Self-Organizing Model for Virtual Enterprises

Dianxun Shuai[1], Qing Shuai[2], Liangjun Huang[1], and Yuzhe Liu[1]

[1] East China University of Science and Technology,
Shanghai 200237,China,shdx411022@online.sh.cn
[2] Huazhong University of Scienceand Technology,
Wuhan 430074,China,echoshuai@163.com

Abstract. This paper presents a new self-organizing model based on the generalized particle dynamics (GPD) for virtual enterprises.[1] Differing from traditional organizing model for virtual enterprises, the GPD-based self-organizing model may provide the optimal organization for desired virtual enterprises according to historical organizations and current requirements. The GPD conception, algorithm and its properties are discussed. The GPD-based self-organizing modeling has advantages in terms of the real-time performance, adaptability, reliability and the learning ability over traditional organizing models for virtual enterprises

1 Introduction

Distributed virtual enterprises are featured by the geographically distributed resources and jobs, heterogeneous collection of autonomous systems, and collaboration based large-scale problem-solving. Recently, Grid technologies are being integrated with Web Services technologies to provide a framework for interoperable application-to-application interactions for virtual enterprises. The orchestration of virtual enterprises requires the optimal dynamic organization for systematic resources and services.

Most of existing models usually not only regard concurrently occurring service approaches in virtual enterprises to be in dependent of each other, but also regard current service approaches to be independent of past and future service approaches in virtual enterprises. Thus the traditional virtual enterprises take into consideration neither the impact of historical organizations on current organization of virtual enterprises, nor the interference among a great number of concurrently occurring services in virtual enterprises.

This paper proposes a generalized particle dynamics (GPD) for self-organizing virtual enterprises, which is based on a Markov-chain-type proliferation process and an automatic optimal selection process for service approaches in virtual enterprises differing from traditional organizing models, the GPD-based self-organizing model not only considers influences of historical organizations on current virtual enterprises, but also considers interference among concurrent services in virtual enterprises. As a result, the GPD-based self-organizing model may provide the optimal organization for virtual enterprises. The analysis and simulations have revealed many advantages of the self-organizing model in terms of the real-time performance, adaptability, reliability and the learning ability over traditional organizing models for virtual enterprises.
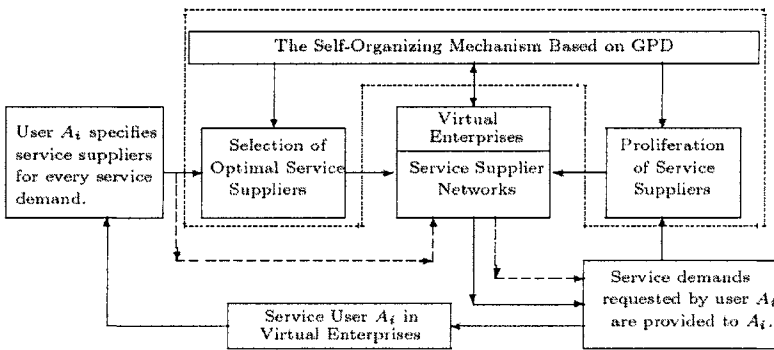

Fig. 1. The difference between the traditional model and the self-organizing model for virtual enterprises. The directed dash-lines represent the service approach in traditional virtual enterprises. The blocks that are framed by dot-lines represent the service approach in the self-organizing virtual enterprises. By the self-organizing virtual enterprises, the optimal service approaches may be realized.

2   Self-Organizing Model for Virtual Enterprises

The essential difference between the traditional model and self-organizing model for virtual enterprises is shown in Fig. 1. By the traditional model, any service user A_i in virtual enterprises should fully know and exactly specify service suppliers that may offer its service demands. On the other hand, however, by self-organizing virtual enterprises, service approaches for service demands of the user A_i are determined optimally by self-organizing service networks, rather than directly by service suppliers indicated by the user A_i. Current self-organizing service networks are established on the basis of the past service approaches through a proliferation process of service suppliers in virtual enterprises.

As shown in Fig. 2(a), for example, by traditional virtual enterprises a user in virtual enterprises exactly specifies 6 distinct service suppliers to offer it 16 kinds o

service demands in a given time session τ. Once the No. 5 service supplier fails to work, 4 kinds of service demands, denoted by d, e, f, g, could not be provided. However, by self-organizing virtual enterprises, a self-organizing service networks as shown in Fig. 2(b) have been built by using service approaches before the time session τ, so that 16 kinds of service demands might be offered from as many as 15 distinct service suppliers, and most of service demands can be provided from more than one service supplier. Even the No. 5 service supplier fails to work; there is no influence on offering the user's demands. The comparison with respect to service robustness for two kinds of virtual enterprises is shown in Fig. 3, which gives the relation between the average number of lost services and the number of faulty service suppliers.
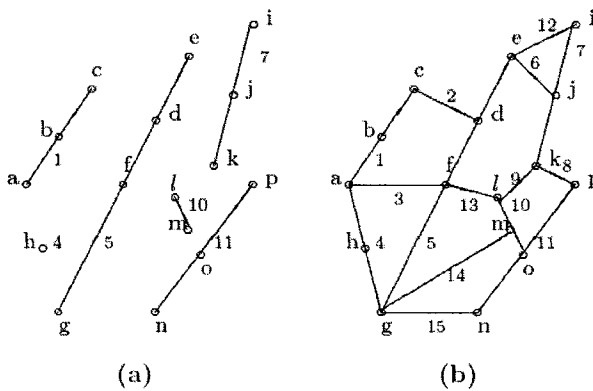

(a)   (b)

Fig. 2. The comparison of service approaches between the traditional model and the self-organizing model for virtual enterprises, where each service supplier is denoted by a solid line labeled in a digit number; each kind of service demands is denoted by a node labeled in an English letter; and if a kind of service demands can be provided by a service supplier, then the corresponding node lies on the corresponding solid line. (a) The service approaches in traditional virtual enterprises, where 6 service suppliers need to be rigidly specified to offer 16 kinds of service demands. (b) The possible service approaches in self-organizing virtual enterprises, where optimal service suppliers to offer 16 kinds of service demands may be selected from as many as 15 service suppliers.

3   Generalized Particle Dynamics

We make use of a generalized particle dynamics (GPD) to implement the proliferation process of service suppliers and service demands, and accordingly to construct powerful self-organizing service networks for self-organizing virtual enterprises. The GPD method is based on stochastic Markov processes over the space of service supplies and the space of service demands, so that current organization of virtual enterprises may be related both to past organizations and to current service demands.

Two two-dimensional $N \times N$ particle arrays, GPA-I and GPA-II, as shown in Fig. 3, are used to perform GPD's for service demands and service suppliers,

respectively. In the time session τ , initially all the kinds of service demands and all the service suppliers are randomly mapped onto particles on the array GPA-I and GPA-II, respectively. Let the data object carried by a service demand particle represent a service supplier that can provide the service demand; and the data object carried by a service supplier particle represent a kind of service demands that can be provided by the service supplier. Arrays, GPA-I and GPA- II, are motion fields for service demand particles and service supplier particles, respectively. The state $s_{ij}(t)$ of the particle c_{ij} that is located at the coordinate point (i,j) on GPA-I (GPA-II, resp.) at a time t in the time session τ is equal to the data object carried by the particle. If there is no particle at the coordinate point (i,j) on GPA-I (GPA-II, resp.) at time t, then denote $s_{ij}(t) = \emptyset$. A special distribution of particles on a GPA array is called a configuration, with all the possible configurations forming a configuration space.

By the same local transitive rule, particles on GPA-I (GPA-II, resp.) randomly and concurrently move back and forth, with a motion probability being determined by a harmony function. We thus obtain a Markov stochastic process over the configuration space on GPA-I (GPA-II, resp.), which can give rise to stationary probability distributions with the maximal entropy over the corresponding configuration space. By maximal stationary probability distributions of service demand particles on GPA-I and service supplier particles on GPA-II, we may easily establish self-organizing service networks to offer optimal service approaches and to organize virtual enterprises in time session τ . For simplicity, in what follows we will discuss particle dynamics only for one of GPA-I and GPA-II.

The similarity $d_{(ij,i'j')}(t)$ between two particles c_{ij} and $c_{i'j'}$ at time t in time session τ is defined by

$$
\begin{cases}
0 \le d_{(ij,\,i'j')}(t) \le 1, & \text{if } s_{ij}(t) \ne \emptyset \text{ and } s_{i'j'}(t) \ne \emptyset; \\
d_{(ij,\,i'j')}(t) = -1, & \text{if } s_{ij}(t) = \emptyset \text{ or } s_{i'j'}(t) = \emptyset.
\end{cases}
\tag{1}
$$

The harmony $h_{(ij,i'j')}(t)$ between two particles c_{ij} and $c_{i'j'}$ at time t is defined by

$$
h_{(ij,\,i'j')}(t) = \begin{cases}
1, & \text{if } d_{(ij,\,i'j')}(t) \ge \theta_{ij}; \\
0, & \text{if } d_{(ij,\,i'j')}(t) = -1; \\
-1, & \text{if } 0 \le d_{(ij,\,i'j')}(t) < \theta_{ij};
\end{cases}
\tag{2}
$$

The harmony $h_{ij}(t)$ of the particle c_{ij} at time t is defined by

$$
h_{ij}(t) = \sum_{(i',\,j') \in N_{ij}} h_{(ij,i'j')}(t).
$$

where θ_{ij} is a previously given threshold, $0 \le \theta_{ij} \le 1$; and N_{ij} is the neighbor of the coordinate (i,j) that is defined by $N_{ij} = \{(i-1,j),(i+1,j),(i,j-1),(i,j+1)\}$.

The matrix $\Gamma(t) = [s_{ij}(t)]\left[s_{ij}(t)\right]_{N \times N}$ represents a configuration, that is, a special distribution of particles on GPA-I or GPA-II at time t. For descriptive convenience, a possible configuration is indicated by an index within a positive integer set $I_{\Box}$. For $j(t) = u\ 2\ I_{\Box}$, the aggregate harmony of $\Gamma(t)$ is defined by

$$\mathcal{H}(\Gamma(t)) = \mathcal{H}(u) = \sum_{i,j} w_{ij}(\tau)\, h_{ij}(t), \tag{3}$$

where $\omega_{ij}(\tau)$ is a weight coefficient which can be obtained by a learning algorithm according to a priori probability distribution of past organizations of virtual enterprises.
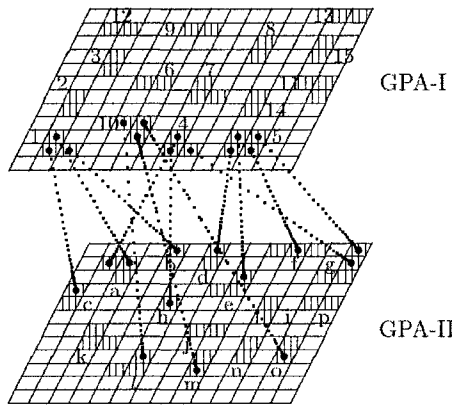

Fig. 3. Particle dynamics on two particle arrays, GPA-I and GPA-II, for self- organizing virtual enterprises. The array GPA-I is the motion field for service demand particles illustrated by shadowed squares, with a particle carrying a data object that represents a service supplier in Fig. 2(b). The array GPA-II is the motion field for service supplier particles illustrated by shadowed squares, with a particle carrying a data object that represents kind of service demands in Fig. 2(b).

If the particle c_{ij} randomly chooses the coordinate (i', j') as its motion destination at time $(t + 1)$, then the probability with which the particle c_{ij} moves to the position (i', j') at time $(t + 1)$, that is, the data object being carried by c_{ij} is successfully transited to the coordinate (i', j') at time $t+1$, is determined by

$$p = f(\triangle \mathcal{H}) = \frac{1}{(1 + e^{\triangle \mathcal{H}/T})}, \tag{4}$$

where $\Delta H = H\left(\Gamma(t)\right) - H\left(\Gamma(t+1)\right)$ is the aggregate harmony increment if the chosen transition happens. The thermodynamic temperature parameter T is used to speed up the transient to a stationary probability distribution, and to avoid stagnating into local minima.

4   Particle Dynamics Algorithms and Their Properties

For GPA-I or GPA-II, we use two exactly same duality lattices, Lattice A and Lattice B, to prepare two randomly generated configurations, u and v.

Parallel Particle Dynamics Algorithm GPDA :

Costep 1. The states of all the coordinates in Lattice A and Lattice B are set to ϕ, which indicates that no particle lies on the GPA array.

Costep 2. At the initial time t_0, all the particles are concurrently and randomly mapped on Lattice A and Lattice B, with each particle carrying a data object that represents a service supplier for particles on GPA-I and a kind of service demands for particles on GPA-II.

Costep 3. At the current time t, every particle c_{ij} on Lattice A and Lattice B concurrently compute its harmony $h_{ij}(t)$ by Eqs. (1) and (2).

Costep 4. At the current time t, Lattice A and Lattice B concurrently compute their aggregate harmonies by Eq. (3).

Costep 5. At the current time t, compute the transitive probability by Eq. (4), and then, using the transitive probability, randomly choose one from the two configurations on Lattice A and Lattice B as a candidate configuration for the next time $t+1$ processing.

Costep 6. The configuration not chosen in Costep 5 is updated by a new randomly generated configuration for the next time $t+1$ processing.

Costep 7. Repeat Costep 3 through Costep 6 until a stationary probability distribution over configuration space is obtained, where that with the maximal probability corresponds to the optimal service organization for virtual enterprises in the time session τ .

Given a priori probability distribution q_u with respect to the configuration u, the weight coefficients of Eq. (3) can be obtained by using the parallel learning algorithm GPD-WLA.

Parallel Learning Algorithm GPD-WLA:

Costep 1. In parallel, initiate the value $\omega_k(t_0), k \in \{1, \cdots, n\}$, where k is the number of service supplier.

Costep 2. In parallel, compute the local harmonic function $h_{i(u,k),j(u,k)}$ for the configuration u and the k-th service supplier.

Costep 3. Using $h_{i(u,k),j(u,k)}$ and the priori probability q_u, in parallel compute the average value $\overline{h}_k, k \in \{1, \cdots, n\}$.

Costep 4. Using $w_k(t)$ and the algorithm GPDA, obtain a stationary probability distribution p_u; and then using $h_{i(u;k);j(u;k)}$ compute the average value $\overset{\square}{h}_k$ under the probability distribution p_u; for $k \in \{1, \cdots, n\}$.

Costep 5. Let $\Delta\omega_k(t) = \overline{h_k} - \overline{h_k}$. In parallel, update the weights w_k, such that $\omega_k(t+1) = \omega_k(t) + \Delta\omega_k(t)$.

Costep 6. Once $\Delta\omega_k(t) \approx 0$ holds true, then $w_k(t)$ is just the required weight coefficient; Otherwise, go to Costep 4.
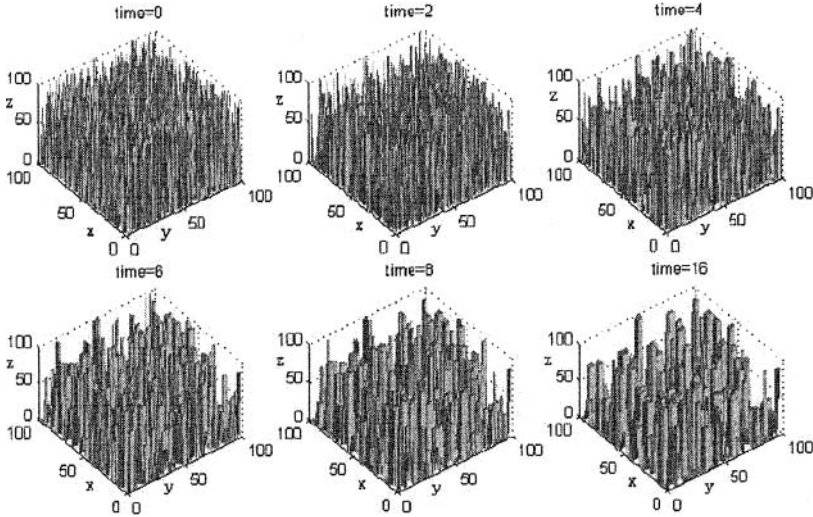

Fig. 4. The state evolution of particles on GPA-I during executing the parallel algorithm GPDA. The number of kinds of service demands: 100; The number of service suppliers: 56; The number of service requests: 4000; The number of service suppliers specified by service users: 5; The number of particles on GPA-I: 10000.

We derive properties of GPDA and GPD-WLA as follows, proofs omitted for page limitation.

Lemma 1. *The stochastic process* $\{\Gamma(t), t = 0,1,\cdots\}$ *generated by the algorithm GPDA is a finite homogeneous Markov chain.*

Lemma 2. *The stochastic process* $\{\Gamma(t), t = 0,1,\cdots\}$ *generated by the algorithm GPDA is an irreducible homogeneous Markov chain, where all the configurations are positive recurrent, that is, any configuration may return to itself in a finite time period with the probability 1.*

Lemma 3. *The stochastic process* $\{\Gamma(t), t = 0,1,\cdots\}$ *generated by the algorithm GPDA must reach a stationary probability distribution over the configuration space, that is, any configuration may occur with a fixed probability. Moreover the stationary distribution is independent of the initial configuration.*
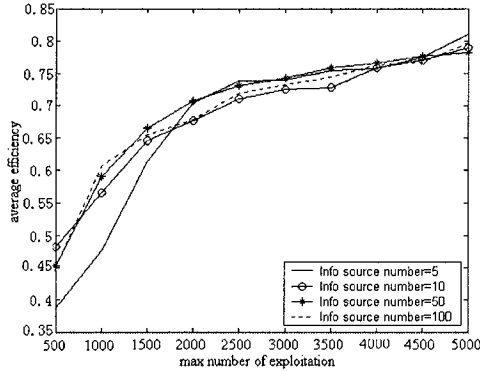
Fig. 5. The efficiency of the GPD-based self-organizing virtual enterprises increases almost monotonically with the number of service requests. The number of particles: 10000; The number of kinds of service demands: 100.

Theorem 1. *The stationary probability distribution obtained by executing the algorithm GPDA is the maximal entropy distribution over the configuration space. Moreover the stochastic configuration with the maximal probability has the maximal aggregate harmony.*

Theorem 2. *A configuration with the maximal probability in a stationary probability distribution produced by the algorithm GPDA has the minimal number of connected regions on GPA-I and GPA-II, with each region having the same service supplier for GPA-I and same kind of service demands for GPA-II.*

Theorem 3. *The parallel learning algorithm GPD-WLA can converge and gives rise to* $\lim_{t \to \infty} \omega_k (t)$ *that is equal to the Lagrange multipliers for maximizing the aggregate harmony subject to the priori probability distribution* q_u .

5 Simulations

Through a large number of simulations, we can see that the GPD-based self-organizing virtual enterprises not only greatly reduce the aggregate cost of service approaches, but also greatly increase the service reliability. The state evolution of particles during executing the parallel algorithm GPDA is shown in Fig. 5. The efficiency of GPD-based self-organizing virtual enterprises is illustrated in Fig. 6. Comparisons between different strategies for service approaches in virtual enterprises are shown in Fig. 7

6 Conclusions

We draw conclusions as follows:
   • In comparison with traditional virtual enterprises, proposed GPD-based self-organizing virtual enterprises not only greatly reduce the average overhead for service approaches, but also significantly improve their real-time performance,
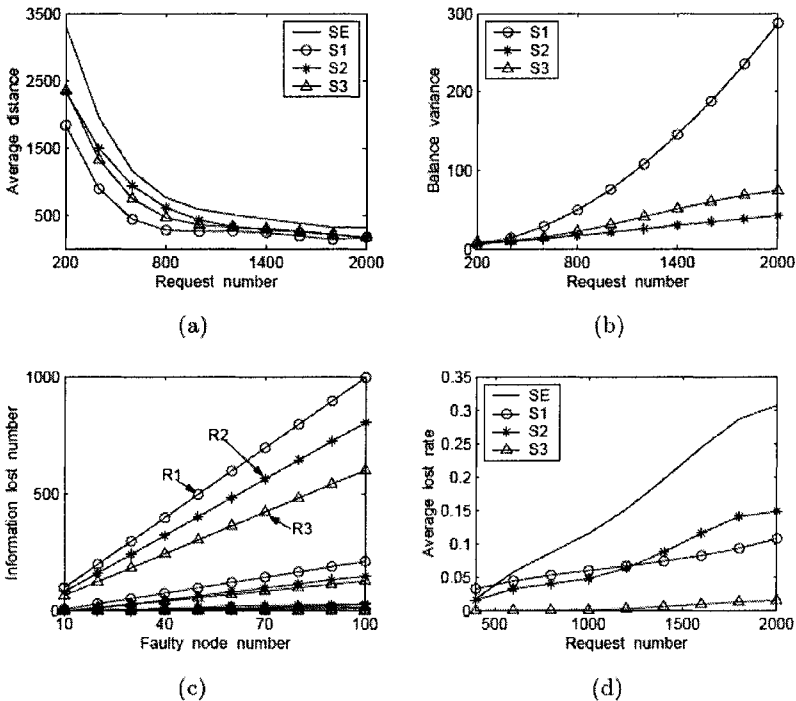
adaptability, and reliability.

(a)

(b)

(c)

(d)

Fig. 6. Performance comparisons among different strategies of service approaches in virtual enterprises, where SE is for the traditional virtual enterprises; S1, S2 and S3 are for the GPD-based self-organizing virtual enterprises; S1 is for the Closest Selection Strategy; S2 is for the Maximal-Load Selection Strategy; S3 is for the First-Fitting Selection Strategy. R1-R3 corresponds to the number of service requests: 1000,800,600, respectively. The number of particles: 2500; The number of kinds of service demands: 50; The number of service suppliers specified by service users: 5.

● Since all the particles evolve independently and simultaneously by the same local dynamic rule, the GPD-based self-organizing virtual enterprises have higher parallelism, Moreover GPD is based on a stochastic process, it is insensitive to many emergent and random events, such as the perturbation, noise, failure and congestion in virtual enterprises.

● The model, architecture and algorithms of GPD are all independent of the problem scale under consideration, so that it is suitable for VLSI hardware implementation.

● The GPD-based self-organizing virtual enterprises can be implemented for different granularity of services. It also has advantages in terms of the ability to learn different service environment, such as different probability distribution over the service demands space and service supplier's space.

References

1. S. Chandramathi and S. Shanmugavel, Fuzzy-Based Dynamic Bandwidth Allocation for Heterogeneous Sources in ATM Network, *Applied Soft Computing Journal* **3**(1), 53-70 (2003).

2. K. Krauter, R. Buyya, and M. Maheswaran, A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing, *Software Practice and Experience* **2**, 135-164 (2002).

3. L. Massonlie and J. Roberts, Bandwidth Sharing: Objectives and Algorithms, *IEEE/ACM Transactions on Networking* **10**(3), 320-328 (2002).

4. Y.F. Huang and B.W. Chao, Apriority-Based Resource Allocation Strategy in Distributed Computing Networks, *The Journal of Systems and Software* **58**, 221- 233 (2001).

5. F.P. Kelly, Mathematical Modeling of the Internet, in B. Engquist and W. Schmid (ed.). Mathematics Unlimited - 2001 and Beyond, (Springer-Verlag, Berlin, 2001), pp. 685-702.

6. I. Foster, G. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organization, *Journal of High-Performance Computing Applications* **15**(3), 200-222 (2001).

7. B. Hamidzadeh, L.Y. Kit, and D.J. Lilja, Dynamic Task Scheduling Using Online Optimization, *Journal of Parallel and Distributed Systems* **11**, 1151-1163 (2000).

8. H.Y. Thomas, B. Li, S.P. Shivendra, and T. Henry, On Network Bandwidth Allocation Policies and Feedback Control Algorithms for Packet Networks, *Computer Networks* **34**, 481-501 (2000).

9. I. Foster and G. Kesselman, *Grid: Blueprint for a New Computing Infrastructure* (Morgan Kaufmann Publisher, 1999).

10. O. Shehory and S. Kraus, Methods for Task Allocation via Agent Coalition Formation, *Artificial Intelligence* **101**, 165-200 (1998).

11. M. Maheswaran, S. Ali, H. Siegen, D. Hensgen, and R. Freund, Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems, *Journal of Parallel and Distributed Computing* **59**, 107-131 (1999).

12. O. Shehory, S. Kraus, and O. Yadgar, Emergent Cooperative Goal-Satisfaction in Large-Scale Automated-Agent Systems, *Artificial Intelligence* **110**, 1-55 (1999).

13. F.P. Kelly, Charging and Rate Control for Elastic Traffic, *European Transactions on Telecommunications* **29**, 1009-1016 (1997).

14. F.P. Kelly, A. Maulloo, and D. Tan, Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability, *Journal of the Operational Research Society* **49**, 237-252 (1998).