

# Local and Global Models for Large-Scale Peer-to-Peer Systems

Dmitry G. Korzun

**Abstract**—This article considers selected mathematical models on the spectrum between local (microscopic) and global (macroscopic) extreme views. The classification is oriented to the large-scale property, which may not be ignored in designs of P2P systems for the today's and future Internet. A local model exploits the knowledge directly available at a given node, leading to easy implementation. A global model needs the knowledge about the entire system, leading to preservation of certain system invariants. The discussion introduces a rich family of analytical models that fall into this classification and can be constructed using rather simple mathematical techniques.

**Index Terms**—Analytical models, Cooperation, Internet, Network analysis, Peer-to-peer computing, Performance, Routing

## I. INTRODUCTION

Internet applications utilize the peer-to-peer (P2P) approach, which supports effective resource sharing in the open and dynamic distributed environment. The approach does not require centralized control and allows building large-scale systems as self-organizing overlay networks. It becomes increasingly popular in the today's Internet and is crucial for future, especially in such prominent paradigms as ubiquitous computing, Internet of Things, and Big Data. The main P2P features include decentralization and governance autonomy, resilience to massive changes and self-organization, load-balancing and fault-tolerance, scalability and low overhead. Background on the P2P approach, its benefits, system designs, and applications can be found in books [1]–[4] and surveys [5]–[9].

A P2P system can be broadly defined as a distributed system with no centralized infrastructure and where autonomous participants (called peers or nodes) have symmetric roles of a server, client, and router. Basic modeling and algorithmic methods are inherited from earlier studied areas, e.g., processor/communication interconnection networks [10], parallel computation in shared memory [11], and computer networks [12]. Nodes and their resources have unique identifiers (IDs). Distributed Hash Table (DHT) is a scalable mechanism for resolving lookups of resource IDs to responsible nodes [13]–[15]. Nodes self-organize by maintaining resources and control data locally and by performing control

operations for the cooperation. Examples include CAN [16], Symphony [17], Chord [18], and Pastry [19].

The Internet provides a ubiquitous connectivity substrate where nodes form an overlay network using the Internet Protocol (IP) for direct node-to-node communication. Each node keeps contacts (node IDs and IP addresses in its routing table) to a few other nodes (neighbors). A node acts as 1) a client requesting resources it needs from a neighbor—consumption, 2) a server for incoming requests—contribution, and 3) a router forwarding requests to a neighbor when the resource is not available locally—both consumption and contribution.

The Internet-oriented case opens the possibility to construct really large-scale P2P systems. The large-scale property meets, however, nontrivial challenges. The key ones are heterogeneity in participant population and shared resources, trust in distributed cooperation, and limitation of local knowledge on the entire system. In certain respects namely these challenges make the P2P area a distinct field for modeling, having with own principles and methods.

The heterogeneity is inevitable property of modern networks, where assumption on homogeneity is mostly violated. Qualitatively equal nodes of a P2P system must be quantitatively different in their responsibility since they have different capacities. The Internet exhibits non-flat structure with hierarchical domain organization. The resource distribution is non-uniform due to data semantics.

Each node is controlled by an independent entity. Nodes have to self-organize in an open distributed environment. Due to autonomy, individual actions are voluntarily chosen and determined by independent and rational behavior. Lack of cooperation eventually degrades any P2P system if there is no mechanism for establishing trust between nodes.

The assumption on global knowledge is impractical for large-scale systems since a node can track no global and precise view to the dynamic network topology and characteristics of all participants. Having only local knowledge of other nodes, each node makes own decisions based on its subjective view. Cooperation activity expands or modifies node's subjective view, providing feedback to the decision making process.

Solutions to the challenges lead to a rich source for elegant mathematical models. They are applied to construct solutions for use in internal algorithms and communication protocols. Motivated by the large-scale property, this article considers a classification of the mathematical models into two extreme classes—local and global. A local model exploits only knowledge directly available at a given node. A global model needs knowledge about the entire system.

Manuscript received November 25, 2013; revised December 3, 2013; accepted December 13, 2013. Date of online publication: December 26, 2013.

This work was supported by Academy of Finland project SEMOHealth.

The article was published with financial support from Strategic Development Program of Petrozavodsk State University.

D. G. Korzun is with the Department of Computer Science of Petrozavodsk State University PetrSU, Russia (e-mail: dkorzun@cs.karelia.ru) and with the Helsinki Institute for Information Technology HIIT and Department of Computer Science and Engineering CSE, Aalto University, Finland (e-mail: dkorzun@hiit.fi).

For each class a set of selected models is discussed, demonstrating the basic modeling principles. The focus is on such well-known P2P application problems as neighbor maintenance and network topology control, next-hop forwarding and lookup query routing, rational and fair cooperation. Solutions based on a global model are limited to some high-level characteristics of the system. Solutions based on a local model suffer from the lack of relevant information to be used in control actions.

On the other hand, a local model can be augmented with additional knowledge and a global model can dismiss a part of knowledge. Therefore, a wide spectrum of classes in between local and global models exists. For practical use, such models should be provided with a mechanism to tune the tradeoff between local and global knowledge. Let us leave this topic for further work.

The models selection reflects the large-scale property as well as author's experience in this area [4]. The article is intentionally limits the selection with rather simple mathematical models. An interested reader can find proofs and further details in the provided references.

The rest of the article is organized as follows. Section II summarizes P2P application problems used for demonstrating the modeling aspects. Section III introduces the models classification and basic principles of modeling for P2P application problems. Sections IV and V review selected local and global models, respectively. Section VI summarizes the paper.

## II. APPLICATION PROBLEMS

A P2P system can be viewed on the microscopic level (local, node-respective) and on the macroscopic level (global, system-wide). This section summarizes some P2P application problems relating this “local-global” point of view. Table I lists examples of the corresponding pairs of application problems.

TABLE I  
P2P APPLICATION PROBLEMS: LOCAL AND GLOBAL INSTANCES

Local	Global
<i>Neighbor maintenance</i> : a node constructs and updates its local routing table	<i>Network topology control</i> : the system keeps all nodes properly interconnected
<i>Next hop selection</i> : a node finds the best neighbors to forward a given search query	<i>Routing</i> : the system constructs an efficient path for any search query
<i>Rational cooperation</i> : a node balances own consumption from and provision to the system	<i>Fair and generous cooperation</i> : the system maximizes the social welfare of all participating nodes

### A. Overlay Network Topology

Consider a P2P overlay of  $N$  nodes. Table II summarizes the basic notation (based on [4]). Let node identifiers (IDs) be assigned from a space  $S$  with a distance metric  $\rho$ . Examples of space distance metric are the Euclidean distance adopted in CAN [16], the length of the common prefix [20] adopted in Pastry [19] and Tapestry [21], and the clockwise distance on the integer ring  $[0, 2^m) \bmod 2^m$  adopted in Chord [18].

On the microscopic view, each node  $u$  maintains a local routing table  $T_u$  of entries  $(v, \text{IP}_v)$ , where  $v$  is a neighbor

TABLE II  
SYMBOL NOTATION

Notation	Description
$S$	Node ID space (scalar or vector). Let $u$ , $v$ , and $w$ stand for P2P nodes and their IDs.
$N$	The number of alive nodes in overlay. In some contexts, $N$ denotes the full set of alive nodes.
$u, v, w$	P2P nodes by their IDs.
$T_u, N_u$	Routing table of $u \in N$ . Although $T_u$ consists of pairs $(v, \text{IP}_v)$ let us write $v \in T_u$ if no confusion. Let $N_u$ be the set of $u$ 's neighbors if IP addresses are not important.
$u \rightarrow^+ v$	Multi-hop overlay path $u \rightarrow w_1 \rightarrow w_2 \rightarrow \dots \rightarrow w_{l-1} \rightarrow v$ . The number of hops is the path length $ u \rightarrow^+ v $ .
$\rho(u, v)$	Distance metric in $S$ that satisfies (i) $\rho(u, v) > 0 \forall u, v \in S, u \neq v$ , (ii) $\rho(u, u) = 0 \forall u \in S$ . The symmetry property and the triangle inequality are optional.
$\tau(u, v)$	Distance metric in network, e.g., the number of overlay hops or the sum latency of $u \rightarrow^+ v$ .
$D$	Resource key space (application-specific).
$R$	The number of resources (data items or data keys) stored (and available) in the overlay. In some contexts, $R$ also denotes the full set of currently available resources.
$R_u$	Resources that $u$ knows; $R_u = R_u^+ \cup R_u^-$ , $ R_u $ , where $R_u^+ = \bigcup_{v \in N_u} R_v^+$ and $R_u^- = \bigcup_{v \in N_u} R_v^-$ are the sets of external-like and local-like resources, respectively.
$a_{vk}, b_{vk}$	Metrics of $u$ 's consumption and provision of $k \in R_u$ through $v \in N_u$ .
$d_{vk}$	Surplus counters: $d_{vk} = a_{vk} - b_{vk}$ , $d_v = \sum_{k \in R_u} d_{vk}$ , $d_k = \sum_{v \in N_u} d_{vk}$ .
$\alpha_{vk}, \beta_{vk}$	Clear $u$ 's consumption and provision of resource $k \in R_v$ through $v \in N_u$ ; $d_{vk} = \alpha_{vk} - \beta_{vk}$ such that $\min\{\alpha_{vk}, \beta_{vk}\} = 0$ .
$s_k$	Local resource rank of $k \in R_u$ , both for provision and consumption.
$r_v^-, r_v^+$	Local node ranks of neighbors $v \in N_u$ , for provision and consumption, respectively.

and  $\text{IP}_v$  is its IP address. The number of neighbors  $|T_u|$  is the node degree. If  $|T_u| \ll N$  then  $u$  has local knowledge only about the overlay network. On the macroscopic view, the nodes form a network. Its topology governs how the nodes are interconnected [6].

Overlay network topology is modeled as a graph, embedded into  $S$ . The graph is directional:  $v \in T_u$  does not necessarily lead to  $u \in T_v$ . It means that  $u$  processes incoming requests from some  $w \in N$  for  $w \notin T_u$ , and the node in-degree  $|\{w \mid u \in T_w, w \in N\}|$  must be used for characterizing the load of  $u$ , in addition to the node out-degree  $|T_u|$ .

Structured P2P overlay networks aim at efficient searching [8] to guarantee location of a target resource within a moderate number of hops even for large  $N$ . The location of resources depends essentially on the network topology. The set of nodes receiving a particular query becomes related to the content of the query. A node arranges its neighbors and selects the best ones to resolve search queries. Network topology is tightly controlled via careful maintenance of neighbors [14], and the resultant distributed maintenance repairs the connectivity among nodes in the face of network dynamics.

### B. Routing

A P2P system takes care of distributed assigning and locating resources. A mapping from  $D$  to  $N$  associates resources with participating nodes. Structured P2P networks use the DHT lookup service [13]. It can be thought as a distributed

“indexing” mechanism to resolve search queries. DHT takes  $(k, \text{data})$  value as input ( $k \in D$ ), hashes  $k$  to establish a mapping from keys to nodes (the lookup service), then assigns the data with the node identified.

Consider a typical (flat) DHT. The resource key space  $D$  and node ID space  $S$  coincide. Each resource has a unique key  $k \in S$ ; DHT implements hashing the key to the node ID. For example, the greedy strategy makes the node  $d$  with the closest ID responsible for a key  $k$ ,

$$d = \arg \min_{u \in N} \rho(u, k). \quad (1)$$

At any instant,  $d$  is responsible for keys

$$S(d) = \{k \in S \mid \rho(d, k) < \rho(u, k) \quad \forall u \in N, u \neq d\}.$$

Let a node  $u$  initiate a lookup for a given key  $k$ . Since  $T_u$  does not contain all nodes, the lookup performs multi-hop overlay routing from  $u$  to  $d$ , where  $d$  is unknown beforehand. Each intermediate node  $w$  finds the next-hop node  $v \in T_u$  to forward the lookup, thus forming the one-hop path  $w \rightarrow v$  in the overlay network. Eventually, an  $l$ -hop path  $u \rightarrow^+ d$  is constructed for  $l = |u \rightarrow^+ d| \geq 1$ :

$$u \rightarrow w_1 \rightarrow w_2 \rightarrow \dots \rightarrow w_{l-1} \rightarrow d. \quad (2)$$

For the convenience, denote  $w_0 = u$  and  $w_l = d$ . This multi-hop routing process is depicted in Fig. 1.

DHT routing exploits the properties of metric space  $(S, \rho)$ . In progressive routing, lookup path (2) consists of nodes which are closer to  $d$ , i.e.,  $\rho(w_i, d) < \rho(w_{i-1}, d)$ . Since  $d$  is unknown beforehand, progressive routing applies the key-based criterion

$$\rho(w_i, k) < \rho(w_{i-1}, k), \quad i = 1, 2, \dots, l-1. \quad (3)$$

Greedy routing is the extreme instance of progressive routing when each next-hop node  $w_i$  is the neighbor closest to  $k$ ,

$$w_i = \arg \min_{v \in T_{w_{i-1}}} \rho(v, k), \quad i = 1, 2, \dots, l-1. \quad (4)$$

Note that (4) is a local approximation of (1): the minimization is performed within  $T_{w_{i-1}} \subset N$ .

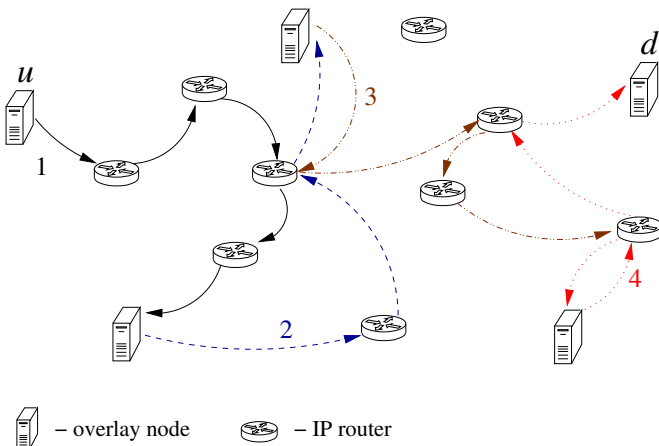


Fig. 1. Multi-hop routing: one hop in the P2P overlay leads to several hops in the underlying IP network

### C. Neighbor Maintenance

DHT lookup service is a base for self-organization. When a new node  $u$  joins, it contacts any existing node to lookup for  $u$  (the node ID acts as a key). According to (1), the lookup returns the closest neighbor  $v$ , identifying the location of  $u$  in the overlay. Then  $u$  communicates with  $v$  to initialize  $T_u$  and  $u$ 's resource storage as well as to disseminate the topology update information. When a node  $u$  leaves, it notifies its neighbors and passes its resources to new responsible nodes. In open and dynamic environments, nodes generally leave the system far less gracefully due to network failures, selfish or malicious reasons, and temporal leaves.

Locally, every node has to perform neighbor maintenance. In the proactive strategy, a node regularly checks its neighbors, e.g., by pinging them periodically. In the reactive strategy, existing overlay communications are used to detect update events, e.g., a neighbor has not responded for lookup by predefined time. When a node  $u$  detects that its neighbor  $v$  has leaved,  $u$  performs the actions similar as if  $v$  notified about its voluntary leave. This local activity implements distributed topology control, making the nodes correctly interconnected.

A fundamental tradeoff problem is routing table size vs. network diameter [14], [15]. Clearly, the larger  $|T_u|$  the shorter  $|u \rightarrow^+ d|$ . Many popular in practice DHTs provide  $|T_u| = O(\log N)$  and  $|u \rightarrow^+ d| = O(\log N)$ . Better tradeoffs are possible, e.g.,  $O(1)$  state and  $O(\log N)$  routing or  $O(\sqrt{N})$  state and  $O(1)$  routing. More complicated network topologies and larger routing tables, however, lead to higher maintenance costs.

Selecting the best tradeoff becomes a challenging problem in case of high dynamics and heterogeneity. The former is one of the P2P characteristic property, which distinguishes P2P systems from many other decentralized systems. It is also known as churn—the system experiences rapid membership changes. The heterogeneity is an inevitable property of such networking environments as today's and future Internet.

The routing distance  $\tau(u, d)$  is one of the key performance criteria for the tradeoff. For example,  $\tau(u, d) = |u \rightarrow^+ d|$  is the number of hops in the overlay network. More accurate metrics take into account the latency in the underlying network, e.g.,  $\tau(u, d) = |u \rightarrow^+ d|_\tau = \sum_{i=0}^l \tau(w_i, w_{i+1})$  is the total lookup latency for (2).

### D. Cooperation

Each node provides and consumes resources to and from other nodes, forming node's costs and gains. In open autonomous systems, nodes are intrinsically selfish. They attempt to maximize own utility, lowering the overall system welfare. Individual consumption tends to come without proper contribution (free-riding). Anonymous nodes are less accountable for their activity and change their IDs with near zero cost (whitewashing). Due to dynamics, many interactions among nodes are one-time; nodes have no idea about other nodes' behavior history, except their current behavior, so very short-term decision-making is needed.

Cooperation among nodes is supported with incentives [22]. A system-wide incentive mechanism encourages every node to

TABLE III  
DECISION-MAKING AT A NODE  $u$  IN MULTI-RESOURCE CONSUMPTION AND PROVISION

Consumption: $u$ is a client, $v \in N_u$ are servers	Provision: $u$ is a server, $v \in N_u$ are clients
Selection of $(k, v)$ for $k \in R_u^+$ and $v \in N_u$ is a consumption decision problem: which resource and from which neighbor to request.	Selection of $(v, k)$ for $v \in N_u$ and $k \in R_u^-$ is a provision decision problem: to which neighbor which resource to provide.
Resource ranks $s = (s_k)_{k \in R_u}$	
Node ranks $r^+ = (r_v^+)_{v \in N_u}$	Node ranks $r^- = (r_v^-)_{v \in N_u}$
1. Allocate consumption capacity among resources $k \in R_u^+$ proportionally to $s_k$ . 2. For given $k$ , allocate its capacity share among neighbors $v \in N_u$ proportionally to $r_v^+$ .	1. Allocate provision capacity among neighbors $v \in N_u$ proportionally to $r_v^-$ . 2. For given $v$ , allocate its capacity share among resources $k \in R_u^+$ proportionally to $s_k$ .

balance rationally in its resource contribution and consumption. In reciprocity-based schemes [23], nodes maintain histories of behavior of other nodes, analyze trends, and apply this information in local decision making. When a node  $v$  become low-contributing then other nodes detect this behavior and reduce their cooperation with  $v$  such that links to  $v$  eventually disappear. That is, if  $v$  is still interested in participation in the system then  $v$  is incited to increase its contribution, i.e., supporting  $v$  to be rational in the cooperation.

Recall that all communications are through neighbors. Bilateral exchange (or direct reciprocity) naturally happens between nodes interested in the same content at the same time. The idea is “I help you and you help me”. Collecting past observations can help analyzing stable good neighbors for longer-term time periods. Any node  $u$  knows at time  $t$  two local metrics,  $a_{vk} = a_{vk}(t)$  and  $b_{vk} = b_{vk}(t)$ , which are  $u$ ’s consumption and provision of resource  $k$  through neighbor  $v$ , respectively. The metrics aggregate the local history of  $u$ ’s direct observations.

Bilateral exchange  $u \leftrightarrow v$  is not always possible, since for two arbitrary nodes  $u$  and  $v$  the former may have no resources that the latter needs. Indirect reciprocity can provide useful extension of the local knowledge. The principle is “I help you and someone helps me”. Exchange cycles [23], [24] like  $u \rightarrow w \rightarrow v \rightarrow u$  can be formed, when  $u$  has resources interested for  $w$ ,  $w$  has resources interested for  $v$ , and finally  $v$  has resources interested for  $u$ . Such cycles are the base for multilateral exchange in P2P systems [25].

A reputation system is often employed to detect malicious nodes or reward well-behaving ones [26], [27]. It provides a distributed mechanism to assign each node with a global reputation value according to its past behavior. Differentiated services are then provided to nodes. A reputation value is an aggregation of opinions from all nodes, i.e., a reputation system is a global mechanism to support indirect reciprocity. The basic aim is at maximizing social welfare—a global measure of economical efficiency of the cooperation.

### E. Resource Exchange

Let  $R$  denote the index set of all system resources. For simplicity assume that each resource is of unlimited supply, e.g., video streaming. A node  $u$  consumes external resources  $k \in R_u^+$  and provides local resources  $k \in R_u^-$  (let  $R_u = R_u^+ \cup R_u^-$ ). Consumption and provision are through neighbors  $v \in N_u$ . Discrete time periods  $\{1, 2, \dots\}$  are indexed by  $t$  and

possibly infinite. Each node  $u$  decides its activity for period  $t + 1$  based on the local knowledge ( $a_{vk}$  and  $b_{vk}$ ) collected up to period  $t$  (including). Further omit  $t$  in the notation when the context of repeated transitions  $t \rightarrow t + 1$  is clear.

BitTorrent [28] is an example of a single-resource P2P system, where the shared resource is bandwidth. Neighbors  $v$  consume from  $u$  in period  $t + 1$  according with their provision to  $u$  in  $t$ . This is an instance of bilateral exchange, and  $u$  makes  $b_v(t + 1)$  close to  $a_v(t)$ . Table III shows the general case when a node  $u$  performs two parallel iterative processes in  $t + 1$ . In provision,  $u$  iteratively selects  $(v, k)$  to provide  $v$  a portion of  $k$ , so having full control of  $b_{vk}(t + 1)$  for  $v \in N_u$ ,  $k \in R_u^-$ . In consumption,  $u$  observes  $a_{vk}(t + 1)$  for  $v \in N_u$ ,  $k \in R_u^+$  since it depends on  $v$ ’s provision strategy. Nevertheless,  $u$  can reduce  $a_{vk}$  by requesting less from  $v$  or  $u$  can stimulate increasing  $a_{vk}$  by more requests to  $v$ . Accordingly,  $u$  iteratively selects  $(k, v)$  to make a request for  $k$  to  $v$ .

These two local processes result in sharing  $u$ ’s consumption and provision capacity between neighbors and resources. The bottom part of Table III shows two-phase rank-based sharing. Resource ranks  $s_k$  and node ranks  $r_v^+$  and  $r_v^-$  are non-negative real numbers that quantify importance of  $k \in R_u$  and  $v \in N_u$  to  $u$  (local ranks or scores). The iterative rank-proportional allocation can be implemented in a round-robin fashion, typically with normalized ranks

$$\sum_{k \in R_u} s_k = 1, \quad \sum_{v \in N_u} r_v^+ = 1, \quad \sum_{v \in N_u} r_v^- = 1. \quad (5)$$

Since  $u$  in consumption is primarily interested in resources, not from whom to consume them,  $u$  decides which resource to request and only then from which neighbor. Similarly, in provision,  $u$  serves neighbors, allocating its capacity for them, and only then  $u$  decides which resources to provide to a selected neighbor. A node  $u$  computes separate local ranks for nodes and resources. The ranks are used in the two parallel iterative processes of provision and consumption. Provision of  $k$  from  $u$  to  $v$  is proportional to  $r_v^- s_k$ . Consumption of  $k$  from  $v$  to  $u$  is proportional to  $s_k r_v^+$ .

### III. APPROACH FOR MODELING

A diversity of mathematical models is known for the P2P application problems from Section II. Given a model, a solution can be constructed applicable either on global (system-wide) or local (node-respective) level. The solution then is implemented on control elements of the system, e.g., on individual nodes or using a distributed algorithm. Although

many mathematical techniques have been already employed in P2P system designs and communication protocols, the common goal is to provide means for control elements to make decisions based on arranging available alternatives.

#### A. Local and Global Extremes in Modeling

From the point of solution implementation, two extreme classes of models can be considered: local and global. Local models are oriented to operation on an individual node, using locally observable information. Global models need input from many nodes and can provide results to many nodes.

Potentially, a global model can describe all required characteristics of a given system if all input data and relations are precisely specified. Applicability of such models is limited due to the practical implementation reasons. There are three ways to implement a global model. Its computations can be delegated to a node, to a centralized subsystem, or to all nodes via a distributed algorithm.

*Node-hosted implementation:* The capacity of a single node is limited, and the computations can be resource-expensive in a large system. The assumption on availability of global knowledge on a single node is impractical due to large-scale distribution and high dynamics of the system.

*Centralized subsystem:* Data collection and resource expensive computation can be implemented in a centralized manner. This way, however, contradicts with the decentralized nature of P2P systems, makes the centralized subsystem a single point of failure, and introduces a third-party element of trust. In the large-scale case, even a powerful subsystem cannot instantly recognize the up-to-date global state of the system, since the communication latency has significant effect.

*Distributed algorithm:* Nodes exchange control information to collectively compute certain global characteristics. In a result, each node knows a global state of the system based on direct observations of neighbors behavior and indirect information from its neighbors about the others. An example is computation of reputation values of nodes. The key problems are low information propagation speed and trust issues.

Local models effectively support adaptive algorithms, which each node should perform during its participation in the system. Based on a local model the node makes own decisions. Directly observable information is used for input, indirect information about non-neighbor nodes is ignored.

Although being local, such models also support distributed implementation. Individual decision-making at each node provides new input information for other nodes, i.e., the system evolution is iterative. Nevertheless, the parallel activity of many nodes is loosely coupled and not tightly synchronized. Any node affects only a small subset of other nodes. The information propagation speed in the distributed system can be very low as well as the communication latency becomes a significant factor.

Global models are limited to high-level characteristics, which are relatively stable and can be effectively computed within the system. Local models suffer from the lack of relevant information to be used in control actions. Models that are in between local and global ones can be constructed based

on their extreme analogues. That is, a local model can be augmented with additional knowledge, e.g., routing benefits when a node can look ahead its neighbors. Similarly, a global model can limit itself with recently available part of global knowledge, e.g., ranking using a smaller graph instead of global topology graph.

#### B. Related Mathematical Techniques

Techniques of network analysis form the base for mathematical modeling in the P2P area. Various graph characteristics and algorithms behind are used. Examples are network diameter, which is important for routing [29], [30], and network connectivity structure, which provides information for ranking [31], [32]. In particular, this kind of modeling explains [29] the fundamental limit that the local knowledge problem induces. The diameter of a network with  $N$  nodes and equally fixed node degree  $m = |T_u| \forall u \in N$  has the logarithmic lower bound

$$D = \max_{u,v \in N} \min_{u \rightarrow^+ v} |u \rightarrow^+ v| \geq \lceil \log_m (N(m-1) + 1) \rceil - 1.$$

It is a direct consequence of the Moore bound

$$N \leq 1 + m + m^2 + \dots + m^D = \frac{m^{D+1} - 1}{m - 1}.$$

When all the global relations of nodes are known and nodes behavior can be completely specified as some types of stochastic processes then the methods of queuing networks can be applied, e.g., see [33], [34]. Each node is modeled as a processor with certain arrival and service patterns. Similarly, certain equation systems can describe node population dynamics or resource distribution evolution, based on techniques of fluid models and control theory [35], [36]. Such models allow a node to understand the current global state and react correspondingly. For large-scale P2P systems this approach often becomes impractical since a node can track no global and precise view to the dynamic network topology and characteristics of all participants.

As in many dynamic systems, nodes make own decisions based on observations (feedback)—the problem studied in control theory. Nodes consider the entire P2P system as a plant to be controlled [37]. Each node tracks contributions from other nodes and adopts its own contribution accordingly. These observations of others' contributions represent a global state of the system, hence leading to similar difficulties of global knowledge as in other models of global system state.

Nodes can reduce the observation space to direct observations from neighbors and aggregate indirect observations from the rest of the system. This approach is used in reputation schemes, and distributed learning-propagation algorithms are required [38], [39]. Indirect observations suffer from malicious nodes. The propagation efficiency can be low in large-scale high-dynamic systems.

Game theory focuses on P2P exchange economies with system-wide indexes like the overall node reputation metric or global resource prices, e.g., see [25], [40], [41]. It allows clear economic models with the intuition and properties from classic monetary-based economies. If a node knows the index

then the node can rationally select its participation strategy. The system converges to an equilibrium point, e.g., when there is no node that can improve its utility by deviating from the optimal strategy (Nash equilibrium). A distributed algorithm is required for computing such indexes, similarly to reputation learning-propagation schemes.

### C. Basic Arrangement Rules

Local and global models provide the system and its nodes with information for decision-making. Control processes are adaptive and based on differentiation: selection of the best control action for given input at given time. Although there exist many rigorous mathematical methods, all of them conceptually follow one of the basic arrangement rules, listed below.

They are applicable in P2P systems for taking the heterogeneity of nodes into account [9]. Let  $\mathcal{X}$  be a set that represent some knowledge about the system. For example,  $\mathcal{X} = N$  is the set of all nodes in the system or  $\mathcal{X} = N_u$  consists of all neighbors of some  $u \in N$ . In general case,  $u$ 's knowledge of  $\mathcal{X}$  may differ from the knowledge of other nodes  $v \neq u$ .

*Ordering:* A node  $u$  uses a binary relation  $\prec$  such that for any  $x, y \in \mathcal{X}$  either  $x \prec y$ ,  $y \prec x$  or  $x = y$ . In other words,  $u$  can arrange elements of  $\mathcal{X}$  in accordance with some "preference". The following two rules are extensions (continuous and discrete) of the ordering rule.

*Ranking:* There is a rank function  $r : \mathcal{X} \rightarrow \mathbb{R}$ , and  $u$  computes a real numerical value  $r(x)$  for each  $x \in \mathcal{X}$ . Thus, elements of  $\mathcal{X}$  are ordered on the real line  $\mathbb{R}$ . The important additional information is the value  $|r(x) - r(y)|$ , which is the preference level for  $u$  to compare  $x$  and  $y$ .

*Classifying:* The elements of  $\mathcal{X}$  are categorized into groups or levels  $i = 1, 2, \dots, M$  according to the preference. Although this rule leads to less precision than the ranking rule the former allows tradeoffs between the complexity and accuracy. For convenience, we assume that  $i = 1$  is "most preferable" and  $i = M$  is "least preferable".

## IV. LOCAL MODELS

Selected models of this section illustrate that moderate amounts of input information can be satisfactory for making individual decisions on nodes. Distributed activity of such local decision-makers achieves efficient routing and fair co-operation on the system level.

### A. Progressive Next Hops

The known routing efficiency of DHT is due to specific arrangements in local routing tables. A common case is routing within  $O(\log N)$  hops. In fact, the small-world phenomenon, which supports the existence of short paths, is a result of the local activity of nodes.

*1) Small-World Neighborhood:* A node classifies its neighbors onto local and long-range (Fig. 2). Local neighbors are similar in terms of the distance. Long-range neighbors are arranged in  $u$ 's vicinity.

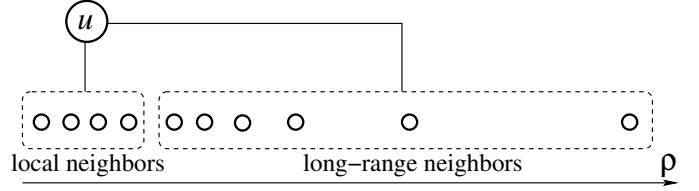


Fig. 2. The small-world hierarchy of neighbors: 1) classification onto local and long-range; 2) arrangement of long-range neighbors with vicinity clustering at all distance scales

Kleinberg's small-world construction [30] highlights the idea of selection distant nodes for neighbors. The node ID space is the  $n \times n$  grid

$$S = \{1, 2, \dots, n\} \times \{1, 2, \dots, n\} \\ \text{with } \rho((x_1, y_1), (x_2, y_2)) = |x_2 - x_1| + |y_2 - y_1|,$$

where  $N \leq n^2$ . Local neighbors of  $u$  are all  $v$  such that  $\rho(u, v) \leq \epsilon$  for a universal constant  $\epsilon \geq 1$ . Additionally,  $u$  selects  $m \geq 1$  long-range neighbors with probability proportional to  $[\rho(u, v)]^{-\alpha}$  for  $\alpha \geq 0$ . The parameter  $\alpha$  controls the density of long-range neighbors at all distance scales. When  $\alpha = 0$  the neighbors are distributed uniformly. As  $\alpha$  increases, they become more clustered. Greedy selection (4) of next hops leads to paths with the expected length  $O(\log^2 N)$  if  $\alpha = 2$ .

Symphony [17] is a particular inspiration of Kleinberg's small-world construction. The ID space is a ring (the unit interval  $[0, 1) \bmod 1$ ). The distance  $\rho(u, k)$  is the clockwise arc length from  $u$  to  $k$ . It is asymmetric, and routing is unidirectional (clockwise). A node  $u$  is responsible for the arc  $[u, v)$  where  $v$  is closest to  $u$ . Node IDs have uniform distribution in  $S$ . Every node maintains  $m' = 2$  local and  $m \geq 1$  long-range neighbors. Local neighbors are the closest nodes on the ring. A node  $u$  selects a long-range neighbor  $v$  closest to the point  $u + \delta \bmod 1$  for a random  $\delta$ . The generation of  $\delta$  is independent with the harmonic probability density function  $h(x) = 1/(x \ln N)$  for  $x \in [1/N, 1)$ , where  $N \geq 2$  is estimated locally. Intuitively,  $\delta = 1/N$  is the mean distance to  $u$ 's closest node and  $\delta = 1$  is supreme for the most distant one. Hence all candidates for long-range neighbors are in  $[u + 1/N, u + 1) \bmod 1$ . Greedy selection (4) in Symphony results in paths of the expected length

$$\mathbb{E}[|u \rightarrow^+ d|] = \begin{cases} O\left(\frac{1}{m} \log^2 N\right) & \text{if } m = O(1), \\ O(\log N) & \text{if } m = \Omega(\log N). \end{cases} \quad (6)$$

The local state conditions in (6) define tradeoffs in the amount of local information per node.

*2) Geometrically Progressive Routing:* A hop  $u \rightarrow v$  for  $k$  is called  $b$ -progressive (geometrically progressive with the ratio  $1/b$ ) if

$$\rho(v, k) \leq \frac{\rho(u, k)}{b} \quad \text{for } b > 1. \quad (7)$$

That is, the current distance is reduced at least by  $b$ .

Routing is called geometrically progressive if  $u$  always selects  $b$ -progressive hops in resolving lookups. It can be easily proved [4] that geometrically progressive routing has

logarithmic routing efficiency in the worst case, i.e., the number of hops to resolve a lookup is at most  $O(\log_b N)$ .

The progress  $b_j > 1$  at each hop  $j$  may be different due to the majorant sequence

$$\{N/(b_1 b_2 \cdots b_j)\}_{j=1}^l \leq \{N/b^j\}_{j=1}^l$$

with  $b = \min\{b_j \mid j = 1, 2, \dots, l\} > 1$ .

To implement geometrically progressive routing the system design must ensure that every node knows a  $b$ -progressive hop for any key. In Symphony, the long-range neighbor selection is based on a continuous model for the distance scales. It approximates geometrically progressive routing since some hops may be not 2-progressive. As a result, constructing  $O(\log N)$  paths is not possible when the number of neighbors per node is constant. Nevertheless, many popular practical DHT implementations uses designs that accurately follow the principle of geometrically progressive routing.

### B. Local Ranks

Nodes should exchange resources cooperatively. If a node does not contribute its resources it must be debarred. Such decisions can be implemented based on local information only.

1) *Single-Resource Exchange*: BitTorrent [28] is an example of P2P system for single resource exchange. A BT-system shares node bandwidth—bilateral single-resource exchange. In this exchange, resource rank  $s$  is needless ( $s = 1$ ). The BT consumption rule is “consume as much as you can from any neighbor” makes node rank  $r^+$  also needless. The provision process in Table III from Section II becomes one-phase:  $u$ 's upload bandwidth is allocated as  $b_v = r_v^- b$  for  $v \in N_u$ .

A node  $u$  provides its upload bandwidth  $b = \sum_{v \in N_u} b_v$  and consumes download bandwidth  $a = \sum_{v \in N_u} a_v$ . The incentives for  $u$  are straightforward: if  $u$  makes  $b_v(t)$  low then  $v$  may reduce  $a_v(t+1)$ . Formally,  $u$  cooperates with  $v$  if there exists  $\delta_v \geq 0$  such that for any  $t$

$$a_v(t) - b_v(t) \leq \delta_v, \quad (8)$$

where  $d_v(t) = a_v(t) - b_v(t)$  is recent surplus of  $v$ .

Low provision  $b_v$  limits consumption  $a_v$ , up to  $a_v \leq \delta_v$  for  $b_v = 0$ ; high consumption  $a_v$  is possible only if  $u$  provides appropriately high  $b_v$ . Parameter  $\delta_v$  is an upper imbalance bound that  $v$  tolerates. If  $\delta_v = 0$  then  $u$  has no credit, and unit consumption requires provision in advance or immediate response (difficult to realize in open network environment). If  $v$  is altruistic then  $\delta_v = \infty$ .

The provision process in period  $t + 1$  aims at making  $b_v(t+1) = r_v^- b$ . In original BitTorrent [28],  $u$  uploads to  $n$  top neighbors from  $N_u$  sorted by  $a_v$ , i.e.,  $r_v^- = 1/n$  for  $v \in N_u^{\text{best}}$  and  $r_v^- = 0$  otherwise. In proportional sharing [41], [42], all neighbors are fed with  $r_v^- = a_v/a$ . Block-based BitTorrent [43]–[45] immediately sets  $r_v^- = 0$  if  $a_v - b_v$  is lower a threshold. FairTorrent [46] makes instant decisions with one upload per period—to the neighbor with highest rate  $a_v - b_v$ , reducing the largest term in  $\sum_{v \in N_u} (a_v - b_v)$ .

The above decision-making aims at the rational use of (8) with uncertainty about  $\delta_v$ . On one hand,  $d_v < 0$  is unprofitable

for  $u$ . On the other hand,  $u$  has to merely assume that  $\delta_v$  is small, and  $d_v > 0$  signals on possible violation in (8).

The traditional technique of operations research can be applied to this problem. Consider ranks  $r_v^-$  that minimize expected deviation from the balance  $d_v(t+1) = 0$ . Setting  $b_v(t+1) = r_v^- b$  and approximating unknown in advance  $a_v(t+1)$  with observed  $a_v = a_v(t)$ , we yield the following optimization problem (the mean-square deviation was selected).

$$\begin{cases} \sum_{v \in N_u} (a_v - r_v^- b)^2 \rightarrow \min \\ \sum_{v \in N_u} r_v^- = 1, \\ r_v^- \geq 0. \end{cases} \quad (9)$$

The solution to (9) is

$$r_v^- = \frac{a_v}{b} + \frac{b-a}{|N_u|b} \quad \text{if} \quad \min_{v \in N_u} a_v \geq \frac{a-b}{|N_u|}, \quad (10)$$

where  $\frac{b-a}{|N_u|b}$  is average relative skew of  $u$ 's participation in the system. The condition in (10) is due to  $r_v^- \geq 0$ . If  $a = b$  then  $r_v^- = a_v/b$ , coinciding with proportional sharing [41], [42].

If  $u$  provides in sum less than it consumes ( $a > b$ ) then there can be neighbors lower the average ( $a_v < (a-b)/|N_u|$ ), and solving (9) becomes more complicated. Some BT-exchanges use the discrete heuristic: highest  $d_v$  are important only. If  $a_v(t) \gg b_v(t)$  then  $u$  takes  $r_v^- > 0$  to keep  $v$ 's generosity and sets  $r_v^- = 0$  otherwise. The case is similar to (9) but ranking is reduced to the best neighbors  $N_u^{\text{best}} \subset N_u$ .

When  $u$  has limited consumption capacity then node consumption rank  $r^+$  can be computed similarly to (9) and (10). The solution is symmetrical to (10):

$$r_v^+ = \frac{b_v}{a} + \frac{a-b}{|N_u|a} \quad \text{if} \quad \min_{v \in N_u} b_v \geq \frac{b-a}{|N_u|}. \quad (11)$$

Model (9) captures the property important for reciprocal incentives. If  $u$  minimizes deviation from the balance  $d_v = a_v - b_v = 0$  then  $u$  supports preserving (8), even in unfavourable cases  $\delta_v \approx 0$ .

Ranks  $r_v^- = r_v^-(a_v)$  in (10) and  $r_v^+ = r_v^+(b_v)$  in (11) are non-decreasing functions on  $a_v$  and  $b_v$ , respectively. It allows generalization of the BT heuristics and incentives. Instead of fixing an optimization problem which ranks are solutions to, let us assume that node ranks  $r_v^-$  and  $r_v^+$  are monotone functions of  $x = d_v = a_v - b_v$ , reflecting imbalance between the actual provision and consumption as shown in Fig. 3.

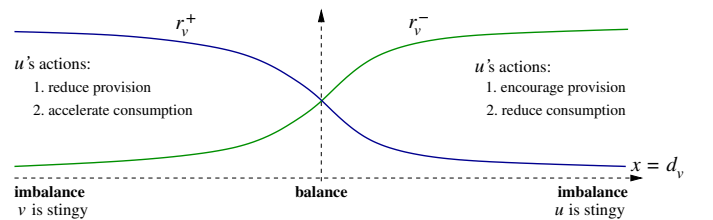


Fig. 3. Local single-resource exchange with  $v \in N_u$ . Let  $u$  have surplus  $x = d_v$  for given  $v$ . Activity of other neighbors  $w \neq v$  is fixed (in terms of  $d_w$ ). Node ranks  $r_v^-(x)$  and  $r_v^+(x)$  behave monotonically



2) *Multi-Resource Exchange*: In multi-resource exchange, surplus of  $v$  is a sum of its resource surpluses,  $d_v = \sum_{k \in R_v} d_{vk}$ . Resource compensation is possible since  $u$  operates with a set of resources. For example,  $u$  tolerates low provision  $a_{vk}$  from  $v$  because of high provision  $a_{wi}$  for  $k, i \in R_u$  and  $i \neq k$ . Cross-node resource compensation can also happen: small  $a_{vk}$  and high  $a_{wi}$  for  $w \neq v$ .

Node ranks  $r^+$  and  $r^-$  are functions of the surplus matrix  $D = \{d_{vk}\}$ . Varying  $d_{vk}$  when other surpluses are fixed leads to appropriate changes in ranks to reflect the imbalance. Let every  $d_{wi}$  be fixed about  $x = d_{vk}$ . Then  $r^+$  and  $r^-$  are local node ranks if and only if they satisfy the following properties

*Monotony*:  $r_v^-(x)$  is a non-increasing function and  $r_v^+(x)$  is a non-decreasing function.

*Marginality*:  $\lim_{x \rightarrow -\infty} r_v^-(x) = \lim_{x \rightarrow +\infty} r_v^+(x) = 0$  and  $\lim_{x \rightarrow +\infty} r_v^-(x) = \lim_{x \rightarrow -\infty} r_v^+(x) = 1$ .

There are incentives for  $u$  to cooperate with  $v$  if there is  $\delta_v \geq 0$  such that for any  $t$

$$\sum_{k \in R_u} (a_{vk}(t) - b_{vk}(t)) \leq \delta_v, \quad (12)$$

where  $d_v(t) = \sum_{k \in R_u} (a_{vk}(t) - b_{vk}(t))$  is sum surplus of  $v$ .

Although the same resource can be used in exchanges with different nodes there is no neighbor that parasitizes because of generosity of some other neighbors. The latter easily happens when  $u$  considers only the overall sum:

$$\sum_{v \in N_u} \sum_{k \in R_u} (a_{vk}(t) - b_{vk}(t)) \leq \delta.$$

A node  $u$  realizes the incentives by balancing near  $d_v = 0$  for all  $v \in N_u$ : high  $d_v$  likely violates (12) and low  $d_v$  is unprofitable for a rational node. Expected deviation is minimized by controlling  $b_{vk}(t+1)$  for  $v \in N_u$  and  $k \in R_u$ . For each  $k$  reaction to observation  $a_{vk}(t)$  may affect  $b_{vk}(t+1)$  as well as  $b_{vi}(t+1)$  for some  $i \neq k$ . Differentiation is needed to select appropriate resources for the reaction. Resource ranks  $s_k \geq 0$  quantitatively implement the differentiation.

Resource rank needs different assumptions than node rank. In the latter case, a neighbor  $v$  is both provider and consumer; each role receives its own rank. The simplest case for resource rank happens if any resource  $k$  has a single role being either local ( $u$  provides  $k \in R_u^-$ ;  $a_{vk} = 0, b_{vk} \geq 0 \forall v \in N_u$ ) or external ( $u$  consumes  $k \in R_u^+$ ;  $a_{vk} \geq 0, b_{vk} = 0 \forall v \in N_u$ ).

A generalization is transit resource, which can be used for better trading opportunities in resource exchange. For example, assume  $u$  downloads file  $k$  from  $v$  ( $a_{vk} > 0$ ); the file is not of interest of  $u$  itself; uploading  $k$  to  $w$  ( $b_{wk} > 0$ ) allows  $u$  to download another file  $i$  from  $w$  ( $a_{wi} > 0$ ). Hence  $k$  is transit resource that  $u$  uses to receive another resource  $i$ . A particular case of transit resource is bidirectional resource ( $a_{vk}, b_{vk} > 0$ ), e.g., bandwidth in BT-systems.

Denote the net consumption and provision

$$\alpha_{vk} = \begin{cases} d_{vk} = a_{vk} - b_{vk}, & \text{if } d_{vk} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

$$\beta_{vk} = \begin{cases} -d_{vk} = b_{vk} - a_{vk}, & \text{if } d_{vk} < 0, \\ 0, & \text{otherwise.} \end{cases}$$

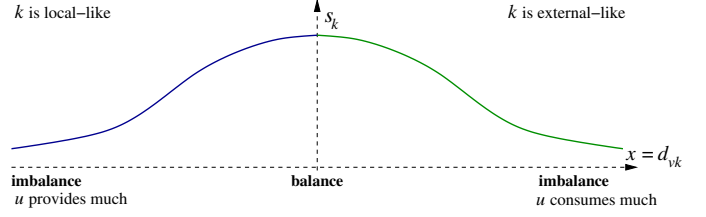


Fig. 4. Qualitative behavior of resource rank  $s_k$  as a function of surplus  $x = d_{vk}$  when other activity is fixed. If  $k$  is pure local or pure external then  $s_k(x)$  is defined on  $(-\infty, 0]$  or  $[-\infty, 0)$ , respectively. The function is neither required to be symmetrical  $s_k(x) = s_k(-x)$  nor contiguous

Clearly,  $d_{vk} = \alpha_{vk} - \beta_{vk}$ . Let  $R_v^+ = \{k \mid \alpha_{vk} > 0\}$ ,  $R_v^- = \{k \mid \beta_{vk} > 0\}$ , and  $R_v = R_v^+ \cup R_v^-$ . The notation emphasizes that  $R_v^+$  consists of external-like resources (currently under  $u$ 's consumption) and  $R_v^-$  consists of local-like resources (currently under  $u$ 's provision). Then  $R_u^+ = \bigcup_{v \in N_u} R_v^+$  and  $R_u^- = \bigcup_{v \in N_u} R_v^-$ . Although  $R_v^+ \cap R_v^- = \emptyset \forall v \in N_u$  the set  $R_u^+ \cap R_u^-$  can be nonempty.

Let  $d_k = \sum_{v \in N_u} d_{vk}$  be surplus of  $k$  for  $u$ . Resource  $k$  is balanced when  $d_k = 0$ , i.e., neighbors provide in sum the same amount of  $k$  as they consume. The less  $|d_k|$  the more preferable for  $u$  to consume or provide  $k$ . Hence  $s_k$  can be treated a balance closeness index.

Let every  $d_{wi}$  be fixed about  $x = d_{vk}$ . Then  $s$  is local resource rank if and only if  $s$  satisfies the following properties for all  $v \in N_u$  and  $k \in R_v$ .

*Monotony*:  $s_j(x)$  is a non-decreasing function on  $(-\infty, 0]$  for any  $j \in R_v^-$  and  $s_i(x)$  is a non-increasing function on  $[0, \infty)$  for any  $i \in R_v^+$ .

*Marginality*:  $\lim_{x \rightarrow -\infty} s_j(x) = \lim_{x \rightarrow +\infty} s_i(x) = 0$  for any  $i \in R_v^+$  and  $j \in R_v^-$ .

The qualitative behavior is shown in Fig. 4. Intuitively, when  $d_{vk}$  becomes high then  $s$  prioritizes reduction in provision of any external-like resource  $i \in R_v^+$ , including  $k$  (since  $k \in R_v^+$  if  $d_{vk} > 0$ ). When  $d_{vk}$  becomes low then  $s$  prioritizes reduction in consumption of any local-like resource, including  $k$  (since  $k \in R_v^-$  if  $d_{vk} < 0$ ).

Resource rank induces node ranks for every  $v \in N_u$  as “sum rank” of resources available through  $v$ . Let

$$r_v^+ = \sum_{i \in R_v^+} s_i, \quad r_v^- = \sum_{j \in R_v^-} s_j, \quad (13)$$

### C. Linear Ranking

In multi-resource exchange, resource ranks become essential for proper differentiation of nodes. Let us discuss a model of resource ranks, which defines them as solutions to a homogeneous linear equation system. It states local exchange balance for all neighbors and resources. Computation is reduced to a linear programming problem.

1) *Linear resource ranks*: Let a node  $u$  maintain surplus counters  $d_{vk} = \alpha_{vk} - \beta_{vk}$  for all its neighbors  $v$  and resources  $k$ . Consider the following homogenous system of  $n = |N_u|$  equations in  $m = |R_u|$  non-negative unknowns.

$$\sum_{k \in R_u} d_{vk} s_k = 0, \quad v \in N_u. \quad (14)$$



Let  $\|s\|_1 = \sum_{k \in R_u} |s_k| = \sum_{k \in R_u} s_k$  (i.e., the  $L_1$  norm). If  $s \neq \mathbf{0}$  (non-trivial) then  $s/\|s\|_1$  is a normalized solution. Linear resource rank is defined as any normalized solution to (14) or the trivial solution if (14) has no other solutions.

Given a linear resource rank  $s$ . Intuitively, if  $s_k = 1/m$  then  $k$  is balanced. If  $s_k < 1/m$  then the lower rank value is due to  $k$  excites the imbalance. If  $s_k > 1/m$  then the higher rank value is due to other resources excite the imbalance.

Now consider a linear rank  $s$  with  $s_k = 0$ . Then the rank-proportional allocation ignores  $k$ . The knowledge  $\{d_{vk}\}_{v \in N_u}$  is unused though  $d_{vk} \neq 0$  for some  $v$  and  $k$ . Consequently,  $k$  does not influence to other ranks, as if  $k$  is not presented in (14). Hence, there is no differentiation compared with inactive resources. Furthermore, if there are several such  $k$  then there is no differentiation between them.

Linear resource rank  $s$  is effective if  $s_k > 0 \forall k \in R_u$ . Rejection of too imbalanced resources is possible when needed: if  $0 < s_k < \varepsilon_0$  then  $k$  is worse than an inactive resource. Additionally, high  $s_i$  does not necessarily lead to high productivity with  $i$ , e.g., (a)  $u$  requests much  $i$  but the neighbors suddenly become stingy, (b) altruistic neighbors provide much  $i$  in spite of  $u$ 's low response, or (c)  $u$  can provide much  $i$  but the neighbors request less. The capacity should be redistributed proportionally to ranks, including low-ranked resources since their sum share in  $R_u$  can be appreciable. In particular, it diminishes the problem of BT-exchange when many nodes are excluded from participation or underutilize capacity.

Given  $v \in N_u$  a  $v$ -equation is the corresponding equation in (14). Rewrite each  $v$ -equation such that its unknowns appear with positive coefficients:

$$\begin{cases} \sum_{i \in R_v^+} \alpha_{vi} s_i = \sum_{j \in R_v^-} \beta_{vj} s_j, & v \in N_u, \\ s \geq \mathbf{1}. \end{cases} \quad (15)$$

Given  $i \in R_v^+$  and  $j \in R_v^-$  define the  $i$ th and  $j$ th  $v$ -families of solutions:

$$\begin{aligned} \mathcal{H}_{vj} &= \left\{ h = \begin{pmatrix} \sigma \\ \mathbf{e}_j \end{pmatrix} \mid \sum_{k \in R_v^+} \alpha_{vk} \sigma_k = \beta_{vj}, \sigma_k \geq 0 \right\}, \\ \mathcal{H}_{vi} &= \left\{ h = \begin{pmatrix} \mathbf{e}_i \\ \sigma \end{pmatrix} \mid \alpha_{vi} = \sum_{k \in R_v^-} \beta_{vk} \sigma_k, \sigma_k \geq 0 \right\}. \end{aligned} \quad (16)$$

Hence  $\mathcal{H}_{vj}$  contains solutions to the  $v$ -equation such that  $s_j = 1$  and  $s_k = 0$  for  $k \in R_v^- \setminus \{j\}$ . Similarly,  $s \in \mathcal{H}_{vi}$  is such that  $s_i = 1$  and  $s_k = 0$  for  $k \in R_v^+ \setminus \{i\}$ .

Further let vectors in  $\mathcal{H}_{vj}$  and  $\mathcal{H}_{vi}$  be of the same dimension  $m = |R_u|$  as solutions to (14) or (15). Since the  $v$ -equation does not necessarily contain all unknowns of the whole system,  $s_k$  takes arbitrary non-negative values for  $k \in R_u \setminus (R_v^+ \cup R_v^-)$ .

Construct the following set of solutions to  $v$ -equation.

$$\mathcal{H}_v = \left\{ s = \sum_{k \in R_v} \lambda_k h^{(k)} \mid h^{(k)} \in \mathcal{H}_{vk}, \lambda_k \geq 1 \right\}. \quad (17)$$

A solution  $s$  to (14) is  $v$ -effective rank if and only if  $s \in \mathcal{H}_v$ . Any  $v$ -effective rank contains one  $h^{(k)} \in \mathcal{H}_{vk}$ , i.e., atomic contribution of  $k$ . Every resource  $k$  always "influences" to the rank with a level  $\lambda_k$ , where  $\lambda_k \geq 1$  is a lower bound. Each  $v$ -family in (16) defines the set of all atomic contributions—a subset of the general solution to the  $v$ -equation.

2) *Reduction to Linear Programming:* To construct a  $v$ -effective rank one must select exactly one  $h^{(k)}$  among many candidates in  $\mathcal{H}_{vk}$  for each  $k$ . Any  $v$ -effective rank  $s \in \mathcal{H}_v$  is a solution to the linear system

$$\begin{cases} \sum_{i \in R_v^+} \alpha_{vi} s_i = \sum_{j \in R_v^-} \beta_{vj} s_j, \\ s_k \geq 1, \quad k \in R_v^+ \cup R_v^-. \end{cases} \quad (18)$$

The converse is not true. The accuracy is within multiplication to a constant: there exists a  $v$ -efficient rank  $s'' = Cs'$  for any solution  $s'$  to (18) and a constant  $C \geq 1$ . Consequently, there is no difference whether to use  $s'$  or  $s''$  when computing the normalized rank.

A rank is effective if it is  $v$ -effective for all  $v \in N_u$ . The set of all effective ranks is  $\mathcal{H} \subseteq \bigcap_{v \in N_u} \mathcal{H}_v$ . They are among solutions to (15). A linear programming problem can formulated to find the best effective rank in  $\mathcal{H}$ .

Depending on specifics of the resource sharing problem some ranks account  $u$ 's interests better. We can bound  $s_i$  since if (18) is solvable then there are solutions with arbitrary high  $s_k$ . Note that normalization preserves the proportions despite of how high the absolute values are. The following generic linear cost function provides the bound if  $c > 0$ :

$$\sum_{k \in R} c_k s_k \rightarrow \min. \quad (19)$$

The weights  $c_k$  allow additional problem-aware resource differentiation; the higher  $c_k$  the lower  $s_k$ .

From the point of view of market pricing, if many nodes provide resource  $i \in R^+$  then its rank decreases. If many nodes consume resource  $j \in R^-$  then its rank increases. Hence, let  $c_i$  count all nodes  $v \in N_u$  such that  $i \in R_v^+$  and let  $c_j$  count all  $v$  such that  $j \in R_v^-$ . These counters appear implicitly in node ranks from (13), i.e.,

$$\begin{aligned} \sum_{v \in N_u} r_v^+ &= \sum_{v \in N_u} \sum_{i \in R_v^+} s_i = \sum_{i \in R} c_i s_i, \\ \sum_{v \in N_u} r_v^- &= \sum_{v \in N_u} \sum_{j \in R_v^-} s_j = \sum_{j \in R} c_j s_j. \end{aligned}$$

Table IV shows possible linear cost minimization criteria. Criterion (C+P) states a tradeoff since neighbor consumption and provision ranks are bounded equally. Criteria (C-P) and (P-C) state thresholds of moving toward the respectable and selfish direction, correspondingly. Obviously, they can lead to an unbounded optimization problem.

This ranking model allows a generic function  $f(r_v^+, r_v^-)$  that specifies the relation between  $r_v^+$  and  $r_v^-$ . For instance,  $u$  can prioritize nodes  $v \in N_u$  using mutual trust among them and use quadratic cost functions

$$\sum_{v \in N_u} c_v (r_v^+ - r_v^-)^2 \rightarrow \min.$$

Instead of the use of homogenous linear constraints one can consider the equivalent linear programming problem:

$$\begin{cases} f = \sum_{k \in R_u} d_k \sigma_k \rightarrow \min \\ \sum_{k \in R_v} d_{vk} \sigma_k = -d_v, \quad v \in N_u, \\ \sigma_k \geq 0 \quad \forall k \in R_u. \end{cases} \quad (20)$$

Any feasible solution  $\sigma$  to (20) yields a solution  $s = \sigma + 1$  to (15) and vice versa. The cost function  $f$  prioritizes resources by their surplus—resources with high  $d_k$  will be of low rank.

The dual problem to (20) is

$$\begin{cases} \phi = \sum_{v \in N_u} d_v \rho_v \rightarrow \min \\ \sum_{v \in N_u} d_{vk} \rho_v \leq d_k, \quad k \in R_u. \end{cases} \quad (21)$$

(The dual variables  $-\rho_v$  take any real value.) According to the duality,  $\rho_v = \frac{\partial f}{\partial d_v}$ , i.e.,  $\rho_v$  arranges nodes by their influence to the resource-aware cost. Consequently, they can be used to compute node ranks, alternatively to (13).

## V. GLOBAL MODELS

Global models are limited to high-level characteristics, which can be determined before the system installation or are relatively stable allowing inexpensive computation run-time. This section considers models that support architectural solutions (structure and resultant performance) and fair cooperation (global reputation).

### A. Hierarchical Architectures

Kleinberg [30] considered three conceptual models for decentralized networks: grids, hierarchies, and set systems. It forms a general framework for hierarchical multi-layer architectures for P2P systems [4].

1) *Cluster-based Model*: P2P clusters are a result of classification. Close nodes become densely connected and form a discrete entity—the cluster with explicit space bounds. Within a cluster each node can reach another in few hops. A global structure of tightly interconnected groups appears, forming a two-layer hierarchy. In intra-cluster communication the routing distance  $\tau$  is shorter than in inter-cluster communication.

This model defines a hierarchy as a set of  $n$  interconnected clusters. It embeds the network such that its network topology ensures the following routing distance property:

$$\left( \{\mathcal{C}_s\}_{s=1}^n, \tau \right), \quad \tau(u, v) \ll \tau(u, w) \text{ for } u, v \in \mathcal{C}, w \in \mathcal{C}', \mathcal{C} \neq \mathcal{C}'. \quad (22)$$

Each particular model defines its variant of the relation “ $\ll$ ”.

TABLE IV  
LINEAR OPTIMIZATION CRITERIA

Ref.	Cost	Interpretation
(C)	$\sum_{v \in N_u} r_v^+$	Higher priority in $u$ 's consumption to rare resources since few nodes provides them.
(P)	$\sum_{v \in N_u} r_v^-$	Higher priority in $u$ 's provision to neighbors consuming resources of small variety.
(C+P)	$\sum_{v \in N_u} (r_v^+ + r_v^-)$	A trade-off in $u$ 's consumption/provision.
(C-P)	$\sum_{v \in N_u} (r_v^+ - r_v^-)$	A respectability threshold in $u$ 's consumption/provision: $u$ minimizes its consumption and maximizes its provision.
(P-C)	$\sum_{v \in N_u} (r_v^- - r_v^+)$	A selfishness threshold in $u$ 's consumption/provision: $u$ minimizes its provision and maximizes its consumption.

A simple formal construction is a set of balls in the ID space  $S$ , e.g., in grid. There is a set of landmarks  $\{c_s\}_{s=1}^n$ ; they are points or dedicated nodes in  $S$ . Cluster  $\mathcal{C}_s$  consists of all nodes  $u$  that satisfy  $\rho(c_s, u) \leq R$  for a given radius  $R$ .

2) *Tree-based model*: A given hierarchy in  $S$  with two or more layers induces the space distance  $\rho$  in terms of the hierarchy tree. The hierarchy embeds the network, forming the hierarchy-aware network topology with the resultant routing distance  $\tau$ . An obvious example is location and administrative hierarchies: nodes are categorized on lowest-level groups depending on which local area network they belong to, the upper level groups are defined in accordance with their scale, e.g., city, region, and state.

A formal construction is based on a complete  $b$ -ary tree  $\mathcal{T} = \mathcal{T}(b, N)$  with  $N$  leaves (hence  $\mathcal{T}$  is of height  $M = \log_b N$ ). For leaves  $u$  and  $v$  the hierarchy-induced distance  $h(u, v)$  is the height of their lowest common ancestor in  $\mathcal{T}$ . Then a network of  $N$  nodes is constructed such that the probability of establishing a link  $u \rightarrow v$  is proportional to  $b^{-\alpha h(u, v)}$  for a model parameter  $\alpha \geq 0$ . As in the cluster-based model, shorter paths exist between nodes of the same group.

In general, an arbitrary tree  $\mathcal{T}$  defines a hierarchy where  $N$  leaves correspond to nodes. Other vertices are groups consisting of descendant groups and nodes. The distance metric is the tree distance. The hierarchy is  $M$ -layer. The bottom layer  $i = 1$  consists of  $N$  nodes. On upper layer  $i + 1$  each group consists of all nodes from its descendant groups on the layer  $i$ . Groups of the same layer are node disjoint. On each layer the inter-group connectivity is sparser than the intra-group connectivity.

3) *Group-based model*: Assume that each node may belong to several groups. For example, a node  $u$  can belong to group  $C_1$  (Petrozavodsk State University—administrative entity), group  $C_2$  (powerful machines—performance level) and group  $C_3$  (Russia—geographical area). As in the cluster- and tree-based models, nodes are more likely to be connected if they belong to the same group.

The following properties using parameters  $0 < \lambda < 1$  and  $1 < \mu$  are established for formal construction.

- (i) The full set  $N$  of all nodes is a group.
- (ii) If  $\mathcal{C}$  is a group of size  $|\mathcal{C}| \geq 2$  and  $u \in \mathcal{C}$ , then there is a group  $\mathcal{C}' \subset \mathcal{C}$  such that

$$\mathcal{C}' \neq \mathcal{C}, \quad u \in \mathcal{C}', \quad \lambda |\mathcal{C}| \leq |\mathcal{C}'| < |\mathcal{C}|.$$

- (iii) For any set of groups  $\{\mathcal{C}_i\}$  with a common node  $u$ ,

$$\left| \bigcup_i \mathcal{C}_i \right| \leq \mu \sigma, \quad \text{where } \sigma = \max_i |\mathcal{C}_i|.$$

Property (i) ensures that for any subset of nodes there is a group such that it includes the whole subset. Property (ii) is a type of the “hierarchy balance” requirement when a group consists of subgroups of proportional size. For example, the tree-based model with a complete  $b$ -ary tree defines groups of  $b$  node-disjoint subgroups each, hence  $\lambda \sim 1/b$ . Property (iii) is a type of “bounded size growth” requirement; if groups has a common node then they are close in certain sense, so they cannot contain many distinct nodes. For example, the

cluster-based model forms node-disjoint groups, hence a set in property (iii) always consists of one group.

For two nodes  $u$  and  $v$ , the induced space distance  $\rho(u, v)$  is the minimum size of a group containing both  $u$  and  $v$ . Similarly to the previous models, it allows embedding the network into the hierarchy such that the connectivity structure positively correlates the routing distance  $\tau$  with  $\rho$ . For instance, one can use the probability of establishing a link  $u \rightarrow v$  proportional to  $[\rho(u, v)]^{-\alpha}$  for  $\alpha \geq 0$ .

Properties (ii) and (iii) results in two dimensions in hierarchy: vertical and horizontal. The vertical dimension defines layers and rules for nesting groups. Any hierarchy contains at least one chain of nested groups for any node  $u$ :

$$u \in \mathcal{C}_1 \subset \mathcal{C}_2 \subset \dots \subset \mathcal{C}_m = N, \quad (23)$$

where each  $\mathcal{C}_i$  belongs to a distinct layer and there is no  $\mathcal{C}$  such that  $\mathcal{C}_i \subset \mathcal{C} \subset \mathcal{C}_{i+1}$ .

The tree-based model is pure vertical since the only chain of nested groups exists for a given node. In the general case, there can be several chains for the same node, e.g., if the tree-based model uses several trees. The following requirement preserves the vertical structure “approximately nested”. If  $\{\mathcal{C}_i\}$  are arbitrary groups having a common node and any two of them do not belong to the same layer then

$$\left| \bigcup_i \mathcal{C}_i \right| \leq \mu(\max_i |\mathcal{C}_i|), \quad (24)$$

where  $\mu(\cdot)$  is a monotone increasing function.

The horizontal dimension defines classification onto groups on the same layer  $i$ .

$$\bigcup_j \mathcal{C}_{ij} = N_i, \quad |\mathcal{C}_{ij} \cap \mathcal{C}_{ik}| \ll |\mathcal{C}_{ij} \Delta \mathcal{C}_{ik}| \quad \forall j \neq k, \quad (25)$$

where  $\Delta$  is the symmetric difference.

The cluster-based model is pure horizontal; its group distribution is always a partition of  $N$ , thus  $\mathcal{C}_{ij} \cap \mathcal{C}_{ik} = \emptyset$ . In the general case, groups on the same layer may overlap. Requirement (25) preserves groups “low overlapped”. Note that alternative chains in (23) can appear because of overlapping.

## B. Performance

Hierarchical architecture meets the problem of the optimal number of layers and the optimal population size on each layer. On one hand, the more layers the more differentiation, which improves the performance and other characteristics of the system. On the other hand, a high hierarchy degree can be expensive. Consider some cost models for this tradeoff: cost of local and total states (routing table sizes), routing (path length in overlay hops), and traffic (lookup and maintenance).

1) *State Cost in Two-Layer Architectures*: The total state cost is the sum of local state costs over all nodes. Consider a two-layer architecture with  $N_2$  supernodes, each maintains its own non-overlapping overlay of  $N/N_2$  nodes on the bottom layer. Any supernode has local state of  $N/N_2 + N_2 - 2$  entries due to maintenance of two routing tables of size  $N/N_2 - 1$  (all supernode's nodes) and  $N_2 - 1$  (all other supernodes).

Assume the extreme case when every node is ready to be a supernode (full-redundancy) and every overlay has fully-meshed topology. Consequently, any node needs to maintain  $N/N_2 + N_2 - 2$  entries, and the total state cost is

$$\text{StateCost}(N_2) = N \left( \frac{N}{N_2} + N_2 - 2 \right). \quad (26)$$

The only minimum is at  $N_2 = \sqrt{N}$ .

Assume the case with no redundancy. Every node maintains a routing table of size  $N/N_2 - 1$  to participate in its overlay on the bottom layer. In addition, every supernode maintains a routing table of size  $N_2 - 1$ . The total state cost becomes

$$\text{StateCost}(N_2) = N_2(N_2 - 1) + N(N/N_2 - 1). \quad (27)$$

Taking the derivate and equalizing it to zero lead to the following cubic equation respect to  $N_2$ :

$$2N_2^3 - N_2^2 - N^2 = 0.$$

There exists the only positive real root  $N_2 = \Theta(N^{2/3})$  for large  $N$ . Although model (27) states that the total state cost is less than for (26), more supernodes are required ( $N^{2/3} \gg N^{1/2}$  for large  $N$ ). The benefit of this node differentiation is that the local state cost ratio between supernodes and regular nodes is proportional to  $N^{1/3}$ .

Let any overlay in the network be DHT-based with logarithmic-size routing tables. Similarly to (27),

$$\text{StateCost}(N_2) = N_2 \log N_2 + N \log(N/N_2). \quad (28)$$

Taking the derivate and equalizing it to zero, one obtains the following transcendental equation respect to  $N_2$ :

$$N_2 = \frac{N}{1 + \ln N_2}.$$

Since  $\ln N_2 + 1 < N_2$  for large  $N_2$ , then  $N_2 \geq N/N_2$  and the upper bound  $N_2 = O(\sqrt{N})$  is true for the optimal solution. If we approximate  $N_2 = N^{1/2-\epsilon}$  with a small  $\epsilon > 0$  then the local state cost is  $(1/2 + \epsilon) \log N$  and  $\log N$  for a regular node and a supernode, respectively. The local state cost ratio between supernodes and regular nodes is  $2/(1 + 2\epsilon)$ .

2) *State Cost in Multi-Layer Architectures*: Generalization to  $M > 2$  is based on induction for the number of layers. Basic conclusions are as follows. Given  $N$ , the optimal number of layers is  $M = \Theta(\log N)$ . Given  $N$  and  $M$ , the optimal number of supernodes on the top layer is  $N_M = \Theta(N^{1/M})$ .

Let  $N_1 = N$  and  $N_{M+1} = 1$  in  $M$ -layer architecture. The total state cost is given by

$$\text{StateCost} = \sum_{i=1}^M N_i \delta_i \quad \text{for fixed } M, \quad (29)$$

where  $N_i$  is the number of nodes on layer  $i$  and  $\delta_i$  is the routing table size in any layer- $i$  overlay. When  $\delta_i = N_i/N_{i+1} - 1$  and  $\delta_i = \log(N_i/N_{i+1})$  we generalize (27) and (28), respectively.

Moving from  $M$ -layer architecture to  $(M + 1)$ -layer one, both  $N_i$  and  $\delta_i$  are changed because of node redistribution among layers. The redistribution essentially depends on the clustering algorithm: nodes on layer  $i$  form groups and elect

TABLE V  
ASYMPTOTES OF THE TOTAL STATE COST IN P2P ARCHITECTURES.

Architecture	Total state cost as big- $O$ asymptote
Flat logarithmic DHT	$N \log N$
Flat fully-meshed	$N(N-1)$
$M = 2$ , full-redundancy, fully-meshed, see (26)	$2N(\sqrt{N}-1)$
$M = 2$ , non-redundant, fully-meshed, see (27)	$N(2N^{1/3}-1-N^{-1/3})$
$M = 2$ , logarithmic DHTs, see (28)	$\frac{N+\sqrt{N}}{2} \log N$
$M = \Theta(\log N)$ , fully-meshed	$N \log N$

their representatives for layer  $i+1$ . Layers  $i = 1, 2, \dots, M$  become less populated ( $N_i$  is reduced) and overlays on these layers become smaller ( $\delta_i$  is reduced). It decreases the total state cost. On the other hand, the new top layer  $M+1$  leads to higher state cost for  $N_M$  nodes. The compensation of these two terms determines decrement or increment of the total cost.

In fully-meshed topology, increasing  $M$  up to  $\log N$  reduces the cost to  $\Theta(N \log N)$ . Further increasing does not provide better result than (26). In logarithmic DHT, any flat  $N$ -node DHT network already has the  $\Theta(N \log N)$  total state cost, and increasing  $M$  is not essential for the cost improvement. Table V summarizes the asymptotic cost behavior for large  $N$ .

3) *Routing cost*: Consider the routing cost model for the optimal value of  $N_2$  in two-layer CAN-based architecture [47]. Let  $d$  be the CAN ID space dimension. Starting from a regular node, the number of hops needed to find the destination cluster supernode on the top layer (a CAN overlay of  $N_2$  nodes) is  $1+dN_2^{1/d}$  on average [16]. Then  $d(N/N_2)^{1/d}$  hops are needed on average to find the responsible node on the bottom layer (a CAN overlay of  $N/N_2$  nodes). In total, the routing cost in overlay hops is

$$\text{RoutingCost} = f(N_2) = 1 + dN_2^{1/d} + d \left( \frac{N}{N_2} \right)^{1/d}.$$

The minimum is when  $N_2 = \sqrt{N}$ , independently on  $d$ . Fixing  $N_2 = \sqrt{N}$  the routing cost is minimal for  $d = \ln N_2$ . It is twice lower than the optimal  $d_{\text{flat}} = \ln N = \ln N_2^2 = 2 \ln N_2$  in a flat CAN network. Consequently, a regular node has twice lower local state. A supernode has the same state as in a flat CAN network due to maintenance of two routing tables, each has  $d$  entries ( $d+d = d_{\text{flat}}$ ). The optimal case is

$$\text{RoutingCost} = 1 + N^{1/\ln N} \ln N,$$

which is one hop greater than the optimal routing cost in a flat CAN network with  $N$  nodes and  $d_{\text{flat}} = \ln N$ .

Applying the same technique for a flat DHT network with  $O(\log N)$  routing,

$$\text{RoutingCost} = 1 + \log N_2 + \log \frac{N}{N_2} = 1 + \log N,$$

and the routing performance again is almost equal for hierarchical and flat designs.

### C. Reputation Systems

Nodes can utilize all direct observations from neighbors and aggregate indirect observations from the rest of the system. This approach is used in reputation systems, which require distributed learning-propagation algorithms to aggregate results of concurrent nodes [38], [39].

1) *Market Pricing*: Game theory focuses on P2P exchange economies [25], [37], [40], [41] with system-wide indexes like the overall node reputation metric or global resource prices. It results in models with the intuition and properties from classic monetary-based economies. If a node knows the index then the node can rationally select its participation strategy. The system converges to an equilibrium point, e.g., when there is no node that can improve its utility by deviating from the optimal strategy (Nash equilibrium).

The game-theoretic models are mostly based on defining node utility functions  $U(a(u), b(u))$ , where consumption  $a(u)$  and provision  $b(u)$  define  $u$ 's strategy. The sum of these utilities is the social welfare to be maximized. Less attention is given to constraints that define feasible strategies. The constraints typically reflect node capacity bounds only.

Aperjis *et al.* [25] considered the bilateral exchange balance where nodes charge each other for resources in a common monetary unit with settlement-free transactions. Node  $u$  charges node  $v$  a price  $p_{uv} > 0$  per unit rate. Then  $u$  has one exchange balance constraint for each neighbor  $v$ ,

$$p_{vu} \sum_{k \in R_u^+} a_{vk}(u) = p_{uv} \sum_{k \in R_u^-} b_{vk}(u), \quad v \in N_u,$$

where  $a_{vk}(u) = b_{uk}(v)$  is the rate at which  $u$  downloads resource  $k$  from  $v$  (or the rate at which  $v$  uploads  $k$  to  $u$ ).

In multilateral exchange, the system globally maintains one price per node. Let  $p_v > 0$  be the price of node  $v$ . Then the balance is a single exchange constraint,

$$\sum_{v \in N_u} p_v \sum_{k \in R_u^+} a_{vk}(u) = p_u \sum_{v \in N_u} \sum_{k \in R_u^-} b_{vk}(u).$$

The model of one price per resource  $p_k > 0$  defines the exchange constraint

$$\sum_{v \in N_u} \sum_{k \in R_u^+} a_{vk}(u) p_k \leq \sum_{v \in N_u} \sum_{k \in R_u^-} b_{vk}(u) p_k,$$

which provides an aggregated view.

These game-theoretic models are focused on characterizing global equilibrium qualitatively, e.g., the existence of a point  $(a, b, p)$  such that maximizes the utility of each node  $u$  subject to its exchange balance constraints. Their runtime application requires a system-wide mechanism to collect model input parameters  $a$  and  $b$ , evaluate prices  $p$ , and distribute results among participants.

2) *PageRank and EigenTrust*: As in the web information retrieval [31], structural properties of the P2P network topology are an essential factor for ranking. Such known graph-based algorithms as PageRank and EigenTrust can be used for computing rank values if the global topology is known. They are an uttermost form of indirect reciprocity: all paths exited between nodes are analyzed. Let us consider the mathematical models behind these algorithms.

The basic idea of PageRank is that a link from a node to another states an endorsement of the latter node, indicating its quality. Analyzing the global link structure one can rank nodes according to their perceived quality. Consider the following random walk in a graph of  $N$  nodes. At each step either with probability  $\alpha$  a node  $u$  selects a link  $u \rightarrow v$  to follow uniformly among available  $v \in T_u$  or with  $1 - \alpha$  jumps to a random node among  $N - 1$ . Jumping prevents permanent confinement in a strongly connected component of the graph. The parameter  $\alpha$  is the damping factor—the probability of following the link structure. The probability the random walk is in node  $u$  is the PageRank value

$$p_u = \alpha \sum_{v: u \in N_v} \frac{p_v}{N_v} + \frac{1 - \alpha}{N - 1}. \quad (30)$$

The sum is over all  $v$  such that  $u \in N_v$ . If the walk is at some  $v$  then  $1/N_v$  is the uniform probability of selecting  $v \rightarrow u$  and thus the walk returns to  $u$ .

The rank  $p_u$  is divided among  $u$ 's forward links  $u \rightarrow v$  for  $v \in N_u$  evenly to contribute to the ranks  $p_v$  of the nodes they point to. A node  $u$  has high  $p_u$  if the sum of the ranks of its ingoing links  $v \rightarrow u$  is high. Either  $u$  has many ingoing links or  $u$  has a few highly ranked ingoing links.

Instead a random jump in (30), a personalization vector  $\pi$  of damping factors can be used. Each  $\pi_u \geq 0$  represents the likelihood of jumping to  $u$ . This modification is personalized PageRank. Another non-uniform modification is weighted PageRank when forward links have probabilities relative to the link weights. Let  $b_{uv} > 0$  be the weight of a link  $u \rightarrow v$  and  $p_{uv}$  be the probability of selection of  $u \rightarrow v$ , then

$$p_{uv} = b_{uv} / \sum_{w \in N_u} b_{uw}.$$

PageRank with these two modifications can be computed iteratively starting from some initial values  $p_u^{(0)}$ :

$$p_u^{(i+1)} = \alpha \sum_{v: u \in N_v} p_{vu} p_v^{(i)} + (1 - \alpha) \pi_u. \quad (31)$$

A link  $u \rightarrow v$  of P2P topology can also be considered as indication of the quality of  $v$  for  $u$ . PageRank computed over the global topology graph has the following interpretation. If  $p_u$  is high then  $u$  is a neighbor of many other nodes or a neighbor of a few highly ranked nodes. This interpretation is useful for route and neighbor selection problems; each node  $u$  prefers those neighbors that have high rank.

EigenTrust [32] is designed for reputation management. It computes a trust score (rank  $p_v$  or reputation value) that indicates how likely a node  $v \in N$  is to be malicious. The system evolution is divided into rounds  $i = 1, 2, \dots$  in which nodes interact by making queries and consuming resources. At the end of round  $i$ , the record of correct and incorrect consumptions is used to calculate the trust values  $p_u(i + 1)$ .

A node  $u$  rates each transaction with its neighbor  $v \in N_u$ . In original EigenTrust, a transaction is rated as positive (1) or negative (−1), which is a common way in rate-based schemes [48]. Then a local trust value  $d_{uv}$  is defined, e.g., as the sum of the ratings of the individual transactions that node  $u$  has performed from node  $v$ . Original EigenTrust computes

$d_{uv}$  as the difference between the sum of positive transactions and the sum of negative transactions. The local trust values are normalized; let  $D$  be the normalized matrix.

EigenTrust aggregates local trust values using the notion of transitive trust: a node  $u$  will have a high opinion of those nodes who have provided it resources well. A node  $u$  is likely to trust the opinions of those nodes, since nodes who are generous about the resources they provide are also likely to be correct in reporting their local trust values.

EigenTrust global trust value is defined iteratively,

$$p_u^{(j+1)} = (1 - \epsilon) \sum_{v \in N} D_{vu} p_v^{(j)} + \epsilon t_u, \quad (32)$$

where  $0 \leq \epsilon < 1$  is a constant and  $t = (t_v)_{v \in N}$  is a probability distribution over pre-trusted nodes. Pre-trusted nodes are essential, as they guarantee convergence and break up malicious collectives. In the simplest case,  $t_u = 1/N$ . The iterations  $j = 0, 1, \dots$  start with some initial vector  $p^{(0)}$  and stop when  $|p_u^{(j+1)} - p_u^{(j)}|$  is made small enough.

The definition is similar to PageRank iterations (31) with damping factor  $\alpha = 1 - \epsilon$  and personalization vector  $\pi = t$ . The same interpretation is applicable. If a random surfer was searching for reputable nodes it can crawl the network using the following rule: at each node  $u$ , it will crawl to node  $v$  with probability  $D_{uv}$ . After crawling for a while in this manner, the surfer is more likely to be at reputable nodes than un reputable ones. The stationary distribution of the Markov chain is the global trust (reputation) vector  $p = (p_u)_{u \in N}$ .

## VI. CONCLUSION

The article proposed to consider mathematical models for large-scale P2P systems from two extreme views: local and global. The former is modest in terms of amount and quality of input information. The latter utilizes the overall knowledge about the system. The presented selection of models showed that both classes are important for P2P systems. Effective solutions for the large scale case are possible when the implementation takes into account what input information can be achieved, where it is processed using given models, and how the result is made available for nodes making control decisions.

## ACKNOWLEDGMENT

The author would like to thank Prof. Andrei Gurtov for valuable discussion and comments on the applicability and implementation issues of the considered models.

## REFERENCES

- [1] Q. H. Vu, M. Lupu, and B. C. Ooi, *Peer-to-Peer Computing: Principles and Applications*. Springer, 2010.
- [2] J. F. Buford, H. Yu, and E. K. Lua, *P2P Networking and Applications*. Elsevier, 2009.
- [3] Y.-K. R. Kwok, *Peer-to-Peer Computing: Applications, Architecture, Protocols, and Challenges*, ser. Chapman and Hall/CRC Computational Science. CRC Press, Taylor and Francis Group, 2011.
- [4] D. Korzun and A. Gurtov, *Structured Peer-to-Peer Systems: Fundamentals of Hierarchical Organization, Routing, Scaling, and Security*. Springer, 2013.

- [5] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335–371, Dec. 2004.
- [6] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica, "The impact of DHT routing geometry on resilience and proximity," in *Proc. of ACM SIGCOMM'03*. ACM Press, Aug. 2003, pp. 381–394.
- [7] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.
- [8] J. Risson and T. Moors, "Survey of research towards robust peer-to-peer networks: search methods," *Computer Networks*, vol. 50, no. 17, pp. 3485–3521, 2006.
- [9] D. Korzun and A. Gurtov, "Survey on hierarchical routing schemes in "flat" distributed hash tables," *Peer-to-Peer Networking and Applications*, vol. 4, pp. 346–375, 2011.
- [10] S. B. Akers and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Trans. Comput.*, vol. 38, no. 4, pp. 555–566, 1989.
- [11] E. Upfal and A. Wigderson, "How to share memory in a distributed system," *J. ACM*, vol. 34, no. 1, pp. 116–127, Jan. 1987.
- [12] L. Kleinrock and F. Kamoun, "Hierarchical routing for large networks: Performance evaluation and optimization," *Computer Networks*, vol. 1, pp. 155–174, 1977.
- [13] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up data in P2P systems," *Communications of ACM*, vol. 46, no. 2, pp. 43–48, 2003.
- [14] J. Xu, A. Kumar, and X. Yu, "On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 151–163, Jan. 2004.
- [15] J. Li, J. Stribling, R. Morris, M. F. Kaashoek, and T. M. Gil, "A performance vs. cost framework for evaluating DHT design tradeoffs under churn," in *Proc. of IEEE INFOCOM'05*, vol. 1. IEEE, Mar. 2005, pp. 225–236.
- [16] S. Ratnasamy, P. F. M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *Proc. of ACM SIGCOMM'01*. ACM Press, Aug. 2001, pp. 161–172.
- [17] G. S. Manku, M. Bawa, and P. Raghavan, "Symphony: distributed hashing in a small world," in *Proc. 4th USENIX Symp. on Internet Technologies and Systems (USITS'03)*. USENIX Association, 2003, pp. 127–140.
- [18] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," *IEEE/ACM Trans. on Networking*, vol. 11, no. 1, pp. 17–32, 2003.
- [19] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," in *Middleware'01: Proc. of IFIP/ACM Int'l Conf. on Distributed Systems Platforms*, ser. Lecture Notes in Computer Science, vol. 2218. Springer-Verlag, Nov. 2001, pp. 329–350.
- [20] C. G. Plaxton, R. Rajaraman, and A. W. Richa, "Accessing nearby copies of replicated objects in a distributed environment," in *Proc. 9th Annual Symp. on Parallel Algorithms and Architectures (SPAA '97)*, Jun. 1997, pp. 311–320.
- [21] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41–53, Jan. 2004.
- [22] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *Proc. 5th ACM Conf. Electronic commerce (EC'04)*. ACM, 2004, pp. 102–111.
- [23] D. S. Menasché, L. Massoulié, and D. Towsley, "Reciprocity and barter in peer-to-peer systems," in *Proc. of IEEE INFOCOM'10*. IEEE, Apr. 2010, pp. 1505–1513.
- [24] K. G. Anagnostakis and M. B. Greenwald, "Exchange-based incentive mechanisms for peer-to-peer file sharing," in *Proc. 24th Int'l Conf. on Distributed Computing Systems (ICDCS'04)*, ser. ICDCS '04. IEEE Computer Society, 2004, pp. 524–533.
- [25] C. Aperijs, M. J. Freedman, and R. Johari, "Bilateral and multilateral exchanges for peer-assisted content distribution," *IEEE/ACM Trans. on Networking*, vol. 19, pp. 1290–1303, Mar. 2011.
- [26] X. Jin and S.-H. G. Chan, "Reputation estimation and query in peer-to-peer networks," *IEEE Comm. Mag.*, vol. 48, pp. 122–127, Apr. 2010.
- [27] S. Buchegger, J. Mundinger, and J.-Y. L. Boudec, "Reputation systems for self-organized networks," *IEEE Technology and Society Magazine*, vol. 27, pp. 41–47, 2008.
- [28] B. Cohen, "Incentives build robustness in BitTorrent," in *Proc. 1st Workshop on Economics of Peer-to-Peer Systems*, 2003.
- [29] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience," *IEEE/ACM Trans. on Networking*, vol. 13, no. 5, pp. 1107–1120, 2005.
- [30] J. M. Kleinberg, "Complex networks and decentralized search algorithms," in *Proc. Int'l Congress of Mathematicians (ICM 2006)*. European Mathematical Society, 2006.
- [31] A. N. Langville and C. D. Meyer, "A survey of eigenvector methods for web information retrieval," *SIAM Rev.*, vol. 47, no. 1, pp. 135–161, 2005.
- [32] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in p2p networks," in *Proc. 12th Int'l Conf. World Wide Web (WWW '03)*. ACM, 2003, pp. 640–651.
- [33] K. K. Ramachandran and B. Sikdar, "A queuing model for evaluating the transfer latency of peer-to-peer systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, pp. 367–378, March 2010.
- [34] X. Yang and G. de Veciana, "Performance of peer-to-peer networks: service capacity and role of resource sharing policies," *Perform. Eval.*, vol. 63, pp. 175–194, March 2006.
- [35] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 367–378, Aug. 2004.
- [36] F. Paganini and A. Ferragut, "PDE models for population and residual work applied to peer-to-peer networks," in *Proc. 46th Annual Conf. on Information Sciences and Systems (CISS 2012)*, 2012, pp. 1–6.
- [37] W. Wang and B. Li, "To play or to control: a game-based control-theoretic approach to peer-to-peer incentive engineering," in *Proc. 11th Int'l Conf. Quality of service (IWQoS'03)*. Springer-Verlag, 2003, pp. 174–192.
- [38] L. Mekouar, Y. Iraqi, and R. Boutaba, "A contribution-based service differentiation scheme for peer-to-peer systems," *Peer-to-Peer Networking and Applications*, vol. 2, pp. 146–163, 2009.
- [39] B. Q. Zhao, J. C. S. Lui, and D.-M. Chiu, "Analysis of adaptive incentive protocols for P2P networks," in *Proc. of IEEE INFOCOM'09*. IEEE, Apr. 2009, pp. 325–333.
- [40] R. T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau, "Incentive and service differentiation in P2P networks: a game theoretic approach," *IEEE/ACM Trans. on Networking*, vol. 14, no. 5, pp. 978–991, 2006.
- [41] F. Wu and L. Zhang, "Proportional response dynamics leads to market equilibrium," in *STOC '07: Proc. of 29th annual ACM symposium on Theory of computing*. ACM, 2007, pp. 354–363.
- [42] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, "BitTorrent is an auction: analyzing and improving BitTorrent's incentives," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 243–254, August 2008.
- [43] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and improving a BitTorrent network's performance mechanisms," in *Proc. of IEEE INFOCOM'06*. IEEE, Apr. 2006, pp. 2884–2895.
- [44] S. Jun and M. Ahamad, "Incentives in BitTorrent induce free riding," in *Proc. 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, ser. P2PECON '05. ACM, 2005, pp. 116–121.
- [45] W.-C. Liao, F. Papadopoulos, and K. Psounis, "Performance analysis of BitTorrent-like systems with heterogeneous users," *Perform. Eval.*, vol. 64, pp. 876–891, October 2007.
- [46] A. Sherman, J. Nieh, and C. Stein, "FairTorrent: bringing fairness to peer-to-peer systems," in *CoNEXT '09: Proc. of the 5th Int. Conf. on Emerging networking experiments and technologies*. ACM, 2009, pp. 133–144.
- [47] I. Martinez-Yelmo, A. Bikfalvi, C. Guerrero, R. C. Rumin, and A. Maute, "Enabling global multimedia distributed services based on hierarchical DHT overlay networks," *Int'l J. Internet Protocol Technology (IJIPT)*, vol. 3, no. 4, pp. 234–244, 2008.
- [48] R. Zhang and A. Gurtov, "Collaborative reputation-based voice spam filtering," in *Proc. the 2009 20th International Workshop on Database and Expert Systems Application (DEXA '09)*. IEEE Computer Society, 2009, pp. 33–37.