IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Utility Patent Application (Provisional)

DIRECTED SOCIAL NETWORK

SPECIFICATION

CROSS-REFERENCE TO RELATED APPLICATIONS

Not Applicable

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not Applicable

REFERENCE TO SEQUENCE LISTING, A TABLE, OR A COMPUTER PROGRAM LISTING COMPACT DISK APPENDIX

Not Applicable

BACKGROUND OF THE INVENTION

1. Field of the invention

This invention relates to social networking technology, real and virtual peer-to-peer networks and multi-agent system. It includes a coherent peer-2-peer (p2p) network and methods and systems for creating communities within that network and instant groups for collaborative activities. Multiple synchronized communities may exist at multiple levels within a common environment, Group Spaces.

Social networks make it possible to be in contact with a large number of people at one time. This can be useful, as it provides a large pool of people with common interests. An objective may be to rapidly assemble a group for some immediate need without polling everyone. Currently subscription to a group or community may result in a large number of messages of low interest to the individual. What is needed is focus: the individual only wants to see the messages that relate directly to him or herself.

2. Description of Related Art

A network is a digital environment comprised of nodes in which information is exchanged over links among the nodes. The network may have commonly hosted nodes. Distributed individuals in various communities (i.e., communities of interest) may benefit from being able to rapidly form groups. These groups may be:

1) task or issue focused;
2) transient;
3) subsets of some larger community;
4) focused temporary social networks

Such groups will not necessarily have any persistence, and will require minimal effort to establish them on the part of the participants. Typically the participants in such a group are comprised of members of a defined community. Communities may have many areas of interest; the group should be comprised of only those who have a specific interest in the issue or situation at hand. For example, if a salesman is meeting with clients and has a specific question to respond to he may request assistance from within his organization. He must formulate the issue and then request that people who can discuss the issue in a meaningful way will join an online group, e.g., Web meeting. The group must:

1) Form rapidly
2) Have a well-defined focus
3) Restrict membership to relevant participants
4) Be able to communicate effectively
5) Have an appropriate level of security; and
6) Disband when finish.

Such a group is focused, instant and transient.

Current technologies such as instant messaging (Imm Twitter), realy simple syndication (RSS), email,and social networks (LinkedIn, Facebook,) may result in invitations that are disseminated to all members of the community. In doing so, everyone in the community is potentially subject to receiving all invitations, overwhelming them with chatter. Potentially anyone in the community may issue an invitation. Potentially anyone in the community may receive an invitation. It is desirable to restrict invitations only to those for whom the group is relevant. hosted groups, Web meetings and similar technologies have either targeted lists or prearranged meetings. Distributing invitations and the process of joining groups must happen rapidly under the individual's control.

A peer-to-peer (p2p) network is a system of nodes in which nodes communicate with each other with no intermediary. Each node acts autonomously, distributing operations throughout the network. This system allows any node to contact any other node in a system, either directly or through relays. Such a system is sometimes used to transfer files between any two nodes, e.g., BitTorrent. Frequently the system will bring the two nodes into direct communication with each other. Message splitting techniques may also be used to take advantage of the potential multiple paths between source and target. Such systems can overload servers. They often do not have high security. Very large communities may experience significant time delays between points, in part dependent upon network transit time.

There has been active research in the use of synchronous networks for p2p applications built on the work of Kuramoto (1975, Acebrón et al. 2005, Kawamura et al. 2008, Wagemakers et al. 2007; see Arenas et al. 2008 for review). Synchronized networks formed by coupled nodes have properties that are not the simple sum of the nodes' behaviors. Even though the coupling among nodes may be quite limited in number relative to the total number of nodes, and if the coupling moderately weak, the entire network can behave as a unified structure in which events a transferred extremely rapidly throughout the network (e.g., Tanaka et al.2002). Further, such networks offer multiple opportunities to provide security.

Kuramoto Model

Kuramoto's model describes the global synchronization of arrays of locally coupled oscillators. Although this invention includes other topologies, the flat locally coupled matrix provides demonstrates essential ideas. To illustrate the concept, a flat *array* of oscillators can be considered, in this case, magnetic domains (Figure 1). A topology is a geometric structure or space that has properties (e.g., adjacency) that are not affected by changes such as stretching or bending. A topological structure may be defined as containing a *set*. This array of magnetic domains has a topology of a flat sheet in which each element in the set is linked only to its nearest neighbor with no time delay. The links define coupling among the elements, in this case magnets that can rotate freely with an intrinsic frequency but cannot translate (move laterally, changing coupling relationships). Each magnet, M_i, is *bi-directionally coupled* to its nearby neighbors, M_j, through their magnetic fields. Initially all magnets have random orientations and are rotating with their own intrinsic frequency, ω_i. The intrinsic frequency is a characteristic of the node. In this case, the mean angular velocity, ω, for all magnets can be set to 0. By definition the *rate of change* of an angle θ is:

$$\dot{\theta} = \frac{d\theta}{dt}$$

and

$$\omega_i = \frac{\dot{\theta}}{2\pi}$$

ω_i is measured in radians per unit time.

M_i is coupled to k local magnets. Each of these is designated M_j for which:

$j = 1..k$.

According to the Kuramoto model (1975, Kawamura et al. 2008), the coupling effect of the k Mj magnets on the rate of change of angle θi of Mi is:

$$\dot{\theta}_i = \quad \omega_i + \xi \sum_{j}^{kj} \sin(\theta_j - \theta_i) + f(t)$$

for which λ is the *coupling coefficient* between M_i and its k neighbors. $f(t)$ is a driving function of a bias or perturbation. We shall ignore it.

The phase difference between M_i and M_j is:

$$\varphi_{ij} = \theta_j - \theta_i$$

The Kuramoto model describes how the system attempts to reduce this error, φ_{ij} . If φ_{ji} is positive then θ_i *lags* behind θ_j:

$$\theta_j - \theta_i > 0$$

$\dot{\theta}_i$ *increases* to reduce the error. If

$$\dot{\theta}_i \neq \dot{\theta}_j$$

then φ_{ij} is constantly changing. When

$$\sum_j^{kj} \sin \varphi_{ij} = \sum_i^{ki} \sin \varphi_{ji}$$

then

$$\theta_j - \theta_i = 0$$

and

$$\dot{\theta}_i = \dot{\theta}_j \, .$$

Thus:

$$\varphi_{ij} = \varphi_{ji} = 0$$

The system is *stable*. All magnets rotate at the same frequency and are commonly aligned. The rotation of the agents is both *coherent* and *synchronous*. The combined phase differences, φ_{ij}, drive $\dot{\theta}_i$ toward a common ω. The intrinsic frequency of M_i, ω_i, is not necessarily the same for all magnets. If the coupling, ξ_{ij}, is strong enough to overcome ω_i then it will drive $\dot{\theta}_i$ to a common ω as long as ω_i is not too different from ω_i. ω is a result of the interaction of all of the magnets, M. Thus ω is "negotiated" (converges) on a final value. The stronger the coupling, λ, the greater the synchronizing drive. If λ increases to a *critical value*, ξ_c, all of the magnets in the array are aligned, $\varphi_{ij} = 0$ (Figure 2). The more coupling among magnets, by strength, ξ_{ij}, and number, k, the stronger the array's synchronicity and the faster it is achieved. If the initial mean ω is 0, then all of the magnets are *stationary* and *aligned*:

$$\dot{\theta}_i = 0 \, .$$

A perturbation of a magnet is a change in its phase angle, $\varphi_{if} \neq 0$, with respect to the array's phase angle, θ_f. At this point, the array is still coherent, but has some local loss of synchronicity. This creates a coupling drive among the local connections. Any perturbation of any of the magnets affects the entire array resulting in phase shift propagation through the array. This occurs very rapidly even though all coupling is only local. There is no overriding force that keeps the magnets aligned, only the coherence of the overall array, driven by the coupling of the oscillators. To restate: The local coupling creates global coherence. Any local change is rapidly communicated throughout the entire array. Under these conditions the coherence path length, λ_c, (Figure 3, 3001) approaches infinity, namely the size of the array.

Magnetizing a ferromagnetic material is an illustrative case. Such a material is made up of little magnets or ferromagnetic ions. If the temperature is above some critical value, the randomness of θ and ω created by the thermal noise overcomes the coupling, so the material is not coherent. As the temperature drops, a critical temperature is reached where all of the ferromagnetic ions align in one direction. If there is an external magnetic field, they align along that field. The critical temperature is known as the Curie temperature.

The model of coupled oscillators has been applied in many areas, including networks. In the most basic planar, matrix array, each oscillator is a node (Figure 4) linked ($L_{i..k}$ 4001) to nearby nodes with some coefficient, ξ_{ij} (4002). Although network nodes don't oscillate per se, they do have cycle or transfer times associated with them. The same principles apply.

How do coherence and synchronicity apply to communications nodes, $N_1..N_m$? An example is an array of sub-antennas that are combined into a single large antenna. In order for this array to function as a single antenna, or field, the relationship among all of the component sub antennas must be stable and known. This stability and knowledge comprise coherence. Coherence is frequently associated with fields. In this sense, a field is a nearly continuous set of small array elements. The coherent, synchronous behavior of the whole can be described as emergent or self-generating. Moussaid et al. (2009) have applied the Kuramoto model to the behavior of swarms, flocks and crowds. The behavior of a single bird does not describe the behavior of the flock.

Scale-free network

A randomly connected network such as a p2p network as illustrated in Figure 5 does not have meaningful spatial dimensions. It is a dimensionless space from a spatial standpoint. Spatial descriptions in a network are functionally irrelevant. The concept of spatially "local" is not applicable to an open network, such as the Web. The Web is scale-free. Instead, topologies of connections are used. Networks are often described as linear graphs of and vertices and edges. In our case, the vertices are nodes and the edges are links. This document will not explain linear graphs. The number of nodes, *k*, that any single node is connected to and their connections describes the space. A scale-free network is a complete extension of the limited randomly linked small world synchronous network, in its more general form, the Watts-Strogatz (W-S) model (Arenas 2008).

The addition of random non-local connections increases both the coherence of the array and the speed with which synchronization occurs (Figure 5, $n = 25$, $k = 4$). An important aspect of the behavior of a scale-free system with random connections is that under the right conditions only 4 to 10 connections per node can synchronize 10,000 nodes (Arenas et al. 2008). The mean number of hops, ℓ, for a message to travel between any two points in a network in which $k = 10$ and $n = 10{,}000$ is 4. Message communications among nodes is nearly optimal. For a synchronized network, as the size of a network of nodes (n) becomes much greater than the number of links per node, *k*, the network can be treated as a space. With proper controls, the action of a single node or link will have effects on the entire network.

This view requires that one consider the network to (initially) have essentially uniform (homogeneous) characteristics throughout.

Time delays in synchronized networks

As network delays are introduced such systems have limited success, becoming problematic. If the value of the delays is constrained in, coherence and synchronization become possible (Fischer et al. 2006, Zhou & Roy 2007, Triplett et al. 2006, Tanaka et al. 2002). Equalizing time delays is important in the creation of a synchronized p2p network. Attempts to compensate for unequal time delays by using weighting functions are cumbersome and have only limited success.

$$\dot{\theta}_i(t) = \omega_i + \sum_j^{kj} \xi_{ij} \sin\left(\theta_j(t-\tau) - \theta_i(t)\right) + f(t).$$

Clusters

A subgroup, or cluster, within a set of nodes may be created through several means as described in the review by Arenas et al. (2008). Members of a cluster are more closely associated pr coupled with each other have a higher coefficient of connectedness than would be predicted from a random set of connections. The connectedness coefficient, C, is essentially the fraction of actual link triangles (three vertices forming a loop, edges with common vertices) over the possible triangles. A cluster may also be formed when some or all of the members are connected to a common hub node. Clusters have more direct paths among nodes on average than the main set does. Thus the number and pattern of connections of subgroups of nodes may create clusters within the main set. Individual cluster formations may also occur as a community of nodes becomes synchronized. Clusters typically have faster communications times among its members than in the larger community the cluster is a subset of.

SUMMARY OF THE INVENTION

As the number of people and services on the web increases, the size of social networks increase and the number of communities rises, the memberships in each community increases, messaging increases and the number of groups increase, centralized management of such a system can become difficult, resulting in poor system behavior for the participants. There is a need for systems and methods providing for the combination of increased numbers while providing more focused experiences for participants. Synchronized complex networks can fill such needs. Ensembles of communities, clusters and groups can benefit from an environment that may be synchronized for some or all ensembles. Entire communities may be synchronized. Groups that form as a subset of the community are separated from the community or cluster. They may be synchronized. Synchronized groups may form as subsets of unsynchronized communities. A cluster is an association of ports with a higher than random coefficient of connectedness. A cluster has faster message transfer times than the community it is part of but is not separate from the community. A group may derive as a subset of a community or a group. This description depicts synchronized communities, clusters and groups although communities and clusters may be un-synchronized or partially-synchronized.

Community members should be able to remain anonymous, within the rules set up by the community. A community member may receive an invitation message and choose to ignore it, remaining anonymous and invisible. A member may respond to an invitation while the inviter remains anonymous. A participant may choose to become a member in a group while continuing to receive new invitations.

Individual group members may change their availability status at any time. They may change their interest profiles that describe their appropriateness. They may be intermittently (sporadically) connected but what to have potential invitations stored for later review. Thus the needs of each individual community member may change at any time. Further, new members may join the community and current members leave. As communities grow larger, the number of changes and messages increases. A system may have multiple communities; some participants may be members in more than one community. Each community may have different rules. Each community may have different taxonomies.

While the invention is described herein by way of example for several embodiments and illustrative drawings, those skilled in the art will recognize that the invention is not limited to the embodiments or drawings described. It should be understood, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as described herein. The headings used herein are for organizational purposes only and are not meant to be used to limit the scope or description or any forthcoming claims. As used throughout this application, the word “may” is used in a permissive sense (i.e., having the potential to), rather than the mandatory sense (i.e., meaning “must”). Similarly the words “include”, “including”, and “includes” mean including, but not limited to. Furthermore, each of the embodiments may be considered both alone and in any combination.

Embodiments of a system and method for providing communities from which groups of participants can form rapidly in an interconnected or networked environment are described. The term “participant” may include both human and non-human (e.g., programmatic) entities. Methods for the rapid creation of groups that become isolated from the main community are described. The underlying system and methods create effective mechanisms for the rapid communication of information in a real or virtual peer-to-peer environment are described. This environment embodies an infrastructure that optimizes the communication between any two points within the community. The system and methods described allow multiple communities and their groups to exist within the same environment without interference or interactions among them. Methods and systems supporting the use of different user interfaces and platforms are described. Systems and methods for the generation of groups focused around specific interests, issues or needs are realized. Systems and methods limit the messages, such as invitations to join groups, to those of specific interest to participants are described. Methods and a system that supports security of the system and participants are described.

In an embodiment of the invention the peer-to-peer (p2p) network is comprised of ports that all may be hosted within a common system, the host. The host may span multiple platforms. This constitutes a virtual p2p. Other configurations can be used as equivalent to this including mixed systems of both hosted and remote ports and cloud environments. The commonly hosted model will be used as the general case for the various embodiments; it can provide the fastest network synchronization and message exchange times. In an embodiment each port may created by a software agent. This agent is typically—but not exclusively—a component in a multi-agent system. Each agent is described as “intelligent” as it is able to function autonomously following a set of rules. In an embodiment each port may be provided as an intelligent agent.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1. Array of spining magnets

Figure 2. Synchronously aligned magnets.

DETAILED DESCRIPTION

Major components, methods and systems can be used to describe the formation and operation of A group-spaces environment and its contents.

Environment

Group-Spaces provides a digital environment of interconnected ports in which ensembles can form, exist and dissolve with little central computer control. The environment features a virtual peer-to-peer (p2p) network that can be synchronized through a small number of connections among ports. Ensembles include communities [Figure 6, 6001], clusters [6002] and groups [6003, 6004]. The environment created by the ports automatically optimizes, balances load and provides rapid communication between any two ports, irrespective of their location in the network. Further, communications propagate to all ports in the synchronized ensemble rapidly but can be ignored easily.

Synchronous complex networks

Membership in a synchronized ensemble is constrained to those participants with synchronous ports. Participants may be human and non-human (e.g., programmatic, services, data sources, sensors, web cams). The system can self-manage. Small groups, even those derived from larger communities, may have increased benefits of speed, robustness of messaging and security. Small groups may have access to enriched interactive experiences such as collaborative work spaces, activity spaces, simulations and games. The participant's interface can be aligned with the participant's platform. A synchronous complex p2p network tightly binds members of a community and even more tightly binds participants in groups from within that community for a smooth, rapidly responsive, highly focused experience while minimizing excessive messages for participants to process.

A synchronized network is comprised of a population of logical nodes. In this description, the nodes are ports. These ports can be realized as intelligent agents in a multi-agent system (e.g., Tryllian, Ascape, Cougaar, CybelePro, JAS, Madkit, MAGSY, Moduleco and Swarm). The behavior of the ports couples them into a network. Under the proper conditions this network can become synchronized. A synchronized network has all ports operating with a common frequency with small, transient variations. This means that there is little or no phase lag in events across the network within a time interval resulting from the cumulative delay times between two ports separated by hops (see below). It does not imply that the ports are actually oscillating.

In an embodiment the members of an ensemble are members of a synchronous peer-to-peer (p2p) network. A system may have multiple types of ensembles with different nestings. Synchronization does need to be true of all types of ensembles in a system. The greater the degree of synchronization of each of the ensembles, the more responsive the system will be to the user. A synchronous complex p2p tightly binds the participants into a community. It is like a bell—no matter where it is struck, the entire bell rings. Multiple independent ensembles can co-exist within the same environment. Information is disseminated rapidly within a synchronous complex network. Information is balanced from all sources to all recipients: there are no inherently favored sources.

In the Group-Spaces environment each port is coupled to a small number of other ports yet an overall synchronization of the entire network emerges. As described in the existing art (above), the synchronization of a network can be expressed by its coherence path length, the greatest distance over which separated events maintain a meaningful relationship. In a synchronized network the coherence path length approaches infinity with only limited coupling over small numbers of coupling links among ports. The properties of a synchronized network emerge from the properties of the ports and their links.

The messages that pass over the synchronized network result in the user's experience. Messages can be tightly focused by both the originator and the recipient. Tightly focused groups can be formed and dissolved rapidly on an as-needed basis.

Initially a single community of ports within a system will be described. This set of ports may create a synchronous (i.e.;., coherent) real or virtual p2p environment. The system can be realized in a multi-agent system, a distributed network or a mixture of both. A system can be hosted within a set of one or more connected computers as servers or equivalent platforms for supporting agent based systems. A system of ports that are hosted entirely within such set constitute a virtual p2p network. This summary description will use such a virtual p2p, but the concepts are not limited to it. Such a system may be incorporated into, incorporate or be a companion to a community of interest as conventionally defined.

Ports

In an embodiment the behaviors of a set of ports in a real or virtual p2p environment create a community comprised of synchronous, or coherent, sparsely linked ports. This is analogous to a flock of birds that acts as a unit, each bird coupled only to nearby birds, yet a coherent flock behavior emerges.

A node represents a participant's presence in the system. The port is a participant's connection into a specific current or potential ensemble (e.g., community, cluster or group). A node contains none or more ports. A port has several components that may be realized separately or be integrated into a single agent. The separation here is functional, illustrating the system and methods. A port agent might be termed an "intelligent agent." Conceivably ports could be realized in separate small processors or software objects. Each port [Figure 7] has a connector [J_i,7001] that exchanges data blocks with other connectors [J_1 … J_k] over logical links [$L_{i,j}$, 7002, 7003, 7004]. The input [7005] and output [7006] tracts pass and receive data blocks from the connector to a port processor [7007] function. The port processor has associated memory [7008]. The port processor may pass data blocks onto the interpreter [7009], potentially using data queues [Buffer Qs 7010, 7011]. The interpreter processes the data blocks it receives. It may derive information that is subsequently passed through a buffer Q [7012, 7013] to the interface. The interface interacts with the participant's platform to create the desired user interaction. The user may take some action. This results in data that are passed though a buffer Q, if used, to the interpreter that may perform some action on the data, sending through a buffer Q to the port processor. The port processor sends the data, transformed if necessary, as a block through a buffer Q to the connector J_i [7001]. Connector J_i subsequently sends the data block out over some or all of its links to other connectors.

Links

Ports communicate among themselves over links. A link has two ends, each at a port [Figure 8]. A link is unique for that connection. It may have different identifiers (IDs) for each end, e.g. $L_{i,j}$, $L_{j,i}$.

Synchronization

A data block is a parcel of digital information that is transferred among ports over links. Several types of data blocks may exist in the Group-Spaces. In this example two types of data block are illustrated: packet [Figure 9, 9001] and ping [9002]. Pings may maintain the synchronization of the ports [9004]through port timing management. As described by Triplett et al. (2006) in the background section above, if all of the time delays, τ_i, for the nodes in an ensemble are the same and all of the weighting values, ξ_{ij}, are the same it is possible to synchronize the complex p2p network. Triplett et al. (2006) provide an equation describing the coupling behavior for port P_i with time delays (τ) in communications among ports:

$$\dot{\theta}_i(t) = \omega_i + \sum_{j}^{kj} \xi_{ij} \sin\left(\theta_j(t-\tau) - \theta_i(t)\right) + f(t)\,.$$

ξ_{ij} is the weighting of the coupling between ports P_i and P_j. As described in the review by Arenas et al. (2008), differentially weighted couplings are problematic; this would be particularly true in a p2p with many participants. A single value, ξ, can be assumed, although conceptually not required. ξ can be placed in front of the summation. The ping must go through both the input and output buffer queues, with their subsequent delays. For synchronization to occur, delays between ports should be the same across all ports in a synchronized community. Pings may be exchanged between two ports to in a method for establishing the proper delay. It is not possible to accomplish this directly, as the port must be synchronized with all of its linked ports, J_1 .. J_k. Thus the process described below may encompass multiple pairs of the port and its linked ports. As illustrated in the profile in Figure 10, τ_{ij} [10001] is the time between when a ping is received at port P_i [10002] and when it is received at port P_j [10003]. The exchange between two ports, P_i and P_j, may follow a profile summarized in Figure 10. A delay of τ_{ij} has three components: a transfer delay, γ_{ij} [10004], which includes the buffers in one direction (e.g., outbound from P_i and inbound into P_j), a processor delay, δ_i [10005], and an error delay, ε_{ij}. Although it is possible to adjust the delay such that it matches each connection, for synchronization all τ_{ij}'s are the same, all δ_i for a port are the same no matter which port it is exchanging pings with and all ε_{ij}'s are the same (j = 1..*k*). As all ε_{ij}s are the same, all δ_{ij}s for a given P_i are the same, thus there is one δ_i. Individualized linked δ_{ij}s could be used, but with increased complexity.

Three methods and systems can be considered for synchronizing ports to form a coherent or synchronized community or group:

1. Negotiated τ
2. Controlled τ
3. Enforced isochronous synchronization

1. Negotiated τ

τ_{ij} [11001] is an intrinsic delay of a packet between ports P_i and P_j as illustrated in Figure 11. The time delay associated with the equation by Triplett et al. (2006) has identical delays for all nodes or ports. Not all port pairs may have the same intrinsic delays. It may not be possible to develop a precisely equal delay, τ_{ij}, between all ports. An approximation may be adequate. The following is an illustration of a system and method for developing and maintaining τ_{ij}. It may be one of several useful methods. An estimation technique is illustrated..

A value of τ_i for a port P_i may be developed through negotiations among the ports. Using this system and method there may be a real or virtual ping exchange as illustrated in Figure 10. When port P_i [10002] initiates a ping [10006] it starts a timer. When it receives a ping acknowledgement [10007] from port P_j [10003] it records the timer value. This is time β_{ij} [10008]. P_i returns a ping [10009] to P_j providing β_{ij} and its own known delay, δ_i [10005]. At the conclusion of this exchange both Ports P_i [10002] and P_j [10003] have the values β_i [10008], β_j[10010], δ_i [10005] and δ_j [10011]. Given τ_i , a port may calculate δ_i. It may not be possible to calculate the time delay τ_{ij} [10001] directly from those values. In this method, several values of τ are used, as defined below:

τ_i = intrinsic time of port P_i;

τ_j = intrinsic time of port P_j;

τ_{ij} = intrinsic time between port P_i and a pinged port, P_j, as currently calculated;

$\overline{\tau}_i$ = intrinsic time of port P_i as a moving average;

τ'_i = adjusted new value of intrinsic time of port P_i;

τ_c = intrinsic time of community.

τ_{ij} is the sum of the delay internal to the port or agent and the transit time from P_i to P_j, including buffers in one direction:

$$\tau_{ij} = \delta_i + \gamma_{ij};$$

similarly

$$\tau_{ji} = \delta_j + \gamma_{ji}.$$

From Figure 10:

$$\beta_i = \delta_j + \gamma_{ij} + \gamma_{ji}, \text{ and}$$

$$\beta_i = \delta_j + \gamma_{ij} + \gamma_{ji}.$$

$$(\tau_{ij} + \tau_{ji}) = (\gamma_{ij} + \gamma_{ji}) + (\delta_i + \delta_j).$$

Summing the two intrinsic times:

Combining and rearranging:

$$\beta_i = (\tau_{ij} + \tau_{ji}) - \delta_j;$$

$$\beta_j = (\tau_{ij} + \tau_{ji}) - \delta_i.$$

Summing:

$$\beta_i + \beta_j = 2(\tau_{ij} + \tau_{ji}) - (\delta_i + \delta_j)$$

For simplicity in notation the values of β and δ can be combined into single values:

$$Z_i = \beta_i + \delta_i,$$

$$Z_j = \beta_j + \delta_j.$$

Combining and simplifying:

$$\tau_{ij} + \tau_{ji} = \frac{1}{2}(Z_i + Z_j).$$

τ_{ji} cannot be known. An assumption can be made that:

$$\tau_{ji} = \tau_j.$$

τ_j is unknown, but can be assumed to be P_j's best estimate of τ_c. That, too, is unknown. However, P_i is also attempting to set τ_i to τ_c. Thus we can use the current value of τ_i for τ_j. This will be in error, but under synchronization of the network, the two nodes will be synchronized and will thus, on average, have close to the same τ. This can be used to calculate the current value of τ_{ij}:

$$\tau_{ij} = \frac{1}{2}(Z_i + Z_j) - \tau_i$$

The current value of τ_{ij} is being calculated, in part, using the previous value of τ_i. If only the value of previous value of τ_i is used to create the next value of τ_{ij}, τ_{ij} might not converge or produce a value that produces a stable or meaningful τ_c. τ_i can be calculated based on a moving average of τ_i that incorporates the estimated timing between P_i and several other ports. An error term can be calculated for the current τ_{ij}:

$$\varepsilon_{ij} = \tau_{ij} - \tau_i$$

Only a partial correction of τ_{ij} will be used to permit influence from other linked ports. A portion of ε_{ij} is applied to τ_{ij}. The resulting value is averaged with the values of τ_{ij} calculated for the previous *k* ports:

$$\overline{\tau}_i = \frac{1}{k}\sum(\tau_{ij} - \xi_i \varepsilon_{ij}).$$

Other weighted averaging methods may be used. $\overline{\tau}_i$ may drift. It will normally be desirable to move toward the shortest τ_i that still supports the port being synchronized with those it is linked to. A small bias, b_i, can be subtracted from $\overline{\tau}_i$ to produce the new, bias offset adjusted value:

$$\tau_i' = \overline{\tau}_i - b_i$$

This value will be used in the subsequent calculations of τ_i, iterating toward, or maintaining, a stable synchronized state.

A strict order for pinging each of the k links is not required. The frequency with which pings are negotiated may be modest for large fairly stable networks. The system free-running may generate an adequate number of pings as an emergent behavior. This may be a control, for example a port may not participate in a ping exchange any more frequently than some threshold timing, Δ. Thus, a port may reject a ping. The port agent's rule system will accommodate this. The port agent may have a rule of refreshing the oldest τ_i first.

A port that cannot maintain the current τ_c within the specified span will not be part of the community. Thus the network may be effectively partitioned into multiple communities, each independently synchronized. These communities are interpenetrated within the network. This separability or segregation of groups or communities is an important aspect of this invention. It is discussed more fully below. At least to must be actively synchronized for a community or group to exist. In some circumstances it may be useful to provide dummy ports that serve only to keep the community or group synchronized, being involved in the ping interchanges. Packets would simply be relayed.

A group-spaces network community may use the following constants, potentially provided by a community service:

k = number of links
k_{min} = minimum number of links
ξ = coupling coefficient

b = bias offset

σ = Allowable deviation

Δ = mean or minimum ping interval

See below for a listing of symbols and notaztions. This limited random coupling by all ports to a small (*k*) number of other ports can result in the synchronization of the community or network. A network using negotiated intrinsic times can produce one or more ensembles that self organize to emerge. They may self optimize and naturally foster the formation of groups. Group formation is discussed below. Reviews of analytical methods, properties and controls for synchronous complex networks are provided by Arenas et al. (2008) and Belykha et al. (2003).

2. Controlled τ

A central, ensemble (community, cluster or group) manager may provide a fixed, but adjustable, τ_e to be used by all ports that maintain synchronization. The manager function may monitor all, or representative port behaviors to adjust τ_e as appropriate. Each port must be able to use this value as the combination of the link time of the buffers and message transit, and the processor delay. The central system will select a value that accommodates all active ports that are to be synchronized. It may be that the reduced processor load of this enforced delay method will offset the time required to set the negotiated delay above.

3. Enforced isochronous synchronization

A central process may enforce synchronous behavior of the ports through the use of strobed processing. This is a saltatory model in which all processors output results at the same time with an interval determined by the central manager. All agents are forced to work in lock step. The manager process must determine the value of the strobe interval that will allow the appropriate number of ports to participate in the community. This will typically be a long strobe interval.

An intermediate level of synchronization control could be achieved through control by a small number of ports out of the total. A small number of control ports could negotiate timing amongst themselves and drive the remainder of the network through directed links. This has similarities with the hub model.

None or more communities may exist in the operating p2p. This combined environment for communities is defined as “Group Spaces” as illustrated in Figure 12 [12001]. In an embodiment a group-spaces community [12002, 12003] may be functionally separate from other group-spaces communities within the common Group Spaces with no interaction or interference among the communities or their groups.

Ports, nodes and ensembles

A participant may have a node. The node may contain none or more ports. The node may provide various resources to the participant. A node has at least one interface to the user’s platform It may not always have a port. A port may connect to an ensemble (community, cluster or group) either through the ensemble’s method or system or, as described herein, through synchronization. A node may combine multiple port interfaces. The user may manage his, her or its interface with ports through an interface, including client-side applications and plug-ins.

Ports

The port is key to the synchronized ensembles in Group-Spaces. Ports create the group-spaces and the communities through links among ports. In an embodiment each port has several components. These may be realized in equivalent systems and methods. A port may be realized as a software module, object or agent. In the following description, the agent model is used, but the system and methods are not limited to that embodiment. The components may be arranged logically in layers as in Figure 7. A connector, J_i [7001], connects to the group-spaces. This communicates with other connectors $J_1 \ldots J_k$ in group-spaces over links $L_{i,1}$ [7002], L_i, … [7003], $L_{i,k}$ [7004]. A port connector both sends and receives data blocks from other connectors over links [$L_{i,1} \ldots L_{i,k}$]. The connector communicates data blocks to and from a port processor [7007] ; buffer storage queues [7017, 7019] may be present in both paths. The port processor is responsible for the actions of the port relative to group-spaces. It manages the output queue to the connector. The port processor also communicates with an interpreter [7009]. The interpreter may be part of the function of the port processor or of the agent as a whole; it is separated here for clarity. The interpreter stores profile information about the participant and his/he/its platform requirements. Messages that are appropriate for the participant are transformed and forwarded to the participant's platform through an interface [7014].

A port does not necessarily have to be connected to a user interface. A participant does not need to be connected to a user interface as illustrated in Figure 13, there are several port states, among them are:

User linked to portal, port synched to ensemble, thus the user is synched to ensemble [13001];

User not linked to port, port synched to ensemble [13002];

User linked to port, port not synched (synchronized) to ensemble [13003];

Node with multiple ports and common interface [13004];

User connected to surrogate, surrogated linked to synched port [13005].

Other configurations may be used. P designates "port," U designates the user's interface at the node that communicates with the use's platform in suitable manner. The configurations illustrate some of the functions possible with group-spaces. A user may be connected through a user interface to a port that is not synched to the community synchronized network. The user may configure his, her or its profile, review messages stored in the I-Buffer queue and create messages to be broadcast through the port when it is synched with the community network. Similar scenarios may apply to synchronization to a group network.

A surrogate for the user may be used. The surrogate may provide connection services to other networks or platforms. It may respond to port messages under actions predetermined by the user. It may provide an interface between an application or plug-in on a user's platform and a communications service, system or method.

A dummy port is a port with no user interaction. It may exchange pings for synchronization. It may relay messages. As illustrated in Figure 14, a hub [14005] is a specialized dummy that may be used to increase connectedness [14004] among members of an ensemble to create a cluster of ports [14003].

A group has Active ports may also be interactive, involving the member. An active port that does not interact with the member may be termed "passive." A passive port participates in the synchronization and message relaying only.

Port rules

Ports, for instance as agents, follow a rule set. Rules may include:

A port may be a member of only one community or group at a time.

A node may contain more than one port.

Ports within a node may, but not must, share information.

A port may have a maximum number of links.

A port must have a minimum number of valid links, k_{min}. Exceptions may occur during initiation of or entry into an ensemble (community, cluster or group), hub, maintenance or dummy port functions.

Pings are responded to only over the link they were received from.

Pings are not relayed.

Packets are relayed to all links except the one over which the packet was received.

A packet that was received previously will not be relayed.

Packets are sent over those links that are valid, e.g., maintained through pings or other means that ensure synchronization.

Input queue order must be maintained, including packets and pings.

Output queue order must be maintained, including packets and pings.

Expired packets are not relayed.

Pings and acknowledgements (Figure 10 10006, 10007) may be issued Δ or more time apart.

Other possible rules include, but are not limited to:

If no successful ping has been executed over a link within a specified period of time, that link may no longer be considered valid. A suggested minimum time is $(2k+1)\Delta$.

Ports have unique IDs – if used at all.

Links have unique IDs.

A link ID may or may not persist.

Packets from an authorized source may modify constants, e.g., k, k_{min}, ξ, b, σ, Δ, default expiration time duration and packet structure.

A new link must provide its ping exchange values using a specific τ, possibly supplied by a gateway port.

A link with timing that does not fall within the specified coherence time (τ)span (τ_{max} - τ_{min}) during a specified number of pings is terminated.

Ensembles: communities, groups and clusters

A community may actually be isolated as members of a database with no ports. In this form it is an aggregate. A cluster may be comprised of members of such an isolated state aggregate. That cluster may be synchronized, each of the member unit's data structure used to initialize an agent (e.g., instantiate an agent object). Synchronization may be partial or non-existent.

Active ports comprise the synchronized group.

An ensemble is a population. Three types of ensembles are described: communities, groups and clusters. The relationships among these ensembles are illustrated in Figure 6. A community is a population of members with some common connection or interest that are associated. An active community may have a synchronized network with more than one active port contributing to its existence. Ports create a synchronized ensemble. A synchronized ensemble requires more than one port, as the synchronized ensemble emerges due to the behaviors of the ports. Members of a community may be synchronized with that community when they have one or more ports that are synchronized with the community network [6001]. A cluster [6002] is a subset within a community that has a higher coefficient of connectedness, C, than would be expected through random linking among ports. Clusters will have on average fewer hops between any two members than in the larger community. This facilities rapid group formation when most or all of the members of a group are from a cluster. A cluster may be useful when the larger community is unsynchronized or incompletely synchronized. Ports in a cluster maintain some links outside of the cluster. A group [6003, 6004]is a synchronized set of ports that are segregated [6005] from the community or cluster, although a cluster may become a group. The environmental management system may collocate clusters or groups within a common host structure such as one or more common servers.

A cluster may be an affinity cluster that is identified within an ensemble by the management system. An affinity cluster is a sub-population of members that frequently participate in groups in common. The system may more closely link these ports without creating a cluster known to the members. This can provide a faster experience for the members. It may allow optimization of the environment. Such a cluster may have greater connectedness, one or more hub ports or both.

An ensemble may be described at several levels:

Aggregate. The total population of members of an ensemble is its aggregate. Individual members may or may not be actively participating in the community. An aggregate community may be stored, e.g., in a database.

Linked. A linked ensemble is a network comprised of the population or a subpopulation of the aggregate ensemble of those members with ports that are, or are attempting to, link to other members to form the network.

Synchronized. A synchronized ensemble is a population of participants' ports that are linked into a network such that the ports' activities are time locked in some fashion.

Ensembles are created by the behaviors of ports.

A community may be synchronized.

A cluster is a set of ports that synchronized with the community it is a member of. A cluster's ports have a higher coefficient of connectedness than other members of the community.

A group is separate from any other ensemble.

A group may emerge from any other ensemble.

A group may not originate in the Group-Spaces environment, only from another ensemble.

A community originates in the Group-Spaces environment.

Ensembles may be instantiated, virtual or logical.

Port states

Ports create ensembles. Ports may assume any of several states. These states include:

Active: Conditions: unsynchronized, synchronized, linked;

Passive: Conditions: unsynchronized, synchronized, linked;

Idle: unsynchronized, unlinked;

Suspended: Idle with data structure stored.

Packets and Pings

Ports exchange data blocks. Two classes are described here: pings and packets. Others may be employed. The use of pings to maintain synchronization and link validity has been described previously. A ping may have:

Target link ID

Source link ID

Type (ping, ack, data 1, data 2)

Data: Z and/or β, δ.

A packet is a data block that carries information between nodes such as messages, invitations, responses, redirections, links and controls. The invitation is a specific form or type of message. A packet may have:

Target link ID (P_j)

Packet metadata:

- unique community ID
- unique packet ID
- packet type declaration
- unique message ID
- exchange or session ID
- cluster ID
- expiration time

Message:

- contents
- integrity code

Security:

- community
- group
- message
- exchange or session

An invitation to join a group is an example of port behaviors relative to packets. Groups are frequently transient, formed for a purpose "on-the-fly" and disbanded when no longer needed. Participation in a transient group is negotiated A profile outlining such an exchange is shown in Figure 15. Security actions have not been included. A packet containing an invitation to join a group is received from another port, P_j. A packet is received through the connector J_i [Figure 7, 7001]. The source link ID is attached to the packet. The GS-Processor [7002] ensures that the community ID is appropriate (e.g., ID, security). If not, the packet is discarded. Optionally it is reported to security services. The identifier of the received packet is compared to the identifiers of previously received packets. If the packet has been received previously it is discarded. If the expiration time has passed, the packet is discarded. If the packet has not been received previously and it has not expired as indicated by the expiration time, it is immediately relayed to the other links through the output queue. Functionally the packet is immediately replicated for each of the *k*-1 target links, with the input target link replaced by the target links. These packets are placed in the output buffer Q [7019] to the connector J_i [7001] for output to all of the links except for the one it was received from. More efficient buffer management strategies may be used. The pings and packets are intermixed in the order received and generated to ensure that the port and transport delays reflect actual traffic.

The GS-Processor [Figure 7, 7007] places the packet's key contents into the buffer queue in the GS-Buffer Layer [7010] to the Interpreter [7009].The port interpreter determines the packet type for subsequent processing. If the packet is an invitation to join a group a check is made of the availability status of the participant. If the participant is not available, the packet may be processed for later action or discarded according to the user's profile. The processor compares the invitation specifics, some of which might be in the in the metadata or message, against the participant's profile stored in memory (7008, local or remote). The user may have a different profile for each community he or she is a member of. If the invitation packet metadata does not match the user's profile according to some rubric, the packet is discarded.

If there is an adequate match between the packet metadata or message and the user's profile, the message is interpreted and converted to a form that will be meaningful to the user. That information is placed in a Q buffer [7013] in the I-Buffer Layer [7012] for transfer to the Interface [7014]. The interface presents information to the user's platform (e.g., personal computer, tablet, smart phone) over an appropriate communication link, thereby notifying the user or an application or plug in on the user's platform of the invitation with the specifics provided. If the user ignores the invitation, nothing happens. "Ignoring" is a default operation. The user may request admittance to the group. The participant, working through a local application, an interface and the interpreter, constructs a message containing relevant aspects of the user and availability. This is packaged through the interpreter with packet, message, session and other information and passed through the GS-Buffer layer to the GS-Processor that places it in the output buffer Q. The packet is subsequently passed to all k links for distribution into the community space.

The user may have a pre-established response profile to specified invitations, either requesting membership or ignoring the invitations. Other automated responses to messages are possible.

The inviting port inspects its incoming packets, searching for references to its inviting packet ID. In addition to following the normal relaying rules, upon detection of such an identifier, the packet is passed up through the port layers, with appropriate processing, to the interface for the action of accept or decline by the inviter. Some or all of the decision process may be automated according to the participant's commands. If the request to join is accepted, a return packet is sent out through the synchronized community to all links. This packet may contain information directing the invitee how to participate in the group, e.g., link to one or more specified link IDs, finding links from a specific directory or connecting to an out of band activity space such as a game space, Web meeting or URL. The invitee may be directed to post some of its links to the group directory for use by others within the group. As the group forms, a new intrinsic synchronization time is developed. It may be shorter that that of the community, although this is not required. Pings will be within the group. Most of the packets will be relevant to the group. The average number of hops, ℓ, between any two ports will be reduced:

$$\ell = \frac{\ln(N_g)}{\ln(k)}.$$

N_g is the size of the group. The newly formed group will “snap” into existence with rapid responses. The fast response and the specificity provide a sense of intimacy and immediacy to the group. As shown in Figure 9, such a group could serve as the backbone control system for a simulation or game in which the Group-Spaces messages in packets[9001] within the group [9004] control the game while the URL or similar activity space provides the visualization and interaction interface [7003].

Such processing may appear to create a load on the host that causes an inordinately slow behavior. Such is not the case. The agents operate in parallel. Not all ports will process information for a participant but may be in a passive mode, only maintaining synchronization (synch) by testing and relaying messages and maintaining critical timing, as described below. As all of the processing occurs within the hosting system considerable operational economy can be achieved. Data blocks may require simpler encryption methods—if any. Much of the metadata can be reduced to identifiers using the community’s taxonomy and metadata structure. Critical timing can be determined external to the actual links, providing the same function as the ping exchange method, although the minimization of timing may not be as efficient.

Instant groups

Among other uses of complex synchronous this embodiment of the invention describes methods and systems for forming groups on very short notice. The methods and systems may be accomplished in a number of environments, not restricted to those described below. Instant groups are formed from members of a community. The instant group does not require it to be a pre-existing group. Its membership may potentially be open to all or some subset of members in the community. A member (originator) in the community may issue an invitation (or request) to join a group that has some level of specificity in the invitation. Each recipient’s port (or proxy) tests the invitation’s description against the recipient’s profile of acceptable invitations. If the match fits at some prescribed level, the request is then passed on to the recipient’s platform for the recipient’s potential response. If the recipient accepts the invitation, that response may be sent back to the originator. The originator may accept or refuse the response. If accepted, the recipient is allowed to join the group Specific instructions may be sent to the recipient or proxy) describing how to join the group. This may be an activity space (such as a Web meeting, simulation, game, voice communication, whiteboard or chat room). The activity space may occur within or external to the community network. Optionally, when all members have left the group, it disbands. It the group may, but not must, be disbanded (adjourned) at the discretion of the originator.

Embodiments of a system and methods for rapidly automatically create collaborative experiences, bringing together people and resources when they are needed are described. The system and methods provides transient groups that emerge from distributed interpenetrated communities [].The system that is capable of creating communities and the groups that may emerge from them is called herein “Group-Spaces.” Group-Spaces can be created in a complex network using methods and systems reflecting the work of Kuramoto and others as described above. A complex network is a system with nodes (or vertices) connected to links (or edges) with a degree, or number of links (or edges), of each node. Within Group-Spaces a system of ports create a synchronous real or virtual peer-to-peer (p2p) environment. Although the system is a p2p system, in can incorporate other processes as a hybrid system. The system has several components that individually and in concert create unique properties.

Activity spaces

An ensemble, most notably a group, may institute an activity space for its participants. As the group activity space is the most common, it will be used to illustrate the systems and methods. An activity space may constitute message protocols and formats within the group for the rapid exchange of information. For example, a group might have chat capabilities. Specialized codes, tokens and formats may be used. An activity space may be accessed outside of the group in another location, such as a Web location accessed by a URL. Each group within the group-spaces may access separate activity spaces in the Web as illustrated in Figure 16. Activity spaces may include:

- Chat
- Web meetings
- Voice over IP (VoIP)
- Whiteboards
- Enterprise applications
- Office
- Operations base
- Coffee house
- Studio
- Clinic
- Simulation
- Game

As the size of a group will be smaller than the ensemble from which it derives, its communications speeds will be faster. This permits larger, more complex or specialized messages to be exchanged. Activity spaces, such as simulations and games, may have ports into the group [17001] as illustrated in Figure 17, supporting high speed direct control of the activity space [17002] outside of the users' Web connections. The activity space may use the ensemble trust services.

Cross-community ensembles

Ensembles may be formed across communities [Figure 18, 18001, 18002], effectively joining the communities through a small number of cross links [18004] (W-S model) as shown in Figure 18. This same method can be used to continue an ensemble over multiple servers. In both cases, the domain of an ensemble is extended. Such methods may be useful in cloud computing. As illustrated in Figure 19 two or more ensembles [19001, 19002] may be synchronized through the incorporation of an additional ensemble, a synchronizer [19003]. The two ensembles may be linked through the synchronizer without a direct link [19005] (see Fischer et al. 2006) or directly coupled [19005] with a common synchronizer [19003] (see Zhou & Roy 2007).

Metadata

Profiles may use standard metadata of data structures and taxonomies. Extensions may be permitted. The ensemble resources may provide the metadata system allowing the use of tokenized structures and taxonomies that may provide fast creation and interpretation of profiles. The metadata system may support rapid creation of messages. This may be accomplished with specialized tools, applications, apps and plug-ins. Ports may be provisioned with metadata data structures and taxonomies within the ensemble or out-of-band.

Profiles

Participants and ensembles may have profiles that describe the interests of either. The participant's profile may change frequently, including expressing a willingness to participate on specific topics and current availability. The ensemble profile provides information about the ensemble that may be reflected, via taxonomies, in the participants' profiles. A participant's profile may:

Establish memberships in ensembles (e.g., communities)

Describe personal interests for each community

Establish security methods

Set availability

An ensembles profile may:

Permits membership in ensembles (tokens)

Describe each ensemble

Set security methods

Set duration rules

Define activity spaces

Applications (client-side software, apps and plug-ins)

Applications, apps, tools, utilities and plug-ins may assist the participant in the management of profiles, creation of profiles and status updates, generation of packets (e.g., messages, invitations and responses) and the interpretation of packets. Standardized metadata may be used.

Ensemble resources

The term "ensemble" is intended as a general term to span the spectrum of collective sets within Group-Spaces including community, group and cluster. An ensemble is a set with common interests. An ensemble may have a set of common resources. Some of these resources are data values, others may be services. Ensemble resources may exist at a single access point, such as a port or URL, or be distributed over several access points. Ensemble resources may include:

directory: available links. A port may use its ID or provide unique IDs for each of its links according to the rules of the ensemble.

rules: descriptions of behaviors of ports

constants: values used in the maintenance of synchronization and group structure

variables: Values that may change. These values may be held in the resource site for access by ports or may be distributed to the group members. Distribution may be within the p2p network or externally (out of band).

packet structure: As described above, the packet is defined for the community. In a group the packet characteristics may change from the community packet characteristics, e.g., size, allowable taxonomies, and security methods.

trust services: A port entering a community or group may require authentication, authorization, and encryption services. Any port within a community or group may also require such services.

gateway: as a port enters a group or community, it must establish one or more links with one or more other ports that will assist in synchronization. This initial link pro

bridge: a connection between two or more independent ensembles. These may or may not be synchronized. Security methods and services may be used.

metadata schema: a formal structure for terms from the taxonomy or other sources.

taxonomies: a commonly agreed upon structured vocabulary

activity spaces: sites that may be outside of Group-Spaces such as games, simulations, Web meeting rooms, voice over Internet protocol (VOIP), white boards, and physical meeting spaces.

Trust services and security

Several methods and systems, individually and in combination, provide security within the Group-Spaces environment. The timing of a ensemble can make it difficult for an uninvited participant to join as approximately the right timing constant is needed to avoid being pruned out of the ensemble (see below). Encryption can be used. The standard trust services of authentication, authorization and encryption may be provided. Trust services may be provided by the ensemble's community resource. A virtual p2p [20001] can be behind a firewall as in Figure 20 [20002]. Security services may involve some out-of-band provisioning and other communications. Such security technology is well known.

Anonymity and invisibility

A port or node may remain anonymous, thus a participant may also remain anonymous. The may publish links to the directory that have unique identifiers (IDs), not referring to any unique port ID. If an ensemble is synchronized as described above a port will know the unique IDs of the link. A system or method may allow direct knowledge of a port ID, supporting directed messaging to a specific port. This is not required. An ensemble may allow the exchange of any identification or profile information that the rules of the ensemble allow. Those are local negotiations that may be outside or new types of the normal packet messages.

If an invitation is received through a profile-based filter the participant or its proxy may wish to examine the invitation in more detail. A participant may be invisible. If a user receives an invitation that user may ignore the invitation. The source of the invitation may not be aware of the targets that receive the invitation. A participant may respond to an invitation without revealing his, her or its identity, remaining anonymous.

Joining

Joining a synchronized ensemble may entail becoming synchronized with the community or group, passing security tests, using the community or group ID, security tokens and methods. A community resource may provide these methods and systems for a participant's entry into the group. As an example, a port seeking to join a group following an introduction upon responding with a request to join, for example in a message using the invitation packet or message ID, may be provided with a link for further validation and instruction. This response may be targeted to the respondent in any of several ways including use of a trust service, the respondent providing a new link ID or the inviter sending a packet into the ensemble or network using a unique packet or message ID provided by the respondent. Following appropriate security checks by the ensemble services, the port is provided with initial variables and constants. The port may link to a gateways port that provides initial synchronization. The joining port is given access to the directory of available links. The port selects a link at random and connects to it. The directory, or some other service, may check to ensure that ports do not link multiply to each other. The port continues to add links from the directory until it reaches the limit, k, for the ensemble. If the directory becomes empty before the port fills its directory requirement, it may submit its unfilled links to the directory. During joining or group formation, the minimum number of links required may be suspended.

Leaving

Leaving an ensemble can be accomplished by freeing up all links, not attempting to establish new links with those released, and not publishing links. Published, unfilled links may be either retracted from the directory or not responded to, presuming the port attempting to link will be able to detect that the link is inactive, such as exceeding a response time. A port may announce that it is leaving, e.g., removal from the directory or an announcement to the ensemble. An unannounced departure is executed by terminating links. A port that is not part of an ensemble is not synchronized with any ensemble. It does not exchange pings or forward packets.

Pruning

As described above, if a link is unable to maintain the synchronization requirements, the port may prune that link. Typically this may occur with the timing of the link does not fit within the required span of values. The pruning port now has a link available for a new linking. A new link ID may be used. Similarly, the pruned link's port now has a free link. If sufficient numbers of a port's links are pruned by other ports, the port may not have the minimum number of links required by the rules. At this point the port itself is pruned from the community. If a port attempts to rejoin the ensemble, some form of joining method may be required.

Pruning may serve to maintain the integrity of the ensemble.

Group formation

A group is a synchronized network this is a subset of a community, hence is equal to or less than the size of the community from which it is derived. A group may be synchronized even though the community is not, with the community function executed in some other manner such as an unsynchronized p2p network of a centralized data center. Using the synchronized community as a model, two methods of group formation can be illustrated: spawning and shedding.

Spawning

As illustrated in Figure 21, spawned group [21002] is formed by a new set of ports that leave the synchronized community [21001]. As a participant is accepted into a group, the participant's node creates, or "spawns," a new port [21007] that becomes synchronized with the forming group [21002]. The group may provide the new ports with some specifics for joining the group such as an initial intrinsic timing, a group ID, a group security token, a list of one or more links, the location of the group resources, a URL and any other such information appropriate to the group. A spawned group closes or disappears when the last two ports leave. As the community resource may be a ported member, the group may exist until the last participant leaves. A group may persist if there are two or more active ports, according to any rules that the group or community may have, for example, a minimum number of links. Ports that do no more than relay unique, unexpired packets and maintain synchronization, for example through ping exchanges, may allow a group to persist even when all human or other active members have left the group. These are "dummy ports." A persistent group may maintain an entry service allowing participants to rejoin the group.

As a port is allowed to enter a group, some synchronizing information is provided, as described above. A gateway function may provide an initial link for synchronization under conditions that satisfy the security requirements. The gateway function may access the trust services for security purposes.

Shedding

A group may form from ports that leave (or are shed or pruned by) the main community. Under the shedding paradigm, no new port is created. The port shifts ensemble membership. The process of joining the group is essentially the same as that for a spawned port. A port may force itself to be pruned from the community as its new links to the group are formed. The pruning mechanism may be automatic if the intrinsic time, τ, of the original ensemble and the target ensemble are adequately different.

Management

Ports

A passive port does not communicate with the user interface, but exchanges pings and relays packets according to the ensemble's rules. A port may be in a passive state in an ensemble that has sufficient other ports to maintain synchronization. If a passive port is unused for extended period of time, a port or agent's state and profile may be stored in a database to be reconstructed as needed. Thus a port may be temporarily destroyed to be later constructed (initiated) from stored information. This may optimize the operation of the environment.

Optimization

The management system may inspect and analyze patterns of activity to balance the load within the host and to serve the participants well. The patterns of messages and history of group formations may indicate potential de facto clusters rather than clusters intentionally established by the participants. These de facto clusters will provide a faster response time for those who frequently interact in groups. Port and community activity logs may be inspected for optimization.

Advertising

Community metadata can provide information to support targeted advertising. Pooling of participants' information such as profiles and port logs can maintain privacy while supporting targeted ensemble advertising. Such advertising may also appear in the activity spaces according to the characteristics of the group.

Conclusion

What do arrays of oscillators have to do with the rapid formation of working groups over a network? Kuramoto (1975, Kawamura et al. 2008) described the behavior an array of *locally* coupled oscillators. Under the proper conditions all of the oscillators in the array will oscillate in unison, being both coherent and synchronous. This model can the basis for the behavior of a system of communications nodes. Messages in such a system can be used to rapidly assemble a group of relevant experts in a single virtual meeting place to take action focused on a specific issue; examples are providing decision support and educational collaborations. Security is distributed throughout the system. Such an array of users' ports, Group-Spaces, will have both balanced communication paths and a low average message propagation time from any node to all other nodes in the network. There is no central command. The system is robust, fault tolerant, inherently secure and degrades gracefully. The messaging is anonymous. Messages can originate and be responded to any place in the Group-Spaces ensemble. The message propagation is robust, being tolerant of missing nodes and transient variations in communication links. The Group-Space system is predominantly oriented toward rapid *action* rather than resource *location*. It is about *doing*. The properties of a Group-Space can be modulated to create clusters, to establish modularity, and to manage connections among groups. The Group-Spaces environment may be created in a host or cloud as a virtual peer-to-peer network with users' ports that access ensembles created in the environment. The actions of the ports, e.g., as agents, may create the ensembles. It may self-administer for optimal properties.

While the foregoing written description of the invention enables one of ordinary skill to make and use what is considered presently to be the best mode thereof, those of ordinary skill will understand and appreciate the existence of variations, combinations, and equivalents of the specific embodiment, method, and examples herein. The invention should therefore not be limited by the above described embodiment, method, and examples, but by all embodiments and methods within the scope and spirit of the invention.

Notations and symbols

N = Number of ports in the ensemble or network

P_i = Port i.

i = port i designation

j = port j designation of a port, P_j, that is linked to port, P_i.

$L_{i,j}$ = designation of link from port i to port j. by port i

τ_{ij} = transmission delay between ports P_i and P_j.

t = time

K_f = the f^{th} Group-Space

M_i = i^{th} magnet

θ_i = Angle of i^{th} oscillator

$$\dot{\theta}_i = \frac{d\theta_i}{dt}$$

ω_i = frequency in radians per second of ith port.

k = number of links of a given port to other j ports, $j = 1..k$.

$k_{\min}$ = minimum number of links

φ_{ij} = the phase difference between ports i and j.

λ_{ij} = coupling strength between ports i and j.

ℓ, = the mean number of hops, for a message to travel between any two points in a network $(\ln(n)/\ln(k))$.

δ_i = delay between the time P_i receives the message and the time it sends it out

β_{ij} = round trip time of a ping from P_i to P_j and back

ε_i= timing error of ith port

ξ = coupling coefficient

ξ_{ij} = link coupling coefficient between ports P_i and P_j.

ID = Identification

C = Clustering coefficient, essentially the fraction of actual link triangles (three vertices forming a loop) over the possible triangles.

b = bias offset

σ = Allowable deviation

Δ = mean or minimum ping interval

References

Acebrón, J.A., Bonilla, L.L., Pérez Vicente, C.J., Ritort, F. & Spigler, R. (2005). The Kuramoto model: A simple paradigm for synchronization phenomena. *Rev. Mod. Phys*., **77**(1), 137-185.

Arenas, A. , D´iaz-Guilera, A., Kurthsl, J., Moreno, Y., & Zhou, C. (2008). Synchronization in complex networks. *Physics Reports,* **469**(3), 93-153.

Belykha, V., Belykh, I., & Hasler, M. (2004). *Connection graph stability method for synchronized coupled chaotic systems*. Physica D: Nonlinear Phenomena, **195**, 1-2, 1 August, 159-187.

Fischer, I., Raul, V. , Buldu, J. M., Peil, M., Mirasso, C. R., Torrent, M.C., & Garcia-Ojalvo, J. (2006). Zero-Lag Long-Range Synchronization via Dynamical Relaying. *Physical review letters*, 2006, **97**(12) : 123902.1-123902.4.

Kawamura, Y., Nakao, H., Arai, K., Kori, & Kuramoto, Y. (2008). *Collective Phase Sensitivity*. Phys. Rev. Lett. 101, 024101.

Kuramoto, Y. (1975). Self-entrainment of a population of coupled non-linear oscillators. In *International Symposium on Mathematical Problems in Theoretical Physics*, 420-422.

Moussaid, M., Garnier, S., Theraulaz, G., & Helbing, D. (2009). Collective Information Processing and Pattern Formation in Swarms, Flocks, and Crowds. *Topics in Cognitive Science*, **1**(3), 469-497.

Olfati-Saber, R. F., J. A. Murray, R. M. (2007). Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings- IEEE* 2007, **95**(1), pages 215-233

Tanaka, H., Hasegawa, A., Mizuno, H., & Tetsuro E., Senior Member, IEEE (2002). Synchronizability of Distributed Clock Oscillators. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, **49**(9), September 2002.

Triplett, B., Klein, D. & Morgansen, K. (2006). Discrete Time Kuramoto Models with Delay. In *Networked Embedded Sensing and Control*, 9-23.

Wagemakers, A., Buldú, J. M., & Sanjuán, M. A. F. (2007). Isochronous synchronization in mutually coupled chaotic circuits. *Chaos* **17**, 023128; doi:10.1063/1.2737820 (7 pages).

Zhou, B. B., & Roy, R., (2007). Isochronal synchrony and bidirectional communication with delay-coupled nonlinear oscillators. *Phys. Rev. E* **75**(2)l 026205 (5 pages).

What is claimed is:

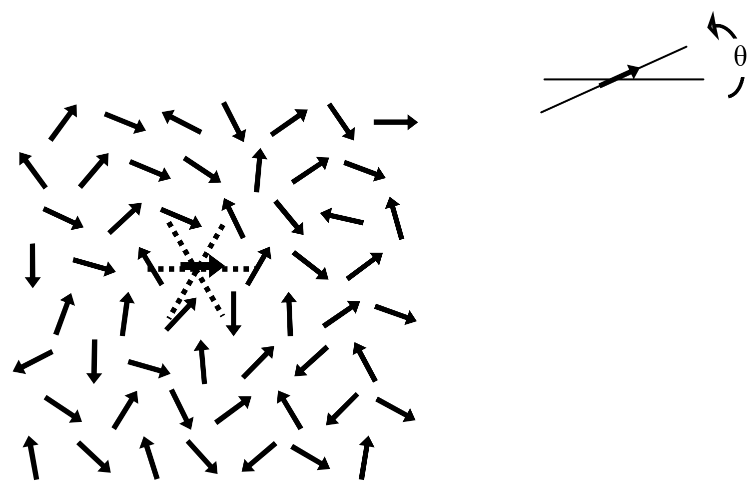


Figure 1

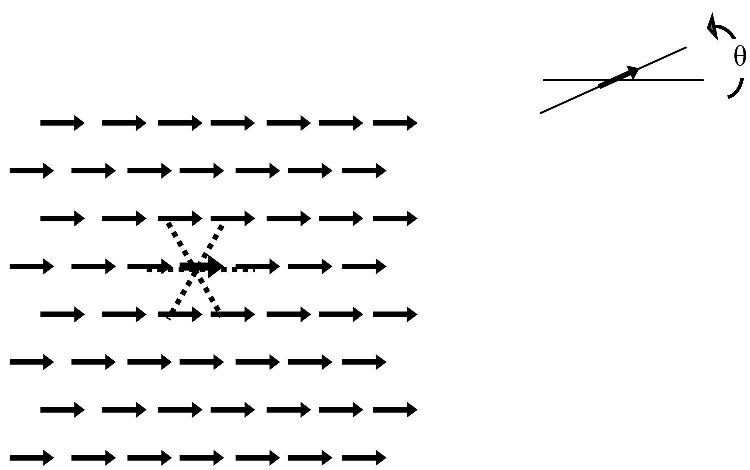


Figure 2

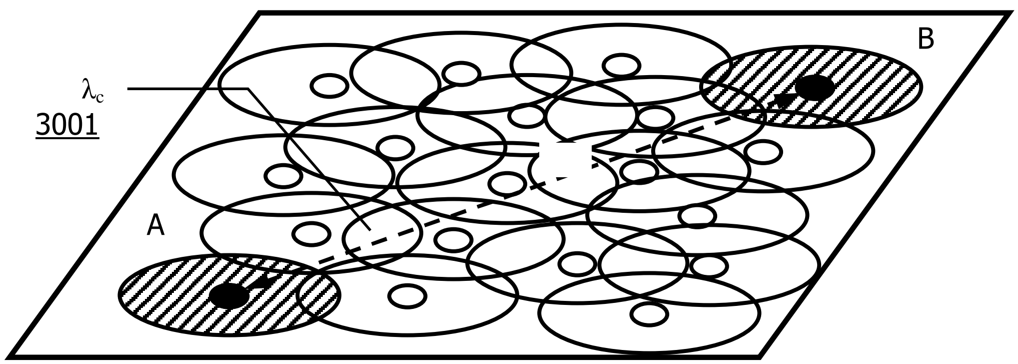


Figure 3

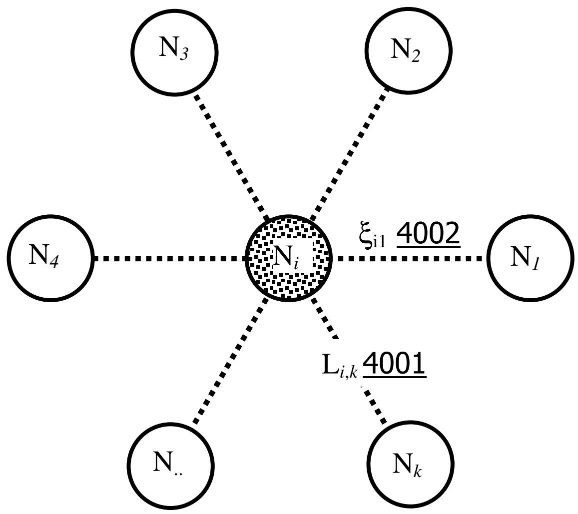


Figure 4

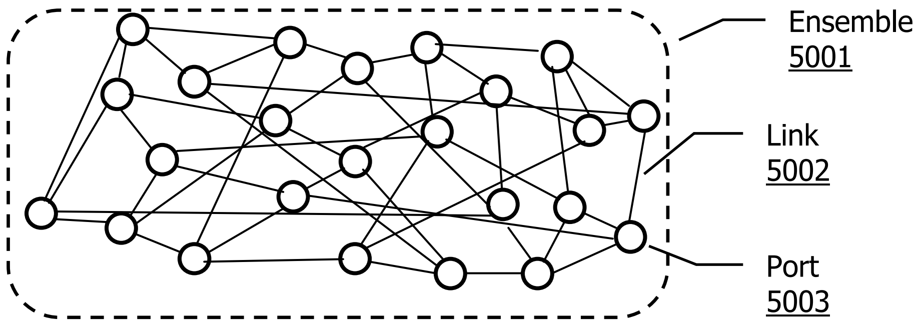


Figure 5

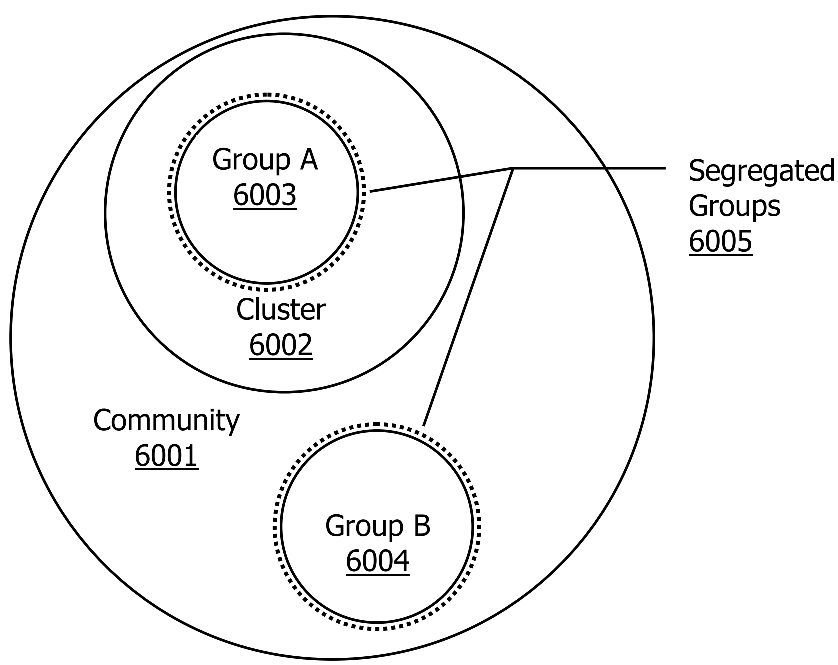


Figure 6

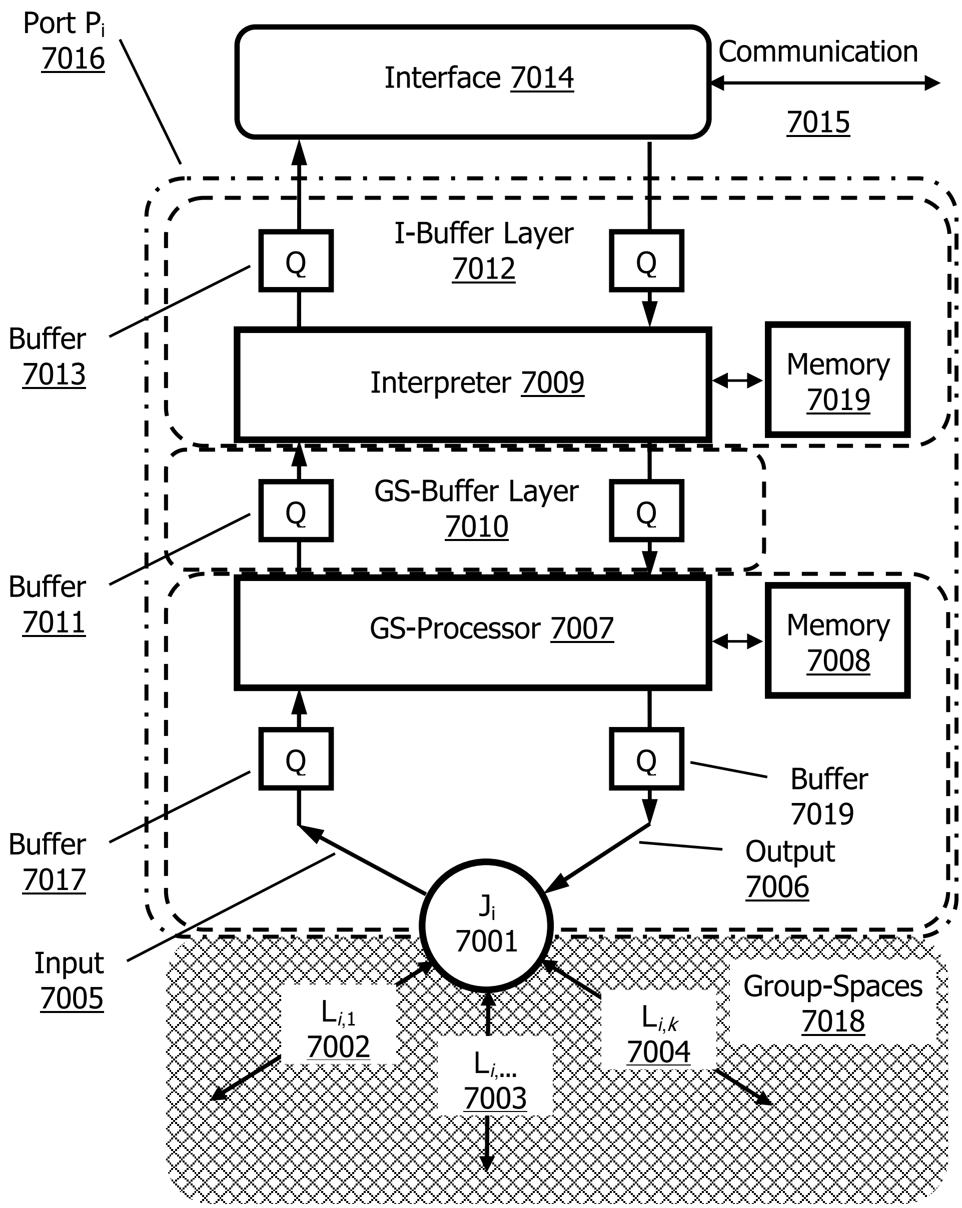


Figure 7

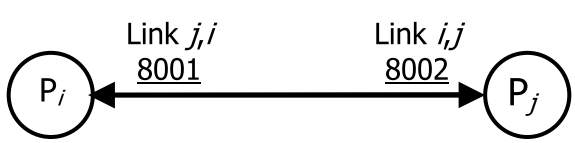


Figure 8

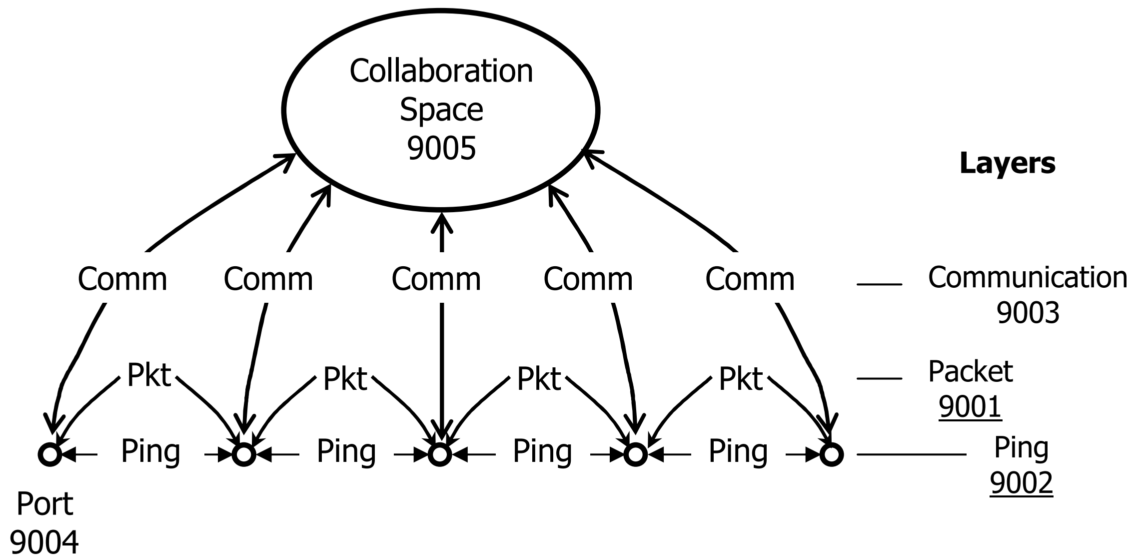
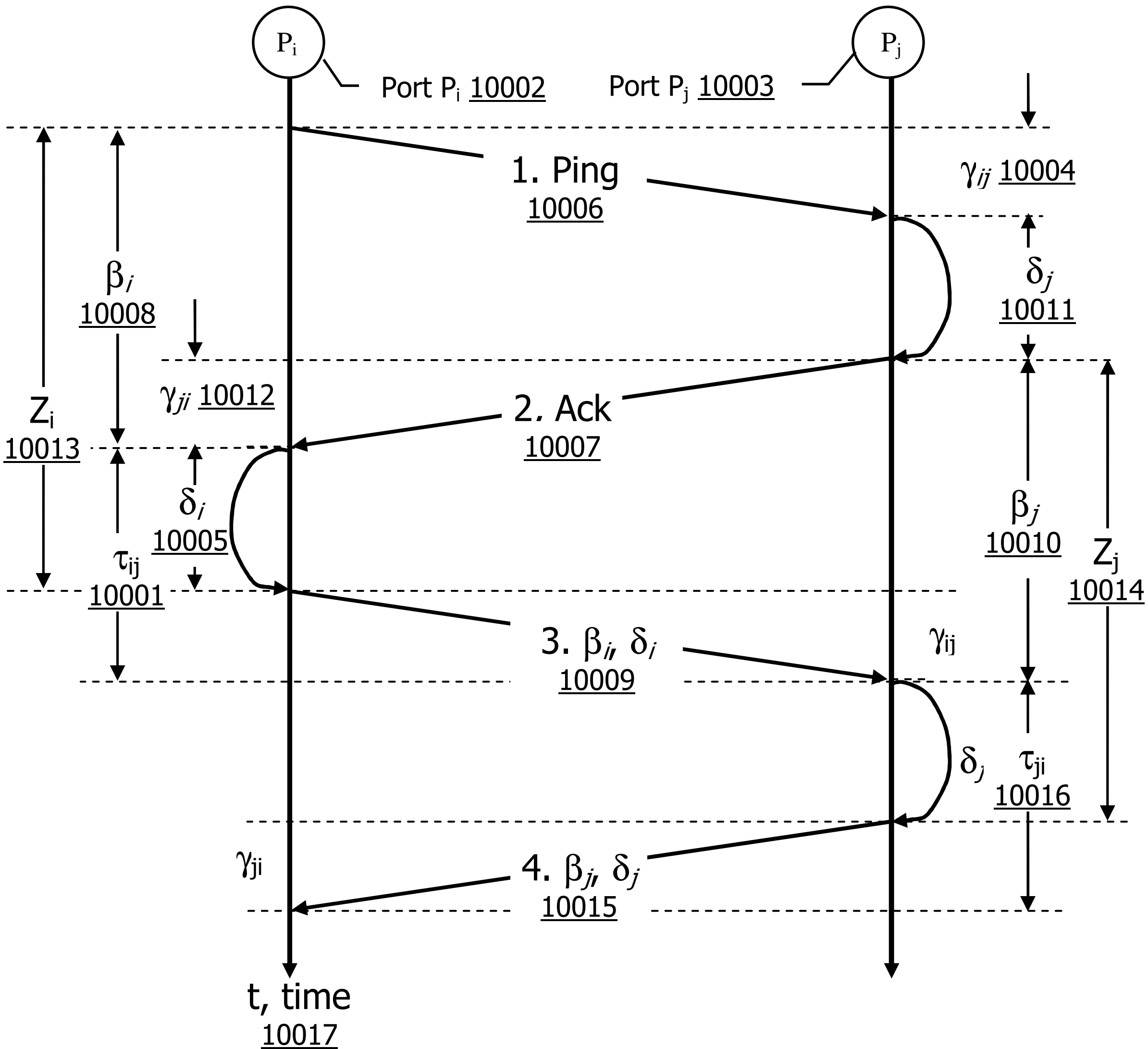

Collaboration
Space
9005
Layers
Comm
Comm
Comm
Comm
Comm
Communication
9003
Pkt
Pkt
Pkt
Pkt
Packet
9001
Ping
Ping
Ping
Ping
Ping
9002
Port
9004

Figure 9



Pi
Pj
Port Pi 10002
Port Pj 10003
1. Ping
10006
γij 10004
βi
10008
δj
10011
γji 10012
2. Ack
10007
Zi
10013
δi
10005
τij
10001
βj
10010
Zj
10014
3. βi, δi
10009
γij
δj
τji
10016
γji
4. βj, δj
10015
t, time
10017

Figure 10

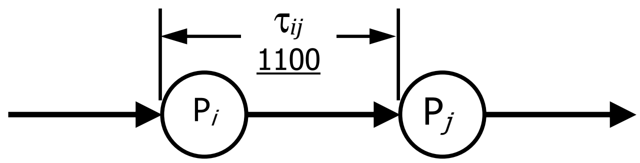


Figure 11

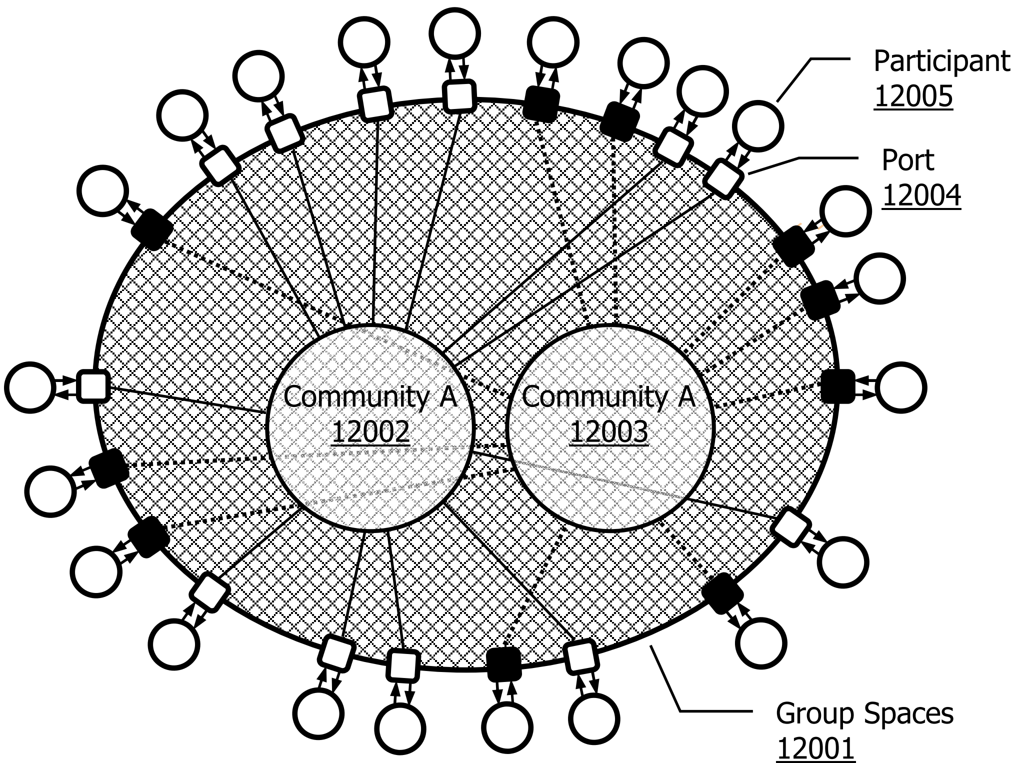


Figure 12

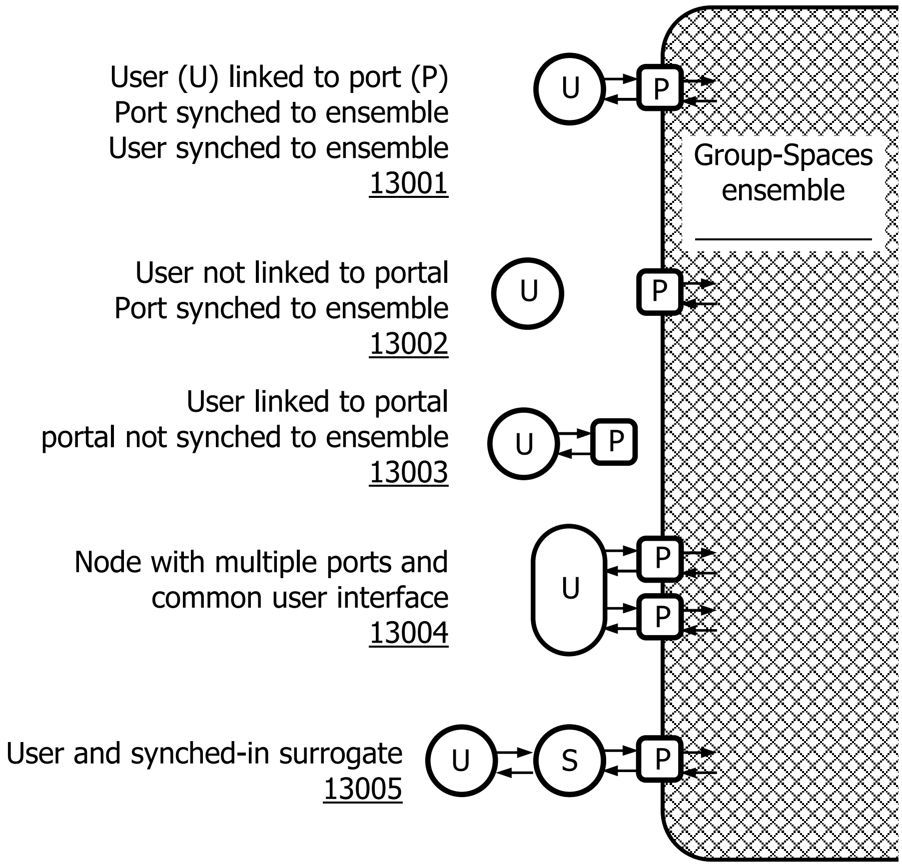

User (U) linked to port (P)
Port synched to ensemble
User synched to ensemble
13001
User not linked to portal
Port synched to ensemble
13002
User linked to portal
portal not synched to ensemble
13003
Node with multiple ports and
common user interface
13004
User and synched-in surrogate
13005
U
P
Group-Spaces
ensemble
U
P
U
P
U
P
P
U
S
P

Figure 13

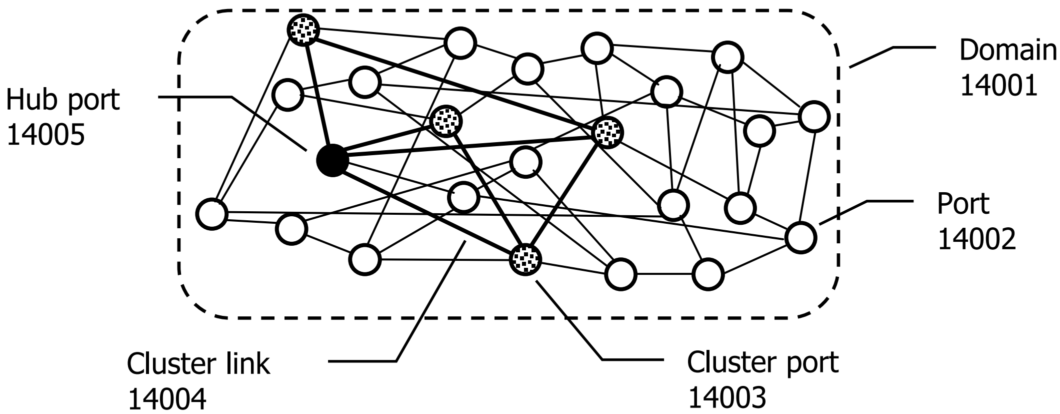


Figure 14

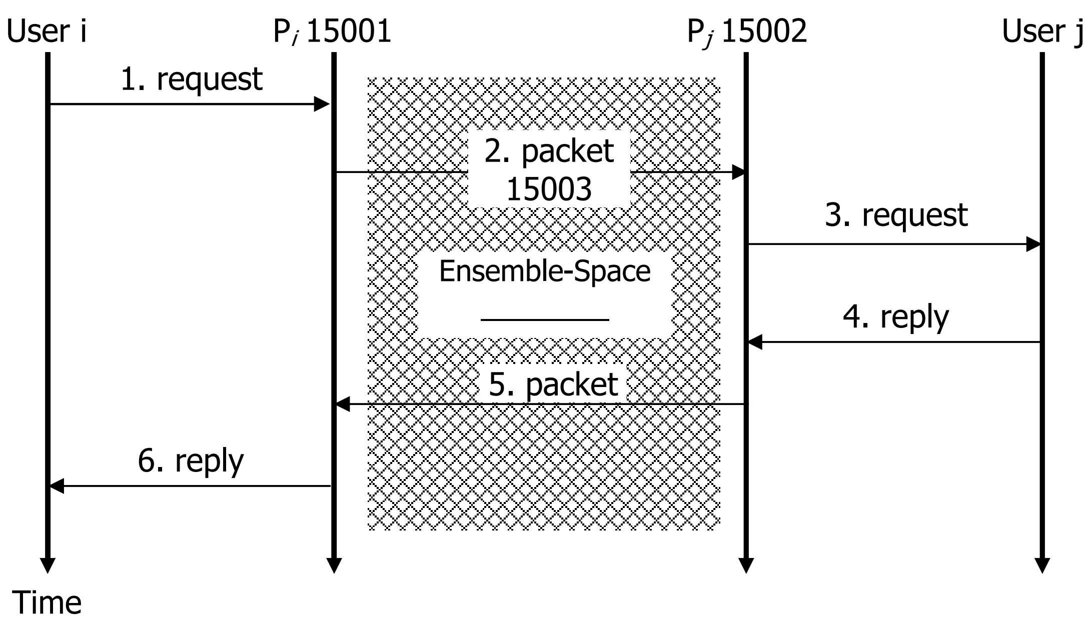


Figure 15

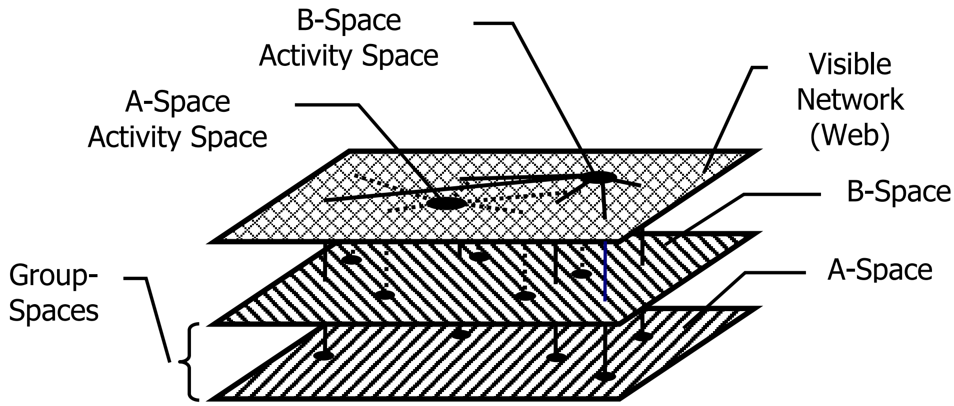


Figure 16

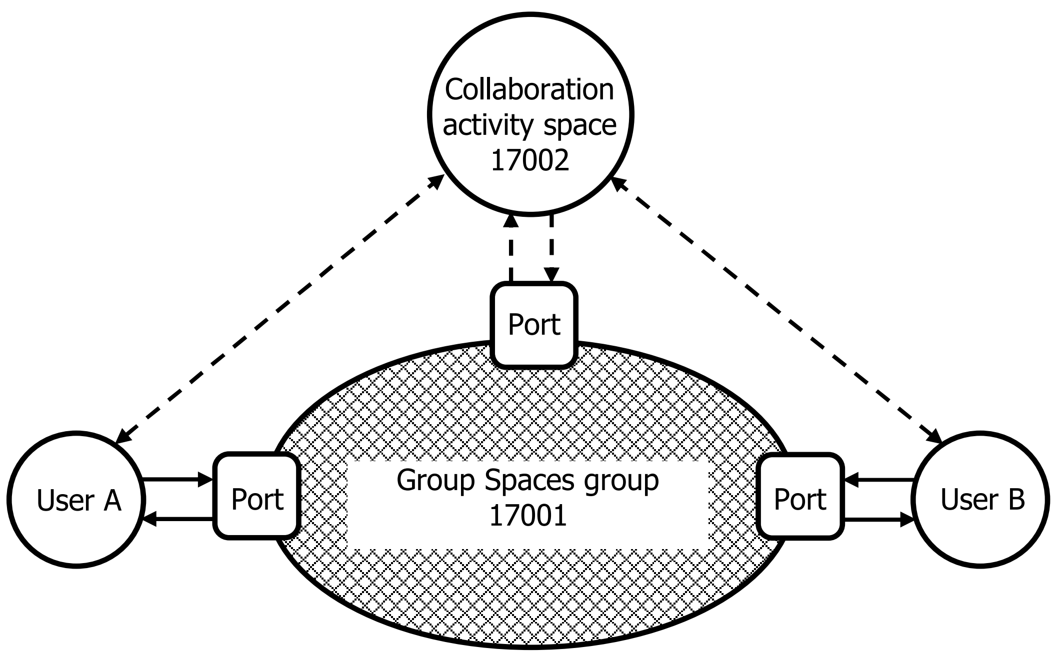


Figure 17

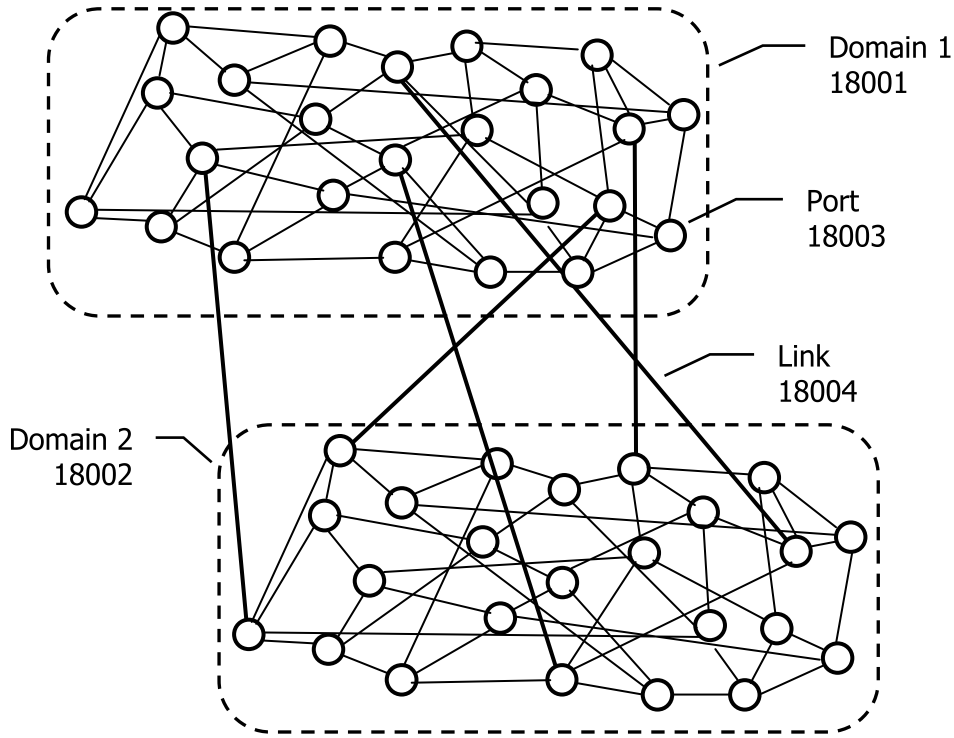


Figure 18

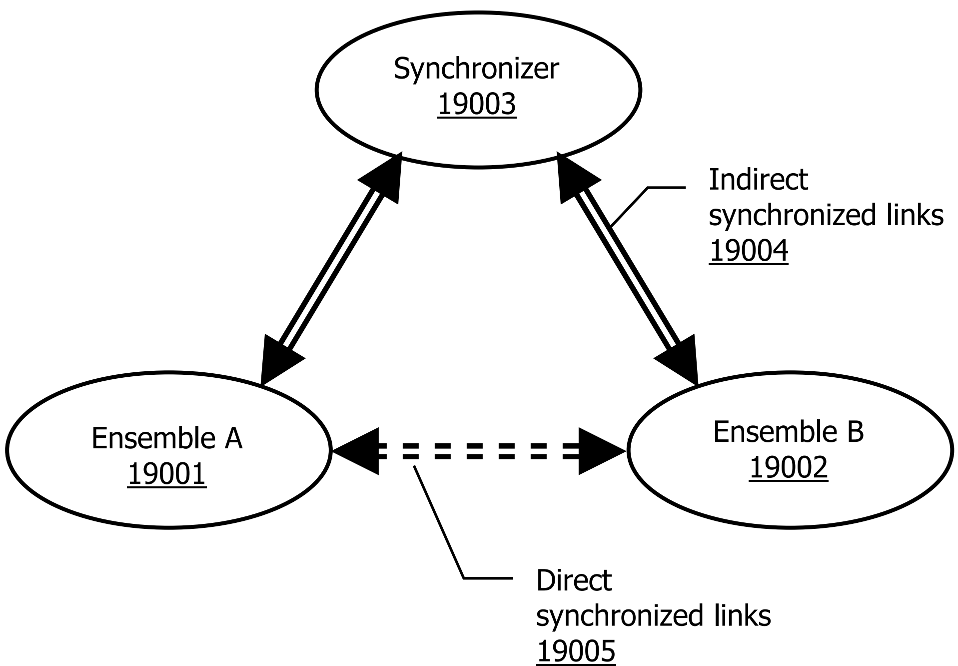


Figure 19

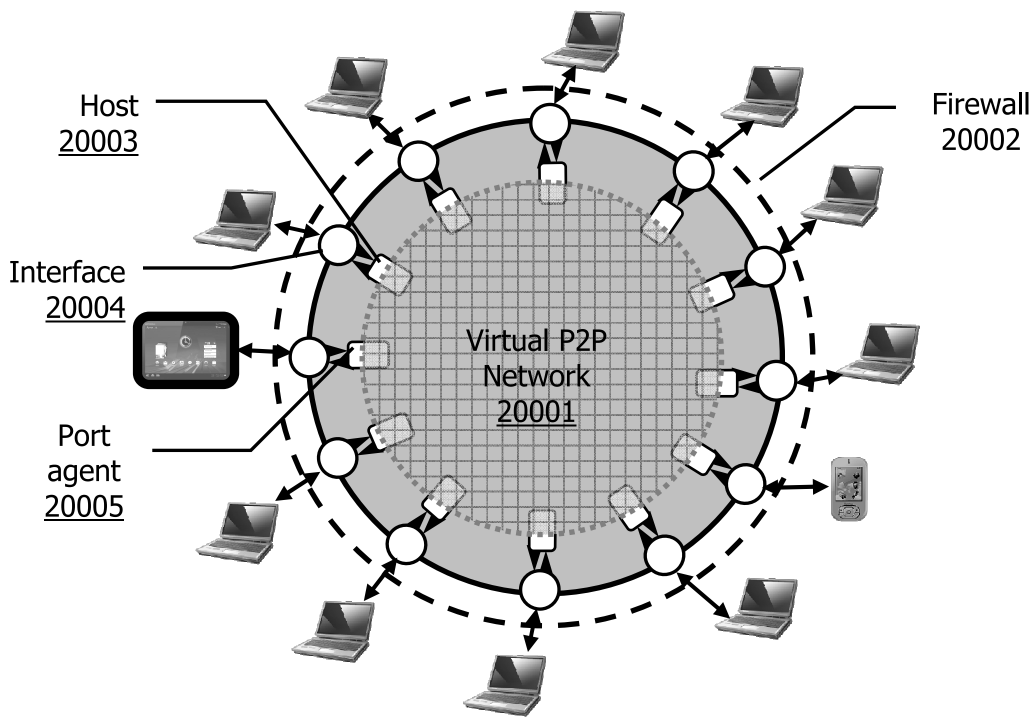

Host
20003
Interface
20004
Port
agent
20005
Virtual P2P
Network
20001
Firewall
20002

Figure 20

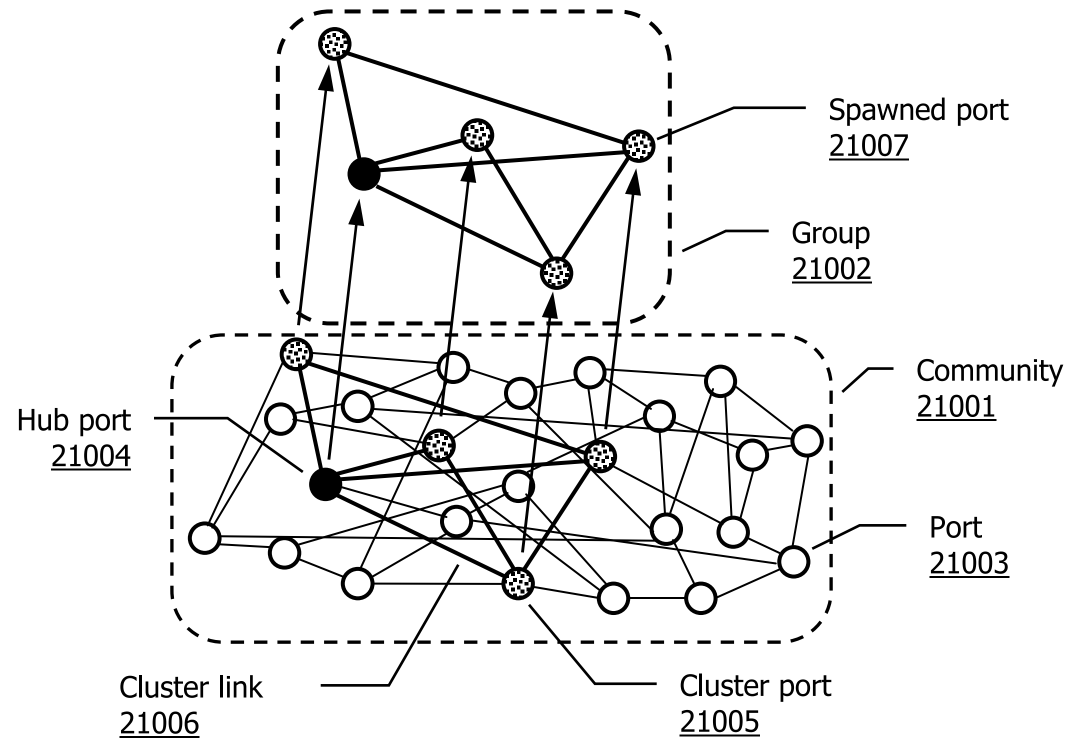

Spawned port
21007
Group
21002
Community
21001
Hub port
21004
Port
21003
Cluster link
21006
Cluster port
21005

Figure 21

ABSTRACT

A directed social network can be created in a synchronous peer-to-peer environment. The complex network environment can support the simultaneous existence of multiple communities. This environment supports the rapid formation of potentially transient groups drawn from the communities in the environment. The complex peer-to-peer network may be real or virtual. Timing control is used to establish and maintain synchronization. The timing control may arise from the actions of independent agents or ports that comprise the complex network.