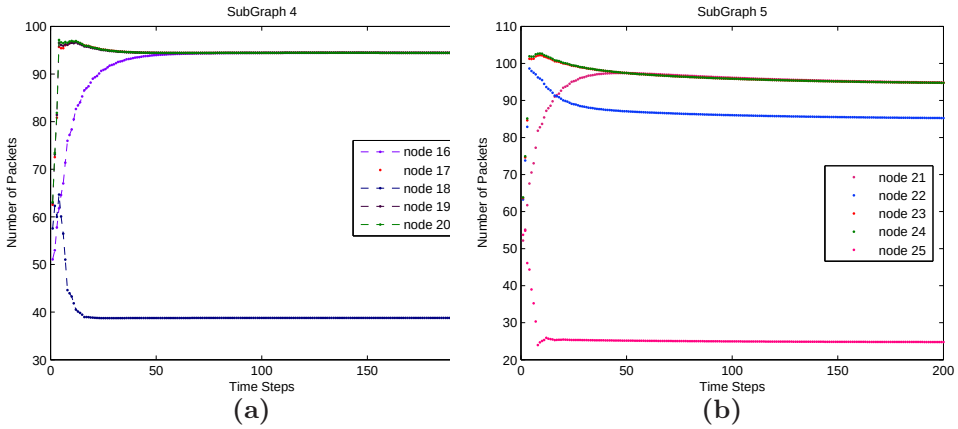

FIGURE 5. Example 3: $\mathcal{G}_4$, $\mathcal{G}_5$: Ravsz-Barabasi graph with non-homogeneously distributed buffer capacities and service powers

Finally, we illustrate results from the third example, according to which all the attributes characterising each node are distributed in a non–homogeneous way. Figure 5 illustrates solution curves produced from nodes that belong to G_4 and G_5. This example shows that regardless of the non–homogeneity of the graph's attributes, the curves are synchronised and lead to analogous equilibrium. Furthermore, nodes assigned a high service power produce overshoot for the first ten time steps, while their high service power affects the number of equilibrium packets. Figure 5.a illustrates the solutions from the nodes that belong to the fourth subnet, $\mathcal{G}_4$. Node 16 has no communication link with node 1 and is the only one that that is not introducing overshoot in its curve. Node 18 is assigned twice the service power, compared with the other nodes within G_4. Therefore, it only forwards packets to its first neighbours. Nodes 17, 19 and 20 have a significantly faster response time with introduced overshoot, as they are directly connected to node 1 and node 18. Therefore, we observe in Figure 5.a that the number of packets within nodes 17, 19, and 20 is dramatically increased, jumping from 50 to 100 packets within the first 10 time steps. They also achieve the same equilibrium point, as they are allocated the same value for both their buffer capacities and service powers. Moreover, we observe that the higher service power affects the equilibrium point. Figure 5.b illustrates curves that describe the number of packets at each time step, for nodes within the fifth subgraph, $\mathcal{G}_5$. Nodes 21, 23 and 24 are allocated the same values for their buffer capacities and service powers, therefore they achieve the same equilibrium point. In contrast, node 22 achieves a lower equilibrium compared with the others. Node 21 is not connected to node 1 and is the only one that is not introducing overshoot. Node 25 is assigned a three times stronger service power, compared with the other nodes within G_5, causing a lower equilibrium point, by means of overshoot.

6. **Conclusions.** Through our model, we find the best possible function of the buffer capacity and service power as a form of the traffic load (number of packets). The function is characterized as the best possible with respect to the synchronizability of solution curves. The coupling element λ_{ij} comprises the óptimalf́eature of our model through which we synchronize our solution curves, taking consideration both the node properties (buffer capacity and service ability) and the network's

connectivity. We inject a specific number of packets and we allocate network resources, such as buffer capacity and service power to initialise our test network. Our model fairly distributes the number of initial packets to each node and very quickly achieves equilibrium. The coupled model DATRAv2 is better than its original version [9], as it achieves equilibrium almost hundred times more quickly, without oscillating around the equilibrium point. We observed the inversely proportional relationship between the buffer capacity and number of equilibrium—packets. Additionally, high service power nodes 'shot out' too many packets to their neighbours, which affected both their equilibrium and nature of the solution curves. Taking the above points into consideration, we conclude that it is vital to properly assign network resources such as buffer capacity and service power, in order to design a stable network (in terms of the solution curves). Our suggested model for assigning buffer capacities and service powers to nodes will be useful in designing practical stable networks, without wasting the resources. During a simulation (numerical analysis of an example), given that the number of initial packets injected into the network is static, our model always reaches an equilibrium solution.

Acknowledgments. The authors would like to thank Alexandre Caboussat (Department of Mathematics, University of Houston) for his valuable comments and suggestions.

REFERENCES

[1] D. Alderson, H. Chang, M. Roughan and S. Uhlig, W. Willinger *The many facets of internet topology and traffic* Netw. Heterog. Media, **1**, (2006), 569–600, http://www.ams.org/mathscinet-getitem?mr=2276254.

[2] C. D'Apice, R. Manzo and B. Piccoli *A fluid dynamic model for telecommunication networks with sources and destinations* SIAM Journal on Applied Mathematics (2008), **68**, 4, 981–1003, http://www.ams.org/mathscinet-getitem?mr=2455475.

[3] D. E. Boschi, C. and Louis, E. and Ortega, G. *Triggering synchronized oscillations through arbitrarily weak diversity in close-to-threshold excitable media* Phys. Rev. E **65** 1 (2001).

[4] Cartwright and Julyan H. E. *Emergent global oscillations in heterogeneous excitable media: The example of pancreatic β cells* Phys. Rev. E **62** 1 (2000), 1149–1154 http://link.aps.org/doi/10.1103/PhysRevE.62.1149.

[5] M. Chavez and D–U. Hwang and A. Amann and H. G. E. Hentschel and S. Boccaletti *Synchronization is Enhanced in Weighted Complex Networks* Phys. Rev. Lett. **94** (2005), 218701–4 http://link.aps.org/doi/10.1103/PhysRevLett.94.218701.

[6] Allen B. Downey *Evidence for long–tailed distributions in the Internet* SIGCOMM conference on Internet measurement, (2001), 149–160.

[7] E. Hairer, F.P. Norsett and G. Wanner *Solving Ordinary Differential Equations* I. Nonstiff Problems. Springer, Berlin-Heidelberg-New York, (1987).

[8] Hong, H. and Choi, M. Y. and Kim, Beom Jun *Synchronization on small-world networks* Phys. Rev. E **65** 2 (2002), 026139-13 http://link.aps.org/doi/10.1103/PhysRevE.65.026139.

[9] Antonia Katzouraki and Philippe De Wilde and Robert A. Ghanea Hercock *Intelligent Traffic Control on Internet-Like Topologies - DATRA - Dynamically Adjusted Traffic Rate Alterations* SOAS IOS Press, Frontiers in Artificial Intelligence and Applications **135** (2005).

[10] F. Kelly *Mathematical modelling of the Internet* ICIAM in Fourth International Congress on Industrial and Applied Mathematics (1999).

[11] F. Kelly, A. K. Maulloo and D. K. H. Tan *Rate control in communication networks: shadow prices, proportional fairness and stability.* J. Operational Research Society **49** (1998).

[12] Lago-Fernández, Luis F. and Huerta, Ramón and Corbacho, Fernando and Sigüenza, Juan A. *Fast Response and Temporal Coherent Oscillations in Small-World Networks* Phys. Rev. Lett. **84** 12 (2000).

[13] A. Marigo *Equilibria for data networks* Netw. Heterog. Media, **2**, (2007), 497–528, http://www.ams.org/mathscinet-getitem?mr=2318843.

[14] Motter and Adilson E. *Cascade Control and Defense in Complex Networks* Phys. Rev. Lett. **93** 9 (2004), 098701–4 http://link.aps.org/doi/10.1103/PhysRevLett.93.098701.

[15] Motter, Adilson E. and Lai, Ying-Cheng *Cascade-based attacks on complex networks* Phys. Rev. E **66**, 6 (2002), 065102–5 http://link.aps.org/doi/10.1103/PhysRevE.66.065102.

[16] G. Restrepo and Edward Ott and Brian R. Hunt *Onset of synchronization in large networks of coupled oscillators* Physical Review E **71** 3 (2005) http://link.aps.org/doi/10.1103/PhysRevE.71.036151.

[17] B. R. Trees and V. Saranathan and D. Stroud *Synchronization in disordered Josephson junction arrays: Small-world connections and the Kuramoto model* Physical Review E **71** 1 (2005) http://link.aps.org/doi/10.1103/PhysRevE.71.016215.

[18] I. Vragovic, E. Louis and A. Diaz–Guilera *Performance of excitable small-world networks of Bonhoeffer-van der Pol-FitzHugh-Nagumo oscillators* IOP Europhys. Lett. **76** (2007).

[19] B. Wang and B. J. Kim *A high-robustness and low-cost model for cascading failures* IOP, EPL **78**,11 (2007).

[20] Western Systems Coordinating Council (WSCC) *Disturbance report for the Power System Outage that occured on Western Interconnection at 1548 PAST* http://www.wscc.com.

[21] Zhao, Liang and Park, Kwangho and Lai, Ying-Cheng *Attack vulnerability of scale-free networks due to cascading breakdown* Phys. Rev. E,**70** 3 (2004).

Received July 2008; revised August 2009.

E-mail address: ak3@imperial.ac.uk
E-mail address: tania.stathaki@imperial.ac.uk