Beyond Anarchy: Self-Organized Topology for Peer-to-Peer Networks

FABRICE SAFFRE AND ROBERT GHANEA-HERCOCK
BT Exact, Intelligent Systems Laboratory, BT Adastral Park, B62 Orion Building, Zone 5, pp12, Martlesham Heath IP5 3RE, United Kingdom

Received January 16, 2003; accepted August 8, 2003

In this article, we discuss the new constraints that will apply to future, highly dynamic networks and investigate one of the major problems that is likely to affect their deployment: traffic/workload distribution. Starting with a very simple model, we demonstrate that a structure made of undifferentiated subunits will almost inevitably hit a scalability barrier, unless new design rules are applied. We then give quantitative evidence that a hybrid network topology termed the “hypergrid” could have the potential of solving some of the most critical problems (relay overload and network vulnerability), yet could still be produced by applying local rules only. © 2004 Wiley Periodicals, Inc. Complexity 9: 49–53, 2004

Key Words: dynamic networks; load balancing; robustness; scalability; self-organization

It has become increasingly clear in the past few years that many aspects of the world we live in can be described using the terminology of networks. Historically, the word itself would only have suggested some kind of human-built artifact, like a railroad, a power grid, or a telecommunication system. In less than a decade, we got used to talk about gene regulatory networks when evoking the complex interactions taking place inside Life’s blueprint (see e.g., Thomas et al. [1]). The sum of all relationships forming an ecosystem is labeled a “food web” (see e.g., Camacho et al. [2]). The very fabric of our society is explored using concepts derived from graph theory (see e.g., Girvan and Newman [3]). And of course, the technological structures that were once the most immediately recognizable embodiment of the term “network” are still there, playing an increasingly critical role in supporting virtually all of our species’ activities.

What most people working in this field often fail to emphasise though, because they take it for granted, is that most systems they describe are networks only on an abstract level. It is indeed obvious that a predator is not linked to its prey in the same sense as a telephone is connected to the local exchange. Yet the simple fact that there is no physical bond between them has a deeper meaning than it seems. A conceptual link does not have the same momentum as a copper wire, and the corollary is that the dynamics of immaterial networks are potentially much more complex than those of their physical counterparts.

THE FUTURE OF COMPUTING
This is particularly true now that manmade systems too are, so to speak, “going virtual.” As the widespread con-
fusion between the “worldwide Web” and the Internet already demonstrates, the border between conceptual relationships and physical infrastructure is rapidly fading away. Peer-to-peer and Grid computing have already started to introduce entirely new paradigms like that of “Virtual Organization” and their associated protocols and services (see e.g., Foster et al. [4]). Moreover, the wireless revolution effectively means that the supporting media itself can become intangible, with many constraints vanishing in the process. But of course, all this cross-referencing and overlapping between what’s material and what’s not comes at a price: simplicity in design, operation and, ultimately, behavior will not be among the strengths of tomorrow’s networks.

Actually, the challenges waiting for us in the near future can already be found, in the very limitations of some of the most inspiring findings about today’s Internet. Recent theoretical and experimental studies have shown that the topology of our planetary information space exhibits a variety of intriguing macroscopic features. These range from the very useful (19 degrees of separation, Albert et al. [5]) to the very worrying (if you know which nodes to target, you can destroy the entire system in only a few blows, see e.g., Albert et al. [6]). But most importantly, all of these properties result from a very specific network design, one in which all constitutive elements occupy a permanent and well-defined place in the global hierarchy. In this so-called “scale-free” architecture, highly connected nodes act as relays between more peripheral ones, and this pattern is repeated from one level to the next, resulting in a coherent, manageable whole. If this structural stability can no more be taken for granted, then everything from routing to resources allocation actually needs rethinking. In other words: when topology itself becomes highly dynamic (which of course virtual and/or wireless connectivity is all about), our present understanding of networks becomes largely irrelevant.

Hopefully, there will be other useful invariants, but finding and exploiting them may prove difficult. Can they replace those critical hierarchical features that are progressively disappearing? What new sources of pathological behavior will those complex and versatile associations of users and devices have to deal with and how can their reliability be guaranteed?

The intricacies of the problem can be illustrated by one of our own numerical experiments. It basically involves modeling a very simple system made of identical devices, none of which being capable of maintaining more connections than the others (and thus acting as a possible hub). In this type of design comprising no dedicated routers or relays, going from one node to another involves making a series of hops between similar devices. In the example shown in Figure 1(a), there is only one route between any two vertices and node usage obeys a very predictable pattern. Indeed, provided traffic is homogeneously distributed between all possible end points, the closer one comes to the centre, the higher the information flow.

This could be a major problem in an undifferentiated network, as in this very small system, one node (A) would already have to handle 13 times more traffic than its least busy counterparts (f, g, ..., q). Keeping in mind that all vertices are supposed to have similar capabilities, this “tree-like” design is obviously impractical. The conclusion is that imposing an upper limit on node degree is no solution for managing such system. On the contrary, by adding this one local constraint, one simply transfers the problem to a higher level. Indeed, trouble comes from the fact that node A is actually forced into a hub position, even though one could not tell from how many connections it has.
However obvious this may seem when looking down at the blueprint, it is important to understand that it is not necessarily the case from the insider’s point of view. Nodes A through e all have the same number of first neighbors, so A cannot directly conclude that it is being “exploited.” The result is that this situation could easily present itself in a real network undergoing a decentralized growth process, whereby nodes with available connections advertise for others to join. Early members are most likely to end up in the unenviable position of core relays, as newcomers gradually fill up empty spaces on the periphery.

THE HYPERGRID

Yet it turns out that the problem mainly comes from the fact that the one constraint so far only involves imposing an upper limit to node degree. Simply adding a new one whereby peripheral vertices are not allowed to have fewer connections than their more central counterparts, and one possible outcome is the architecture shown in Figure 1(b), termed the “hypergrid.” The design rules used to produce it specify that nodes should first be arranged in a tree. Then the remaining connections are cross-allocated at random between peripheral vertices. In a sense, the double constraint on node degree makes the hypergrid the exact opposite of scale-free, its closest relative being the “small world” network. Yet despite the fact that they share some important features, there is at least one critical difference: at the exception of the outermost layer, which is similar to a small world with a rewiring probability of 1, the hypergrid simply is a tree. The result is a hybrid topology with a typically very low clustering coefficient (“my neighbor’s neighbors are not my neighbors”).

So why is that useful and how can it help in managing complex networks? A first answer lies in the amount of traffic actually passing through the core. In the example shown in Figure 1(b), node A is part of only twice as many shortest routes as any peripheral vertex: on average, nodes belonging to [f, g, ..., q] are part of 26 such routes, versus 50 for the “hub” (in the tree, 208 of the same 17 × 16 = 272 directed routes pass through A). But more importantly, this relatively homogeneous distribution of the workload is maintained in much larger systems. The operation of a packet-switching network was simulated by having every node sending 100 packets to randomly selected destinations, resulting in the total amount of information exchanged being a linear function of system size. Those numerical experiments demonstrated that in a hypergrid of degree 4 as in Figure 1(b), but comprising 1457 nodes (7 layers), less than 1% of all packets sent along shortest routes still transit through the core. More precisely, simulations show the workload on the hub to be a logarithmic function of the number of nodes.

To put those figures into perspective, they should be compared with those found for the scale-free “counter-part” of the hypergrid (1457 vertices, comparable number of edges). The diameter of the graph is very similar (8 versus 9 for the hypergrid) even though the average path-length is inferior (4.53 versus 5.99). Yet in this control experiment, close to 20% of the traffic is routed through the most highly connected node (closest equivalent to the “hub”). Furthermore, comparison with smaller scale-free networks of similar design suggests that the workload on this main relay is nearly a linear function of the total number of nodes (it is actually a power law with the exponent slightly <1; see Figure 2). The obvious conclusion is that switching from scale-free to hypergrid topology, although marginally increasing the average number of hops between 2 randomly selected vertices, would result in a dramatic improvement in scalability. Indeed, central nodes would no more be condemned to support rapidly increasing traffic as the network grows, which might well turn out to be the single most critical problem facing large-scale distributed computing.

But there is yet another advantage: because the constraints are exactly the same for any node willing to join and at any time in the system’s history, the connection rules are very simple and easy to apply. Basically, to become part of a hypergrid of degree k, all a newcomer has to do is

1. Identify the innermost member maintaining “horizontal” connections
2. Request one of these links to be terminated and reallocated (becoming “vertical” in the process)
3. Attempt to initiate k − 1 horizontal links with other nodes belonging to the same layer and advertising a spare connection.
If the network keeps growing, other layers will gradually form on top of each other, but without adding too much workload for their “elders.” In order to compensate for the small increase in traffic, a simple rewarding scheme could be devised whereby nodes would obtain services at an incremental discount as the surface keeps moving further away from them. Indeed, as the hypergrid’s size grows faster than the workload on nodes, and considering the fact that the very principle of distributed computing is about sharing resources, “living deeper” could actually become highly beneficial. It could even solve the problem of replacing departing members, as their former “children” would likely compete for the privilege of replacing them, initiating a cascade of inward migrations and restoring the system’s integrity.

Finally, the hypergrid appears more resilient to directed attack than its scale-free counterpart. As already mentioned, one of the drawbacks of today’s networks is that they are usually very brittle if the attacker knows which nodes to target. The scale-free system we used in our control experiment is already much more robust though, as we force each vertex to initiate one backup connection, in order to reach the appropriate number of edges. Yet even so, the hypergrid performs much better in terms of being able to sustain significant damage without suffering major consequences. As shown in Figure 3(a), removing the 1% busiest nodes from the intact network has a considerable effect on pathlength distribution in the scale free architecture, while the difference is hardly noticeable in the hypergrid [Figure 3(b)]. Moreover, the redirected traffic is homogeneously distributed, resulting in the workload on surviving nodes being virtually unchanged (average ratio after/before attack is -1.02, with a maximum of -1.41) unlike in the scale free network (average ratio -1.55, maximum -6.84). This of course is a predictable side effect of the intrinsically less centralized topology, yet it is nonetheless a very useful feature as security and dependability issues are a growing source of concern (for more detailed results, see Saffre [7]).

In short, the hypergrid is a design philosophy for distributed systems that

- Uses only simple local connection rules
- Eliminates scalability problems
- Is intrinsically resilient to random failure and directed damage.

Yet meeting the highly challenging requirements of dynamic architecture has a cost, and the hypergrid is no exception: the price to pay for increased plasticity is likely to be complex routing procedures. Indeed, reducing the workload on the core by using peripheral links implies the ability to identify which horizontal connection(s) to use in order to go from one node to the other. A quick look at Figure 1(b) is enough to realize that this can be pretty hard, and actually the problem of achieving reliable navigation in a small world is a well-known one [8].

**CONCLUSION**

The hypergrid actually seems a remarkably well-adapted topology for future networks, at least if they are to fulfill the many promises of seamless connectivity and on-demand resource allocation. Whether it will be recognized as a viable solution and implemented is an open question. However, supporting a novel design when it is still in development can considerably speed up the process of creating a standard, and in Information Technology, many examples demonstrate that gaining momentum is critical. At present, the hypergrid is only one of several options and admittedly requires considerable refinement. Yet we remain convinced that it has a huge potential in what may become the next revolution in distributed computing: decentralized management of dynamic interactions.
REFERENCES