

# Organisation and Coordination for On-Line Routing in Communications Networks

Steven Willmott\*, Boi Faltings, Christian Frei, Monique Calisti,

Laboratoire d'Intelligence Artificielle,  
École Polytechnique Fédérale de Lausanne,  
Switzerland.

## 1. Introduction

This chapter considers the problem of routing communication demands in high speed networks which support Quality of Service constraints. Routing in the new generation of high speed networks is a particularly thorny problem: the large amount of information required for routing decisions is inherently distributed and may change rapidly over time, guarantees of service quality for every connection need to be met, many connections need to share the same resources and routes need to be found quickly to provide “dial-tone” connectivity to customers. These criteria make the task of distributed resource management and allocation very complex.

Techniques developed in Distributed Artificial Intelligence (DAI) provide ways of managing distributed systems using loosely coupled local problem solvers. Agents<sup>1</sup> can be used to provide autonomous local control whilst cooperating with each other to build more global views when necessary. Loosely coupled agent systems can also flexibly deal with changes in the state of the network and act pre-emptively to avoid network emergencies. This chapter has two main aims:

1. To discuss the pertinent features of the domain with respect to current DAI research on organisations and cooperation.
2. To present an approach based on environment adaptive dynamic organisations of agents which begins to address some of the outstanding issues.

Section 2 presents the routing problem, its difficulty and some discussion as to why DAI techniques may be applicable. Section 3 discusses specific domain features of the routing task and how they relate to the application of current DAI work, Section 3 also includes an overview of previous work on routing. Section 4 presents a method for dynamically partitioning a network and how to apply this to support a community of agents in performing the routing task. Section 5 concludes the chapter.

---

\* Corresponding author: e-mail - [willmott@lia.di.epfl.ch](mailto:willmott@lia.di.epfl.ch).

<sup>1</sup>For the purposes of this chapter we borrow from (Jennings, Sycara, Wooldridge, 1998) and define an agent as *situated*, *autonomous* and *flexible* (*responsive*, *pro-active* and *social*). See (Jennings, Sycara, Wooldridge, 1998) for definitions of these terms.

## 2. A routing problem

Due to the large number of existing network technologies and the multitude of different uses they are put to, routing comes in many forms - the basic task however involves finding paths for communication demands in a network. This chapter focuses primarily on the *on-line state-based source routing* needed in large high speed connection oriented networks.

- *On-line* means routes are found as connection demands arise as opposed to having routes pre-computed before they are required.
- *State-based* means that route choice depends at least partly upon the current state of the network as opposed to purely heuristic or statistical data.
- *Source* means that the node at which a demand originates is responsible for finding a route to satisfy it as opposed to having an entity such as a central route server generating all routes. In practice the meaning of source routing is often extended to allow several nodes to take part in route finding (e.g. see PNNI (ATM Forum, 1996)).

The need to provide Quality of Service(QoS) guarantees in today's networks is a driving factor behind the need for on-line state based source routing. To make QoS guarantees for a connection, a route through the network must be found which supports the specified level of service requested. Finding such a route requires a large amount of information about the network state which is both distributed throughout the network and rapidly changing. Alles (1995) suggests two main reasons for using source routing in the ATM PNNI routing framework:

1. If hop-by-hop routing was used, each node would need to evaluate QoS across the entire network to determine the next hop.
2. Hop-by-hop routing also requires a standard (identical) route determination algorithm at each hop to (help) preclude the danger of looping.

The utility of on-line routing is also supported by the success of dynamic routing strategies (such as RTNR (Ash et al, 1991) and DNHR (Ash and Huang, 1993)) in telephone and circuit switched networks. Lee et. al. suggest that traditional routing strategies based on the optimisation of a metric can no longer be used for today's high speed networks - the routing process "needs to be adaptive to the network state" (Lee, Hluchyj, Humblet, 1995).

Intuitively as the dynamicity of the network increases complete off-line solutions will stay valid for shorter and shorter periods. Since routed demands need to share resources

and a certain level of service has been guaranteed to each, new demands can only be accommodated with some reference to the current network state. The need for this information means that routes can only be allocated on-line.

## 2.1 Generic problem definition

The following explanation gives a technology independent definition of the problem being addressed here:

*Given a network graph  $G=(N,L)$  where  $N$  is a set of nodes connected by the set of links  $L$ , for any demand arising at a node  $n_1$  for a connection to any other node  $n_2$  with a vector  $\mathbf{b}$  of Quality of Service parameters - perform the following actions:*

- Find a suitable route between  $n_1$  and  $n_2$  which fulfils the demand's requirements (the vector  $\mathbf{b}$ ).
- If no route is available - reject the call.
- If a route is found - reserve sufficient resources along the route for the call.
- Hold the reserved route and maintain the QoS levels specified in  $\mathbf{b}$  until the call is terminated.

The main requirement for the use of this definition is having a connection oriented network with soft connection set-up - i.e. a way of establishing a new call in the network once the route has been chosen. This definition can be usefully applied to (among others) ATM<sup>2</sup>, SONET, TDM, SDH networks.<sup>3</sup> There are various extensions to the generic problem such as finding the "best" route for each demand by some optimality criterion or maximising profit over time (given that each accepted call generates a known amount of revenue).

One of the key things to note about the definition above is that this is not just a route finding process, it also includes the reservation of resources. The reason for this is that for on-line routing and routing in high speed networks the resource reservation or connection set-up and routing are often intertwined. The connection set-up needs to be made before the network state changes and the route chosen becomes in-valid.

## 2.2 How difficult is it?

Armed with our problem definition the next step is to examine how difficult the problem really is to solve. Finding a single path between two nodes in a graph can be done using a simple shortest path algorithm such as Dijkstra's (Dijkstra, 1959) or the Bellman-Ford algorithm (Ford and Fulkerson, 1962)<sup>4</sup>. These algorithms run in low order polynomial time<sup>5</sup>, however several things complicate the routing task:

---

<sup>2</sup>Note that the SVC service (soft connection set-up) for ATM has yet to be standardised.

<sup>3</sup>For more information about these networks types see (de Pryker, 1995) for ATM, (Kazovsky, Benedetto, Willner, 1996) for TDM, (Sexton and Reid, 1997) for ATM, SDH and SONET.

<sup>4</sup> See (Cherkassky, Goldberg, Radzik, 1994) for an evaluation of various shortest path algorithms.

<sup>5</sup>The standard form of Dijkstra's algorithm requires  $O(n^2) + O(m)$  where  $n$  is the number of nodes in the network and  $m$  is the number of links.

1. Wang and Crowcroft (1996) show that the metrics used in path selection impact on the complexity of the task. Using two or more “additive” or “multiplicative” metrics results in an NP-complete problem (See (Wang and Crowcroft, 1996) for an explanation of the different types of metric). ATM for example can specify: bandwidth, transfer delay, delay jitter and cell loss ratio which gives just such an NP-complete mix.
  
2. The solution algorithms that might be used to solve these problems require *all the problem information to be in one place* - that is the network topology, the link state parameters and the connection request details must all be available to a single processor. Although there are distributed versions of some shortest path algorithms<sup>6</sup> these are based on hop-by-hop table based routing schemes. These algorithms work at a slower time scale - propagating distance values to build up routing tables and are not designed to be used for routing single demands.

The second of these complications is perhaps the most serious since the efficiency of any on-line state based source routing mechanism depends critically on accessing the required link-state information which is distributed throughout the network. The metrics used are to some extent dependent on the network technology in use but in general one can expect the problem to be NP-complete.

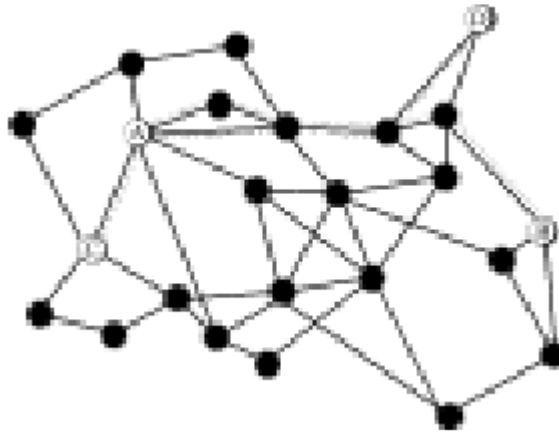
Related problems of choosing optimally which connections to accept onto a link with respect to a cost function have been shown to be NP-complete by analogy to the knapsack problem in (Vedantham and Iyengar, 1998). Off-line calculations to route all the demands in a network are generally NP-complete by analogy with the Multi-Commodity flow problem (Garey and Johnson, 1979).

### 2.3 The need for agents

Most of the difficulties associated with the routing task are due to the harsh environment in which it needs to be performed. Figure 2.1 shows a network graph with demands between nodes A and B and nodes C and D. There are clearly a large number of possible routes for each. Given that some links may be heavily congested an acceptable route may be considerably longer than the shortest path. The link state information on many of the links in the network is therefore important for making the route choice - the region of interesting links is very difficult to bound. Once a route has been chosen it needs to be set up (allocated) in the network. This set-up itself is problematic since there can be subtle and complex interactions between connections being established at the same time (between C-D and A-B in Figure 2.1 for example).

---

<sup>6</sup>Most notably distributed Bellman-Ford which is used in distance vector Internet protocols such as RIP (RFC 1058).



**Figure 2.1:** Network graph with communication demands between A and B (dotted line), C and D (dashed line).

There is a strong need for local control to spread the enormous information and work load in a large network. Local autonomy is vital not only to spread workload but also for robustness - local controllers should be able to continue solving local problems even if there are failures elsewhere in the network. Despite the need for locality some tasks will always need more than just local information (see section 3.1.2 for discussion on setting up a connection spanning several local regions for example). Tight control and management of resources crucial. To support both these requirements communication is essential for the exchange of information between local controllers and its functioning must be succinct and efficient since the communication medium for control messages is often shared with customer message traffic. Any routing system must perform well in the face of changing conditions in the network and must be *pro-active* rather than *reactive* - problems need to be avoided *before* they occur. (See (Lin and Yee, 1993) for a discussion on why the large volumes of data in high speed networks make feedback/reactive systems unsuitable).

Techniques developed in the DAI community provide ways of addressing these (sometimes conflicting) needs. Agents can be used to provide autonomous local control whilst cooperating with each other to build more global views when necessary. Spatial distribution allows information hiding between agents and the potential for passing summarised information to neighbours rather than raw data (Durfee and Lesser, 1991) thus greatly reducing necessary communication bandwidth. When compared with distributed algorithms, loosely coupled agent systems are also likely to be more adaptive and flexible to the changing state of the network and able to act pre-emptively in difficult situations (Gasser, 1992) such as overloads. Large networks may also need to be broken up into more manageable pieces whilst still supporting interactions between these pieces. The need for distributed hierarchical control for network management applications is widely accepted (Siegl and Trausmuth, 1996), (Somers, 1996). This together with the widely accepted need for local autonomy (and hence coordination and communication) and pro-activity represents a strong argument for the use of agent technology in network management and in routing.

### 3. Applying current work in distributed artificial intelligence

Although there are many complex issues in routing, the requirements of the problem essentially boil down to two key needs:

1. The distribution and management of network state information. I.e. the information needed for making routing decisions.
2. The distribution of control and decision making. I.e. the route finding process given a distributed information structure.

The principle area of research in DAI which relates to these two needs is work on coordination and cooperation. Jennings (1996) identifies three main possible reasons for requiring coordination between distributed problem solvers:

- Dependencies between actions.
- The existence of global constraints.
- No individual having sufficient information or resources to perform all tasks.

All three of these appear in this problem domain: resource availability defines global constraints, routing individual demands or parts of demands impacts the choice of other routes (dependencies between actions) and in large networks the impossibility of any agent holding a global view requires all agents to have partial and incomplete information. In the rest of this chapter we make the distinction between *coordination for tasks* and *organisation*. Organisation is often seen as a type of coordination approach, however we separate it out here to emphasise the fact that organisation is more concerned with the long term make up and structure of a community of agents. Most work on coordination is concerned with ensuring coherent action between groups of agents to achieve a *single goal* or complete an *a-priori well defined set of goals*. Organisations provide structures in which agents are able to coordinate more effectively by specifying the actions individuals will take in given circumstances. The following review gives a flavour of the work in these two areas

Coordination for a single task or set of tasks has received a lot of research attention. One strong theme is that coherent joint action is underpinned by mental constructs such as team goals and joint intentions ((Grosz, 1996), (Cohen and Levesque, 1990)). This thinking has led to the development of agent architectures which incorporate explicit representations of cooperation. Best known among these are GRATE\* (Jennings, 1995) and STEAM (Tambe, 1997). Negotiation is also often used to allow groups of agents to come to agreements on actions when they have conflicting goals (See (Laasri et al, 1992) for a good review). Extensive work by Durfee, Lesser and others on Partial Global Planning ((Durfee and Lesser, 1991), (Decker and Lesser, 1992)) and the Distributed Vehicle Monitoring

Testbed (Lesser, Corkill, Durfee, 1987) focuses on the use of meta level communication for coordination.

The theory of organisations has also lead to a lot of research in DAI. Gasser (1992) defines organisations as “patterns of information and control relationships between individuals”, Werner (1989) uses the concept of roles to define organisations. Jennings and Castelfranchi respectively characterise an agent’s decision to join an organisation (or to fill a role) as making a high level commitment (Jennings, 1996) or an organisational commitment (Castelfranchi, 1995). Corkill and Lesser (1983), in one of the first applications of organisation in DAI, present organisation as externally imposed coordination. (Fox, 1981) and (Malone, 1988) both present very useful management science perspectives on computational organisations. Markets are often seen as a organisational technique and there has been significant work in this area<sup>7</sup> - see (Clearwater, 1996) for a good collection of papers on this approach. Other work in the area of organisation includes (Shoham and Tennenholtz, 1992), (Prasad, 1996) and much of the work on Partial Global Planning (mentioned above). Work on adaptive organisations (Organisational Self Design (Corkill, 1983), (Ishida, Gasser, Yokoo, 1992), logical reorganisation (Guichard and Ayel, 1994) and (Durfee and So, 1997), (Decker and Lesser, 1995)) attempts to model organisational knowledge and allow an organisation to change dynamically over time.

### 3.1 The network routing domain

This large body of DAI work (the above is only a very brief overview) has been applied to a number of different problem domains. As with any application domain, working on the problem defined in Section 2 holds particular challenges for the application of DAI techniques. The following domain features all need to be considered when designing an agent based system for the routing task:

**Vast numbers of agents:** Communication networks can be very large and this forces a corresponding increase in size in any distributed control structure. Most previous industrial applications require relatively small numbers of agents (In the range of 3 - 10 - (Jennings, 1995), (Varga, Jennings, Cockburn, 1992), (Durfee and Lesser, 1991)). Only a few systems have been tested with larger numbers of agents ((Cammarata, McArthur, Steeb, 1983) is one). This means that many of the coordination mechanisms still need to be tested on a large scale.

**Spatial distribution:** Although some systems (notably work on DVMT (Lesser, Corkill, Durfee, 1987) and others) use spatial decompositions to divide up work, most systems focus on functional decompositions (GRATE\* (Jennings, 1995), STEAM (Tambe, 1997) to some extent, TAEMS (Decker, 1996)). Spatial distribution typically 1) increases the importance of communication cost during problem solving (see below) and 2) restricts the type of possible relationships

---

<sup>7</sup>Note, markets are also often applied to coordinate agents for a single task.

between agents - agents may only be able to communicate with their direct neighbours for example.

**Communication problems:** A lot of previous research does take into account communication cost in a simple form by adding fixed time penalties for messages sent ((Jennings, 1995), (Kuwabara, 1996)) or simply counting the number of messages (Tambe, 1997). This is insufficient for networks - taking into account the agent communication medium becomes much more important since this is also the medium being controlled. The cost to two agents far apart in the network which need to communicate is in general much higher than that incurred by neighbouring agents. Unreliable links and large variations in message send time are also potentially serious communication problems (These have been shown to cause difficulties with confirmation based protocols). Theoretical work on the utility of communication such as (Gymtrasiewicz, Durfee, Wehe, 1991) is unlikely to be applicable since most messages will be essential.

**Small problems, fast responses:** In general the routing of a single call is a small task and response times in the network need to be fast to satisfy users. This causes problems for coordination schemes which:

- Require long set-up periods before beginning problem solving. Long initial coordination periods are generally only justified if the problem solving process *itself* will be long and expensive.
- Require a potentially large number of iterations to complete. Since the communication cost is a large factor in the overall cost of problem solving communication intensive schemes will suffer.

Although there is some work on fast coordination ((Decker and Lesser, 1993), (Kosoresow, 1993), (Kraus and Wilkenfeld, 1991) and (Kraus, Wilkenfeld, Zlotkin, 1995)) it is often very theoretical and difficult to apply. The scheme in (Kosoresow, 1993) for example requires all agents to simultaneously exchange task lists to enable each to generate bids for task assignments. In general the need for fast responses to relatively simple problems suggests that an organisational structure to support coordination would be beneficial.

**Interactions:** As stated above, routing an individual call is considered a relatively small task. There are however many of these small tasks arising continuously, these interact with one another to produce a dynamic and complex problem solving environment. This flow of tasks cannot readily be modelled as “one big” problem to be solved since the tasks arrive asynchronously and in a distributed fashion (and are not known in advance). The harsh time constraints in the network mean that synchronisation (waiting for several calls and solving a harder joint allocation) is not desirable. Interactions occur at several levels:

- At the domain resource level: An agent may be participating in routing two demands each with different partners. The agent's choices for one allocation could well affect the remaining options for the other.
- At the reasoning resource level: When many demands are actively being routed in the network, an agent is likely to be involved in various different cooperative operations at once (with different colleagues), since the agent's resources are limited; this means that it is hard to guarantee when tasks will be finished.

The effect of outside commitments on problem solving has been taken into account in previous work (Jennings, 1995) but usually this is treated as exceptional, in network applications this is likely to be the rule and is a vital consideration.

**Changing community makeup:** Nodes and subnetworks can appear and disappear in a network and these changes must be automatically dealt with by the system<sup>8</sup>. These domain level changes in turn cause changes in the makeup of the control architecture. Changes in demand patterns can also favour changes in the number, type and distribution of agents in the control architecture. Work on adaptive organisations is relevant to this point and is discussed further in Section 4.3.

**Managing Scope:** Systems may have very large numbers of agents, very few of which are implicated in performing any particular task. Selecting which agents should be included in any problem solving group, or even choosing who to notify that the task exists can be a hard problem. This tends to be somewhat easier in systems using a functional decomposition than in spatially decomposed systems. In GRATE\* (Jennings, 1995) for example a team leader selects which agents should take part in problem solving (participants are can be selected for their functional capabilities). In spatially decomposed systems it may be possible to use some kind of distance metric but this is arbitrary at best. Knowing the "scope" of a problem is in fact *part of the routing problem* to be solved. (Durfee and Montgomery, 1990) presents a way for agents to use abstract summary information to find partners for problem solving, however this also relies on an initial broadcast to the whole community.

The aim behind highlighting these issues is not to say that current frameworks cannot handle them but that these concerns are not their immediate focus. To maintain emphasis on key issues we return to the questions of managing information and control flow.

### 3.1.1 Information management

Representing and partitioning domain level information well is important in the design of any routing protocol and essential when addressing on-line state-based source routing. Achieving good division of information is one of the principle aims in applying agent based

---

<sup>8</sup>Although the probability of such changes may be very low, the fact that the routing system is intended to stay on-line for long periods of time means that the probability of these events occurring rises.

systems to the routing task since agents have the potential for passing summarised information to neighbours rather than raw data.

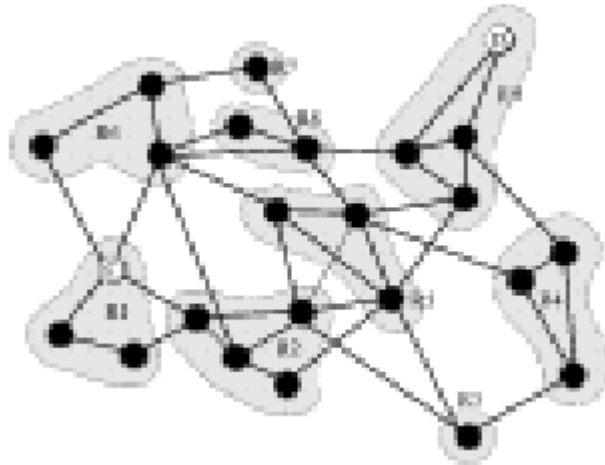
To partition the network state information agents each need to have a local view of the network. The scoping and communication issues together mean that there is no scope for broadcasting network state information. Broadcasting state information would not only overload communication channels (which are most likely shared with customers) but also swamp all the agents in the system with (mainly useless) information.

Since agents have local views of the world and global broadcast is not an option, access to crucial information which is outside an agent's scope must be supported by a clear scheme for knowing *where* a piece of information is represented. This is essential both in facilitating fast access to information and in screening out unnecessary information not required for the current task.

### 3.1.2 Control flow

A second important potential gain of applying agent systems to routing is in managing the flow of decision making. Control flow needs to be as local as possible: if a task can be performed by a small group of agents, letting other agents even know about the existence of the task should be avoided (controlling scope). Control needs to be flexible but clearly structured: each task may involve different coordinating partners but there needs to be a direct and well defined problem solving process since most problems are small and should be solved with a minimal coordination overhead. The number of iterations (exchanges of messages) in the process should be kept low. The number of exchanges between cooperating agents increases the amount of *synchronisation* in the problem solving process which is known to increase the influence of communication cost on overall problem solving time (Bertsekas and Tsitsiklis, 1989). Not only do a large number of iterations increase communication cost, they also increase the possibility that problem solving is interrupted due to a communication failure. To perform local actions agents need to be empowered to make decisions which could affect the global network state. Responsibility for decisions in the network needs to be distributed to allow more informed agents to take control of decisions which could have globally more serious effects.

A relevant feature of the routing problem is the fact that problem solving is in general quite serial or synchronised. Figure 3.1 depicts the same example as Figure 2.1 with the network divided up into distinct regions. Each of the small regions of the network is considered to be controlled by a separate controller agent and the task is to route a demand from C to D.



**Figure 3.1:** Network topology with nodes clustered together into small groups.

Since each region is autonomous each decides its own internal route and these part routes are assembled to form a complete path. However, the choices of internal path are highly interdependent since they depend upon the entry and exit points of the part route to each region. If for example (by some process) it had been established that the demand C to D would pass through regions  $R1^{(\text{start } C)}$ , R2, R5,  $R9^{(\text{end } D)}$ , region R5 would not be able to decide on an internal route without knowing which link between R2 and R5 is to be used. At a simple level there are two principal ways of resolving this: serial search - letting R1 choose a route then R2, then R5 etc. Or by coordinating the problem more globally and selecting the connecting routes between regions first then letting R1, R2, R5 and R9 complete their part routes in parallel. In the general case the problem also includes the selection of which regions to pass through. The key thing to note about this example is the interdependency between the local decisions which need to be made.

### 3.2 Current agent based approaches to routing problems

There is very little agent based work which directly addresses the on-line source routing problem, this section therefore covers the work which is closest and could possibly be adapted.

There have been several market based approaches to routing problems. These mostly address the problem of routing several demands at once given a set network topology ((Wellman, 1992), (Wellman, 1994), (Kuwabara et al, 1996))<sup>9</sup>. Gibney and Jennings (1998) describe a market based approach for selecting between a limited number of pre-set available routes. Yamaki, Wellman and Ishida (1996) and Hayzelden and Bingham (1998) describe systems for managing VPC topologies and bandwidth allocations according to demand and need in ATM networks. (Yamaki, Wellman, Ishida, 1996) uses a market based architecture and (Hayzelden and Bingham, 1998) applies ideas from Brooks' subsumption architecture to develop a functional decomposition.

<sup>9</sup>Note that the work by Wellman in (Wellman, 1992) and (Wellman, 1994) uses the example of transport planning but this is directly analogous to routing in networks.

Appleby and Steward (1994)<sup>10</sup> describe a system based on ant-like mobile agents able to control routing in a network by updating routing tables. Two different approaches to circuit restoral based on spatial distribution are presented in (Conry, Meyer, Searlman, 1988), (Conry et al, 1991) and (Clark, Grossner, Radhakrishan, 1996) . Clark et. al. apply distributed planning and Conry and colleagues use an approach based upon a generalisation of the contract net which formulates the problem in a way similar to a distributed constraint satisfaction problem. The service restoral problem is again a little different from on-line routing since several demands (all circuits on the link that failed) need to be routed at the same time.

There are two main recurring difficulties to applying these approaches to the on-line source routing problem:

**Use of broadcasts:** All of the market based approaches rely on the use of price broadcasts to distribute information, whereas (Conry, Meyer, Searlman, 1988), (Conry et al, 1991) and (Clark, Grossner, Radhakrishan, 1996) broadcast plan and negotiation information. This information flooding is both very costly and has the problems associated with limiting the scope of problem solving (see Section 3.1). This cannot be supported in large networks.

**Synchronising aspects:** The market institutions in the markets based approaches tend to have a centralising effect on problem solving (although (Wellman, 1994) introduces “arbitrageur” agents to reduce this problem), as does the planning mechanism in (Hayzelden and Bingham, 1998). The market types used often also use timed periods for problem solving (waiting for bids to complete the round), this can have highly synchronising effect on problem solving which is undesirable for the on line problem.<sup>11</sup>

Both of these points illustrate that organisations are required to control both who receives information and how decisions are made. The HYBRID system presented in (Somers, 1996) and Chapter 3 (Evans et al, 1999) in this volume is one of the few agent approaches to network management problems which focuses on the use of hierarchies to control scoping effects (Unfortunately the work on HYBRID work does not address routing per se).

### 3.3 Non agent based approaches to the routing

Although there is a vast amount of literature on many varying routing problems<sup>12</sup> the question of how to tackle routing in high speed QoS networks is still very much open. The ATM Forum for example is proposing a whole new architecture (PNNI (ATM Forum,

---

<sup>10</sup>Note, this article is re-printed in this volume, chapter 11 (Appleby and Steward, 1999).

<sup>11</sup>Note however as mentioned above most of these systems were originally designed for problems allocating groups of demands where centralising and synchronisation are more acceptable.

<sup>12</sup>See (Bertsekas and Gallager, 1992) and (Perlman, 1992) for good surveys of various routing problems and algorithms.

1996)) to provide routing services in ATM since much of the previous work on routing was difficult to apply (Alles, 1995). See (Lee, Hluchyj, Humblet, 1995) for discussion on the implications of QoS provision for the type of routing scheme required.

There have been advances using non-linear combinatorial optimisation (Lin and Yee, 1993) and competitive analysis (Plotkin, 1995) but neither of these seems likely to lead to a complete solution - they do not address how to pass around network state data for example. Cidon et al (1995) address this by using specialised hardware switching techniques and maintaining a broadcast tree to give every node complete network state information. The applicability of this type approach is limited by the size of the network and the dynamicity of the network state, rapidly changing network state dramatically increases the number of broadcasts needed.

Perhaps the most detailed and extensive work on this subject was done under the banner of ATM as the PNNI standard (ATM Forum, 1996) which describes an extensive architecture to handle many aspects of routing. The primary interest of PNNI here is as a framework for the handling of network state information and signalling to support control decisions. PNNI could readily be implemented as an agent based system (many provisions in the standard can directly be described in terms of agent structures). PNNI is discussed further in Section 4.2.4.

## **4. Adaptive control structures**

Whilst bearing in mind the problems outlined in Section 3.1 this section outlines the design of an agent based control system for the routing task. The key idea behind the approach presented here is the use of an organisational structure which adapts to network resource availability. Within the scope of this organisation agents work and cooperate to solve individual routing problems. The scheme has two major components:

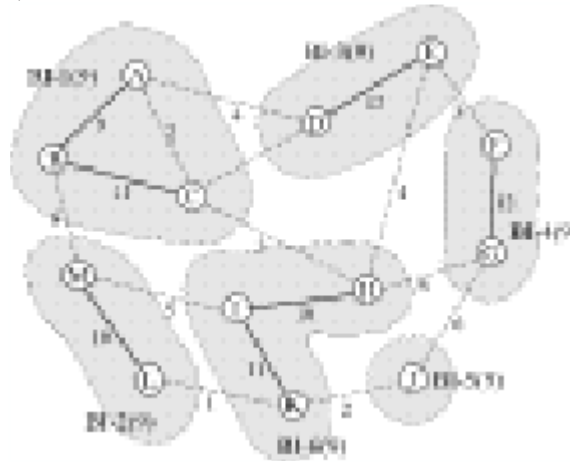
1. A partitioning scheme which summarises availability at various levels of abstraction (Section 4.1).
2. An agent control organisation based upon this information representation (Section 4.2).

Sections 4.2.4 and 4.2.5 go on to discuss how this approach addresses some of the points made in Section 3.1. Section 4.3 discusses the relationship of this work to research on adaptive organisations, organisation and cooperation. In terms of implementation we currently have a centralised tool using the partitioning techniques for resource allocation/routing and the distributed approach presented here is work in progress.

### **4.1 Blocking Islands and space partitioning**

(Frei and Faltings, 1998) introduces a clustering scheme based on Blocking Islands<sup>13</sup> which can be used to represent network resource availability (here specifically bandwidth) information at different levels of abstraction. This section outlines the basic idea (more details can be found in (Frei and Faltings, 1998)).<sup>14</sup>

A blocking island is a group of nodes in the network graph which are connected by links with a specified minimum available bandwidth. The grey areas in Figure 4.1 below are all blocking islands at level 9 Megabits per second (Mb from here on). The grey area marked BI-1(9)<sup>15</sup> for example contains A, B and C which means that there is a path between any two of {A, B, C} at 9Mb or above (note that A and C are not directly connected at this bandwidth level but have a common route through B). The region BI-1(9) also represents the fact that no other node in the network graph can be reached at a bandwidth of 9Mb from A, B or C - the nodes therefore form an isolated group at this level.



**Figure 4.1:** The map of blocking islands at 9Mb shows nodes clustered together into equivalence classes which are reachable at a bandwidth of 9Mb. The small numbers by the links represent remaining capacity, dashed links have less than 9Mb free, solid links of 9Mb or more.

Given any bandwidth requirement blocking islands partition the network into equivalence classes of nodes which are reachable by links of that amount of available bandwidth. The regions are *unique, identify bottlenecks* (inter-regional links<sup>16</sup>) and highlight the *existence and location of routes* at a given bandwidth level. If two nodes are clustered in the same blocking island at a given bandwidth level there *must exist a route between them* - furthermore all links which could form part of the path *lie inside this blocking island*.

<sup>13</sup>Please note that there is a patent pending on the clustering techniques and their applications.

<sup>14</sup>The partitioning scheme as presented here has only been applied to systems using traffic types which have a constant bit rate (bandwidth) requirement. We are currently developing extensions to apply it to complex traffic types such as the ATM VBR service which allow more advanced multiplexing.

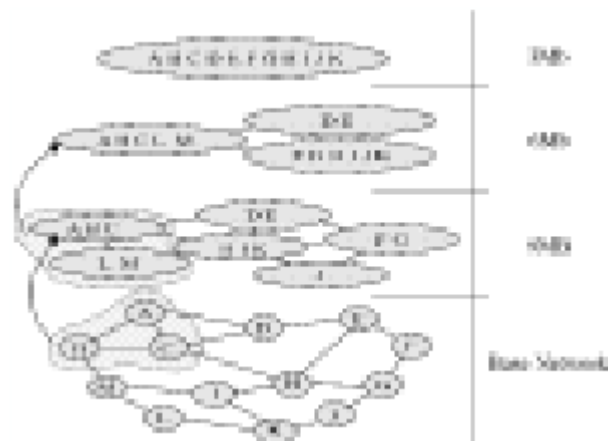
<sup>15</sup>The notation BI-x(y) represents “blocking island number x at bandwidth level y”.

<sup>16</sup>This is more than just measuring which links have low capacity remaining since there can be links with little remaining bandwidth inside blocking islands as well - inter-regional links are links with low remaining capacity for which *there is no alternative route*.



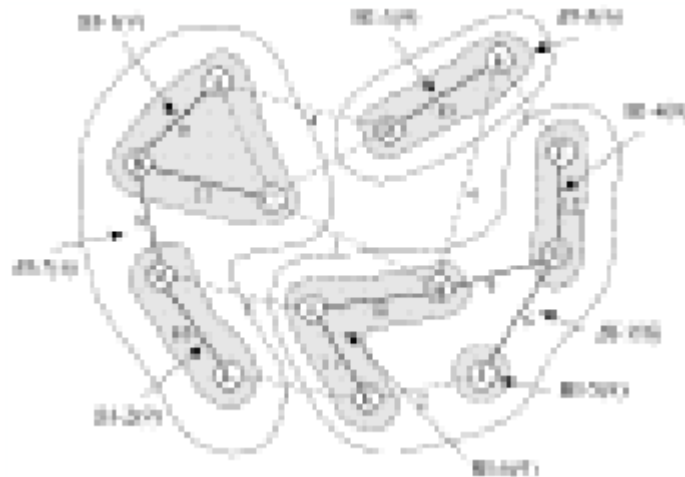
**Figure 4.2:** The abstract representation of the network shown in Figure 4.1 shows which nodes are clustered together and inter-region connectivity. This abstract representation is called the blocking island graph in (Frei and Faltings, 1998).

The regions in Figure 4.1 can be used to construct an abstract graph of the network showing which nodes are reachable at a given bandwidth level (see Figure 4.2). If abstract graphs are constructed for various bandwidth levels the resulting graphs can be combined to form a *hierarchy of blocking island graphs* representing the network at different levels of abstraction. Figure 4.3 shows an example of such a hierarchy for bandwidth levels: 9, 6 and 3Mbs. This hierarchy can be constructed in time  $O(bm)$  where  $b$  equals the number of bandwidth requirements and  $m$  equals number of links in the network.



**Figure 4.3:** A hierarchy of blocking islands. Each level clusters nodes recursively.

There is a useful relationship between levels in the hierarchy which is called the *inclusion property* in (Frei and Faltings, 1998). Given two blocking islands which cluster a common node at different levels in the hierarchy (see Figure 4.4, for example where the node G is both in the region BI-4(9) and BI-9(6)), the inclusion property states that the BI clustering at the higher bandwidth level (9Mb) may not include *any node not clustered at the lower bandwidth level* (6Mb).



**Figure 4.4:** Two levels of blocking islands, the 9Mb level partitioning from Figure 4.1 is now summarized again at the 6Mb level.

The inclusion property is transitive between levels and leads to “complete” inclusions as shown in Figure 4.4: each of the blocking islands at the 6Mb level exactly clusters a set of blocking islands from the level below (the 9Mb level) - none of the 9Mb blocking islands overlap the boundaries of their parent blocking island. The inclusion property thus insures that the hierarchy is a tree. Each blocking island level can be seen as an abstract graph like the one in Figure 4.2, this is repeated at all levels in the structure and produces the type of hierarchy shown in Figure 4.3.

Since the graphs and the hierarchy depend upon the bandwidth available on the links they need to be updated over time as link states change. These changes can lead to merging and splitting between blocking islands. In Figure 4.4 for example freeing up an additional 3Mb or more of bandwidth between nodes H and G would merge islands BI-6(9) and BI-4(9) to cluster the nodes K, I, H, G and F in the same island. Then using up 2 or more Mb of bandwidth between I and H would split the new large island into two: {I, K} and {H, G, F}. The updates for this merging and splitting can be managed in time linearly proportional to the number of links in the network.

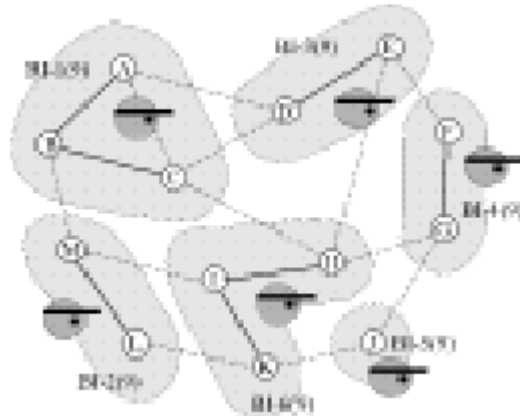
## 4.2 How agents run blocking islands

There are various ways of using the partitioning methods above to build agent based control structures and information representations for routing. The hierarchical model described below is perhaps the most intuitive and illustrates the points we wish to make about the application of coordination techniques.

### 4.2.1 General scheme.

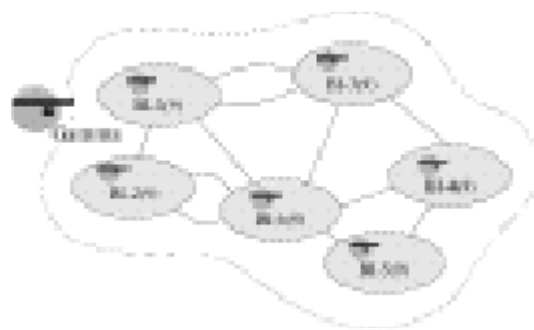
To build a simple control structure *one agent is assigned to manage each blocking island*. An agent controller at the lowest level of abstraction has knowledge of a

set of network nodes and links which form a graph. The controller also knows which level of the hierarchy it is in and where its parent agent can be found (if it has one). Figure 4.5 illustrates this collection of local views of the network for the example from the previous section (Figure 4.1).



**Figure 4.5:** Each blocking island is managed by an agent.

Figure 4.6 shows the knowledge kept by the controller of the region which clusters the *regions* in Figure 4.5. This is also a graph: links are the inter-region links from the level below<sup>17</sup> (the bottlenecks in fact) and nodes represent the blocking island regions from Figure 4.5. Gamma thus manages a region in the level hierarchically *above* the one shown in Figure 4.5. Each of the nodes is a compound node which is managed by an agent one level down in the hierarchy. The agent Gamma in Figure 4.6 in turn manages a compound node in the graph held by the agent clustering this region in the level above. Each of the child controllers shown in Figure 4.6 is responsible for links inside its own blocking island and Gamma is responsible for the inter-region links in its own domain.



**Figure 4.6:** The controller at the next level up has knowledge of a set of complex nodes and their interconnections. The controllers managing the smaller blocking islands are defined as the children of Gamma.

<sup>17</sup>In a variation links at higher levels can be clustered together into logical links.

As resource availability in the network changes blocking islands split and merge (as described in Section 4.1). This has a corresponding effect on the control architecture - adjacent agent controllers at every level can split and merge. If for example as in the previous section BI-6(9) and BI-4(9) where to merge the shape of the abstract graph managed by Gamma in Figure 4.6 would also need to be updated (Gamma would have one less child and manage fewer links). Correspondingly agent controllers need to update and move (logically or physically) to stay with their shifting domains. Estimating the cost of structural updates is difficult since they rely on the implementation of mobility and agent creation/destruction, the time required for updates is however linear w.r.t the number of links in the network.

#### 4.2.2 Finding a route

The control architecture mirrors the bandwidth availability in the network. Given the organisational structure the agents inside it now need to coordinate to find routes. Each agent at every level is responsible for the provision of a routing service. Agents at the lowest level (Figure 4.5) perform real routing tasks from point to point and agents at higher levels (Figure 4.6) perform abstract routing tasks. To find a route for a demand arising at a source node  $n_1$  for a target node  $n_2$  the agents use something like the following scheme<sup>18</sup>:

1. The demand is sent from the source to the agent (agent<sub>1</sub>) clustering  $n_1$  at the lowest level in the hierarchy (i.e. the blocking island which is the smallest region surrounding  $n_1$  and has the highest bandwidth connectivity). This agent now takes responsibility for routing the call.
2. If the target node ( $n_2$ ) is not in agent<sub>1</sub>'s domain it immediately knows it cannot route the demand. If the bandwidth requirement for the demand could be satisfied at the hierarchy level above then the demand is passed to the parent agent agent<sub>2</sub>, otherwise the demand can be rejected immediately. If the demand is passed to the parent, the parent becomes responsible for routing it and agent<sub>1</sub> maintains a responsibility to let the source  $n_1$  know of the eventual outcome.
3. If the node  $n_2$  is clustered in the domain of agent<sub>1</sub> the agent knows that there is at least one route between the target and the source with sufficient bandwidth for the connection. Agent<sub>1</sub> can now search among these routes for one which satisfies the other QoS constraints of the demand. If such a route is found agent<sub>1</sub> reserves the resources required, sets up the connection and informs the two participating nodes of success. If the other QoS requirements (delay etc.) could not be satisfied, agent<sub>1</sub> may be able to pass the demand up to its parent (presuming the bandwidth requirement would still be satisfied at the next level up). The parent has access to more possible routes and may be able to find a route which agent<sub>1</sub> could not see.

---

<sup>18</sup>For simplicity here we assume demands arising always have a bandwidth requirement equal to one of the levels in the blocking island hierarchy. This can easily be relaxed (Frei and Faltings, 1998).

The basic principle is very simple - if a node is not clustered at any one bandwidth level it gets passed on upwards. When demands arrive at a parent agent, the parent assumes responsibility for routing it and needs to find an abstract route:

1. When a parent (agent<sub>2</sub>) receives a demand from a child (agent<sub>1</sub>) it takes responsibility 1) to try and route the demand and 2) to inform the child of the outcome.
2. Agent<sub>2</sub> checks to see if any of its children cluster the target node (it knows the requesting agent agent<sub>1</sub> clusters the source node). If the node is not held by any of its children agent<sub>2</sub> passes the demand on upwards.
3. If the target is held by one of its children agent<sub>2</sub> chooses a route (or set of routes) in its own abstract graph and then coordinates affected child controllers in the level below to establish the route. To do this the parent sends *notifications for tender* on parts of its abstract route to the child agents involved at the level below by specifying the end points of the relevant segment of the path.
4. On receipt of the notification a child agent attempts to find a route between the end points specified. This either results in failure (the network state had changed or the QoS could not be satisfied) or the agent commits to servicing this demand and notifies the parent of the cost of the route found by sending a *bid*.
5. Once the parent (agent<sub>2</sub>) has received a sufficient number of bids to put together a whole route agent<sub>2</sub> matches the offers from child agents with the abstract route plan. Path constraints such as delay can now be checked for the complete route. Note that agent<sub>2</sub> does not know the whole route in detail - only the metric cost of each section.
6. If one of the possible routes is acceptable confirmations are sent to all the relevant child agents to confirm the route and an end-to-end set-up call makes the connection. Cancellation messages are sent to any other agents which had proposed part routes not used in the final solution.

This process is repeated at several levels in the hierarchy thus agents are often acting on behalf of a parent to service a demand and as consumers trying to allocate that same part route through regions controlled by their own child agents in the level below. This example control scheme is perhaps one of the simplest, there are other possibilities: allowing peer to peer negotiation inside a hierarchy level for example opens up other possible approaches.

#### 4.2.3 Information representation

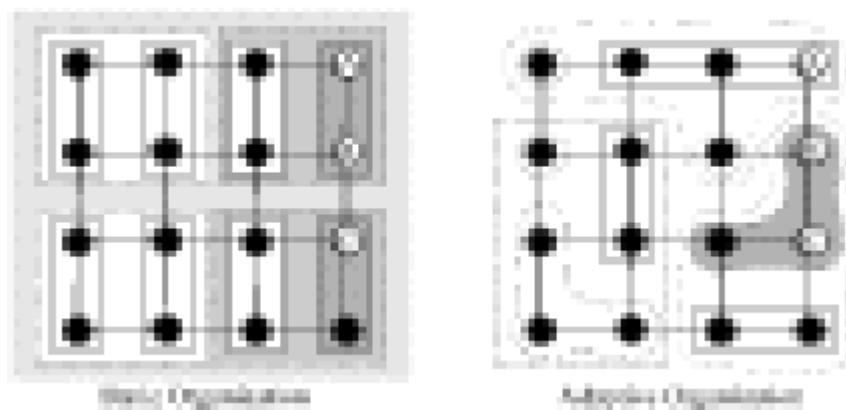
Even though the organisation is adaptive there is always clearly defined ownership of resources, the more critical a link is the higher up in the hierarchy it is managed. There is also a clear definition of where information can be found and accessed. The parent-child relationships define a gradually widening scope and increasingly abstract view of the system.

There is no periodic flooding of state information as in most common routing protocols. This is because decision making is spread out along the route, control is passed up to the level which has enough global (though abstract) information to coordinate route choice, the network state information is then accessed on site.

Perhaps the key benefit is that some of the state information necessary for problem solving is already implicit in the control structure. The existence of a route or the impossibility of connecting two nodes at a certain bandwidth are part of the organisation. The blocking island hierarchy also naturally represents a notion of how critical a link is. The higher a link is clustered the more of a bottleneck in the system it is. This is not just because it is congested but because there exists *no alternative high route with sufficient remaining bandwidth* between the blocking islands it connects. Some effective routing heuristics can be based upon this such as: using links clustered low down in the hierarchy or avoiding splitting blocking islands to preserve high bandwidth connectivity.

#### 4.2.4 Control flow

Each agent in the hierarchy has a clearly defined routing task. Problem solving starts at the lowest level in the hierarchy and problems are passed up to the level that has sufficient global information. The blocking island structure provides a more meaningful division of scope (Section 3.1) than a distance metric or a static organisation structure (such as PNNI) since the bandwidth connectivity is used to discriminate which agents may be involved in a potential solution.



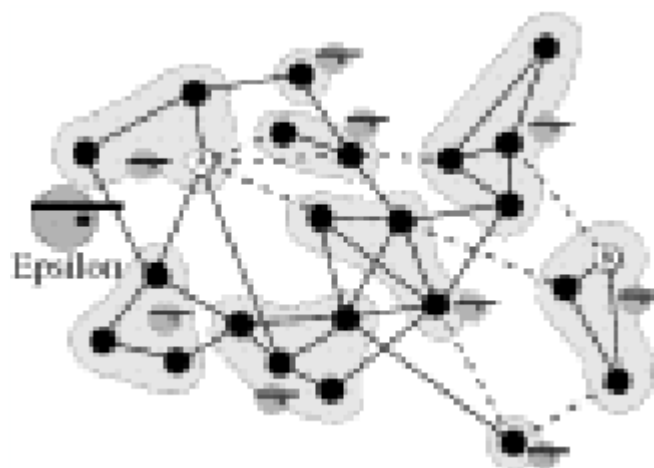
**Figure 4.7:** Example of a static organisation and a organisation adapted to bandwidth availability. The grey areas indicate which regions must be considered by each organisation to find an acceptable route between nodes B and C (the lighter the grey - the higher the level).

Figure 4.7 illustrates the difference between a bandwidth adaptive organisation (right) and a static organisation (left). Allocating a demand between neighbouring nodes B and C (which share high bandwidth connectivity) is quick in the adaptive organisation since the two nodes are clustered at the lowest level whereas in the static organisation the decision can only be made at the highest level of abstraction. This reflects the representation of the

ready availability of resources between B and C in the adaptive organisation which the static organisation does not capture. The situation is reversed for connections between A and B and the adaptive organisation only clusters these two nodes together at the top level of the hierarchy, whereas the static organisation can make a decision at the most local level. The extra effort required in the adaptive organisation reflects the fact that A and C are connected via links which are *resource critical* which may mean that they should be dealt with by an entity with a broader view of the network. Routing traffic on a critical link may have further consequences for the rest of the network (for instance it may unnecessarily disconnect two regions of the network).

Since bandwidth is only one possible QoS parameter, others have to be taken into account once “bandwidth acceptable” routes have been found. Using bandwidth as the first discriminating characteristic is often efficient because, as (Wang and Crowcroft, 1996) and (Lee, Hluchyj, Humblet, 1995) point out, bandwidth often affects the other QoS parameters (insufficient bandwidth means packet queuing inside routers for example, which increases delay).

Returning to the example from Section 3.1.2, Figure 4.8 shows the same network divisions as Figure 3.1. The search for a route between A and B is coordinated by an agent clustering all of the regions in the figure (in general there could be other levels of abstraction in between).



**Figure 4.8:** Epsilon proposes 4 abstract routes through the network for a demand between A and B.

The relationships in the hierarchy now allow the higher level controller Epsilon to find several abstract routes (4 in this case) and ask the regional controllers to complete the part routes internally. The answers of the child agents with their cost metrics are then synthesised to form a complete route. In terms of control flow the child agents are able to work in parallel<sup>19</sup> and the higher level controller is able to use a more global view of the problem to ensure that the whole path meets all the constraints (recall the discussion on the interdependence of local solutions from Section 3.1.2). This is likely to produce better

<sup>19</sup>Whereas in PNNI search is linear end-to-end.

routes than a serialised approach since it avoids problems with having to backtrack on partial routes (such as in crankback in PNNI). The parallel aspect also allows resource reservation to be interleaved with path planning. This reduces the danger of planning a path and then finding that the network state has changed before being able to set up some part of it. The cost is that problem solving can be slower - in lightly loaded networks where a linear search for a solution is normally correct at first guess the extra coordination is generally unnecessary. However in more heavily loaded networks the trade-off often pays, particularly if the connections routed are relatively long lived - taking more time to find a better route pays off.

#### 4.2.5 Weaknesses

Inevitably, aside from the potential advantages the scheme also has some problems and difficulties to overcome. Since the hierarchy is dynamic and nodes change domain membership it cannot itself be used to create an addressing scheme as PNNI can be. This also has an effect on the representation of reachability information: since the groups of nodes held by a blocking island are fluid and cross organisational boundaries it is harder to group them in a compact way (by prefix for example) to represent which nodes are held by whom. The adaptivity to the domain is in the form of a feedback loop (Figure 4.8). This has several benefits (outlined in Section 4.3) but can also create problems. If the influence of the environment on the control structure becomes too great the control structure will spend too much time trying to adapt rather than actually doing its job.

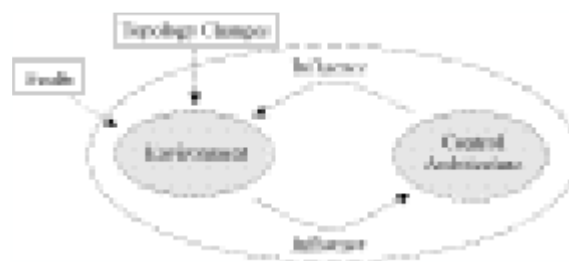
In certain network states the fixed levels may not give a very even covering for the control structure (i.e. in times of light load some agents will end up clustering large areas of the network). This can be addressed by the dynamic creation of levels in the hierarchy or the use of intervals over the bandwidth to define equivalence classes rather than fixed values. This issue still tends to mean that the scheme is most applicable in highly loaded networks. Arguably however this is also when it can be most useful since resources need to be well managed. In general networks are most profitably run at around 80% capacity to fill then links but allow for the foldover in traffic.

As pointed out in (So and Durfee, 1992) hierarchies of all types have a centralising effect in that ultimately there is a head agent at the top. However just as in (So and Durfee, 1992) losing the head agent does not stop the whole system from working - only demands needing to flow over the most bottlenecked links will suffer. At higher levels when blocking islands are large it might also be necessary to have several controller agents share the work for one domain. A final problem is with security and authority in a system which necessarily crosses organisational boundaries. In an environment with several service providers managing different parts of the network different companies are unlikely to want to share enough information to create a globally transparent routing system. In the worst case each company would use its own routing control system and negotiate connectivity at the boundaries.

### 4.3 Description and discussion

The structure of the agent society presented above is a hierarchy defined by parent-child relationships. These relationships are used to formalise the organisational knowledge within the agent society. Agents know their level, the levels of their children and their parents and how to communicate with them<sup>20</sup>. The possible roles in the system stay fixed but the agent occupants of these roles can change<sup>21</sup>, as can the relationship of a role with the physical environment (an agent can control different groups of nodes at different times, agents can have varying numbers of children, changing connectivity to their neighbours etc.). Agents in the organisation are committed to being *coordinated* - that is, wherever possible they undertake to act in concert with the other agents and fulfil their roles. The agents do however retain free will as to whether they *cooperate in any one task*. If for example an agent is overloaded with tasks it is not obliged to take on new tasks.

Clearly the network environment and the control structure are directly linked. The relationship is that of a feedback loop as shown in Figure 4.8. The rules for how the environment may influence the control structure *embody the designers knowledge about what makes an effective correlation between environment state and the ability to satisfy tasks efficiently*. The outside influences shown in Figure 4.8 such as failures or node additions can also influence the control architecture by showing up as changes in the environment.



**Figure 4.8:** The environment and control structure are related by a feedback loop where each has influence on the other. The influence of the environment is determined by rules which represent knowledge about which types of organizations are good for which environmental state.

This is different in emphasis to (as far as the authors are aware) previous work on adaptive organisations. In previous work the *tasks to be completed* have driven organisational adaptation, in (Decker and Lesser, 1995) for example local re-organisation is done when a new set of task-groups arrives. Ishida, Gasser and Yokoo, (1992) use demand arrival rate to determine adaptation, this is analogous to linking adaptation to one environmental factor. The reason for shifting from using tasks and current jobs to determine the organisational shape is to move away from *coordinating for one big task* towards dealing with many small tasks arriving over an extended period of time. The network state is extremely important in the routing problem and each new demand has a relatively small (often subtle) effect on it. Using the network state to help determine the

<sup>20</sup>Note they can reason about what actions to take in case another agent cannot be contacted.

<sup>21</sup>See also (So and Durfee, 1992).

organisation state is liable to be simpler and more effective than relying on metrics based upon demand arrival. A second reason is that the state of the network can affect the infrastructure the agent society is based upon so changes in the network state can force changes in the control organisation.

The influence of the environment is in fact expressed as knowledge the agent has about how to adapt its sphere of control. This is important since external control of the adaptation would impinge upon an agent's autonomy. In turn reliance on an external adaptation mechanism could cause the whole organisation to fail if parts of the network fail.

An organisation that adapts well is still only half the story. Agents still need to cooperate within the organisation to perform individual tasks. In fact the organisation provides a macro structure for coordination, the coarse grain responsibilities entailed by the relations parent, child etc. define a high level order to the agent society. The organisation also completely defines the information distribution in the society, determining exactly who has access to what information. Jennings (1995) and Tambe (1997) both point out that in complex dynamic domains it is almost impossible to pre-plan all possible combinations of interactions off-line. This is supported by experiences with the industrial application of Multi-agent systems (Jennings, Corera, Laresgoiti, 1995). We therefore apply finer grained coordination techniques within the organisational structure to coordinate agents for solving individual tasks.

The issue of coordinating agents for a single routing task is complex and the outline above is a simple example. Jennings and Mamdani (1992) show a considerable time saving in resource use when applying a behavioural specification of cooperation (joint commitments). We apply the formal specification of cooperation from (Wooldridge and Jennings, 1994) and more specifically the methodology behind GRATE\* (Jennings, 1995) to help handle coordination at the task level (these two works were closest to our needs). The analysis of when commitments should end and the exchange of information about freeing up resources is critical in a domain where many small tasks interact and compete for resources. Using a formal specification of cooperation helps to highlight where these issues arise.

One of the problems in applying coordination approaches such as those based upon joint commitments is the time needed to start the process of coordination for each task. As pointed out in Section 3.1 when the task itself is relatively small and there are strict time criteria, long start-up times are not acceptable. Jennings and Mamdani (1992) suggests that this can be improved by the use of common patterns which accumulate over time. In our case some parts of the model of responsibilities are part of the organisation (in the definition of parent-child relationships). This means that some of the work in team formation and plan formation (steps from (Wooldridge and Jennings, 1994)) has already been done before a demand arrives. The organisation provides a structure for deciding quickly who to involve in problem solving and how to control the process.

The cooperation work was applied principally as a design methodology to define the responsibilities of agents to each other during problem solving. This means agents are less flexible (they only have knowledge for coordination in a particular domain and restricted to a specific set of problem solving protocols) but much of the coordination knowledge can be

specified as part of the organisational knowledge. The benefit is an increase in the clarity and speed of the decision making process which is essential in the routing domain.

## 5. Conclusions

Modern high-speed communication networks are rapidly becoming more dynamic and complex. There is now significant demand for routing to be dynamic and adapt to changing demands. Many researchers have argued that Multi-agent systems and Distributed Artificial Intelligence can solve this type of problem, and substantiated these claims with prototype implementations. We have argued that since the information required for routing is distributed in the network, it also brings with it problems of *distributing* and *updating* this information to the different agents and coordinating agents to solve routing problems. These problems are important to address when agent-based solutions need to scale to networks of realistic size.

Section 3 of this chapter identifies some of the key issues involved in the routing problem and suggests areas of DAI which require further development before these challenges can be addressed. Some of the main areas which need further research are:

- Detailed modelling and analysis of communication cost (taking into account spatially distributed systems).
- Team formation - how to decide which agents to involve in servicing a particular task.
- Dealing with the potential interactions of many tasks on both physical (domain) resources and the computational resources of the agents.

The second part of the chapter proposed a scheme based on blocking island equivalence classes for structuring information distribution in an agent system. It was then discussed how this partitioning scheme could be used to address some of the problems identified in a scaleable way by restructuring the organisation as the network state changes. We believe that such structures will be essential for implementing Multi-agent systems at the scale and reactivity required for real networks and hope to stimulate further work that addresses the many open issues which remain.

## 6. Acknowledgements

The authors would like to extend their thanks to the other partners in the on-going SPP-ICC IMMune project (of which this work is a part). Funding for IMMune from the Swiss National Science Foundation<sup>22</sup> is gratefully acknowledged. Special thanks also go to Omar Belahdar and Ulises Cortes for their useful comments on earlier drafts.

---

<sup>22</sup>Project number SPP-ICC 5003-45311.

## 7. References

(Alles, 1995) A. Alles. "ATM Internetworking". Technical Report, Cisco Systems, Inc. May 1995. (Also presented at Engineering InterOp, Las Vegas, March 1995.)

(Appleby and Steward, 1994) S. Appleby and S. Steward. "Mobile Software Agents for Control in Telecommunications Networks". *British Telecom Technology Journal* **12**(2), 1994.

(Ash et al, 1991) G. R. Ash, J. S. Chen, A. E. Frey and B. D. Huang. "Real-Time Network Routing in an Integrated Network". In *Proceedings of ITC-13*, Copenhagen, Denmark, 1991.

(Ash and Huang, 1993) G. R. Ash and B. D. Huang. "An Analytical Model for Adaptive Routing Networks". *IEEE Transactions on Communications*, **41**(11), pages 1748--1759, November 1993.

(ATM Forum, 1996) ATM-Forum, "P-NNIv1.0 - ATM Forum approved specification, af-pnni-0055.000". 1996.

(Bertsekas and Gallager, 1992) D. P. Bertsekas, and R. Gallager. "Data Networks, 2nd Edition". Prentice Hall, 1992.

(Bertsekas and Tsitsiklis, 1989) D. P. Bertsekas, and J. N. Tsitsiklis. *Parallel and Distributed Computation*, Prentice-Hall, Englewood Cliffs, NJ. 1989.

(Cammarata, McArthur, Steeb, 1983) S. Cammarata, D. McArthur and R. Steeb, R. "Strategies of Cooperation in Distributed Problem Solving". In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'83)*. Pages 767--770. 1983.

(Castelfranchi, 1995) C. Castelfranchi. "Commitments: From Individual Intentions to Groups and Organisations". In *Proceedings of the first International Conference on Multi-Agent Systems (ICMAS'95)*. Edited by V. R. Lesser. Pages 41--48. 1995.

(Cherkassky, Goldberg, Radzik, 1994) B. V. Cherkassky, A. V. Goldberg, and T. Radzik. "Shortest Path Algorithms: Theory and Experimental Evaluation". In *Proceedings of the 5th ACM-SIAM Annual Symposium on Discrete Algorithms*. Pages 516--525. 1994.

(Cidon et al, 1995) I. Cidon, I. Gopal, M. A. Kaplan and S. Kuten. "A Distributed Architecture of High Speed Networks". *IEEE Transactions on Communications* **43**(5). 1995.

(Clark, Grossner, Radhakrishan, 1996) R. Clark, C. Grossner and T. Radhakrishan. "Consensus and Compromise: Planning in Cooperating Agents". *Journal of Cooperative Information Systems* 5(1). Pages 27--72. 1996.

(Clearwater, 1996) S. H. Clearwater. *Market Based Control: A paradigm for Distributed Resource Allocation*. World Scientific, Singapore. 1996.

(Cohen and Levesque, 1990) P. R. Cohen, and H. J. Levesque. "Intention is Choice with Commitment", *Artificial Intelligence* volume 42. Pages 213--261. 1990.

(Conry, Meyer, Searlman, 1988) S. E. Conry, R. A. Meyer, and J. E. Searleman. "Multistage Negotiation in Distributed Planning". In *Readings in Distributed Artificial Intelligence*, edited by A. H. Bond and L. Gasser. Pages = 367--384. Morgan Kaufmann 1988.

(Conry et al, 1991) S. E. Conry, K. Kuwabara, V. R. Lesser, and R. A. Meyer. "Multistage Negotiation for Distributed Constraint Satisfaction". *IEEE Transactions on Systems, Man and Cybernetics*, SMC-21(6). Pages 1462--1477. November/December 1991.

(Corkill, 1993) D. D. Corkill. "A Framework for Organisational Self-Design in Distributed Problem Solving Networks". PhD Thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, February 1983.

(Corkill and Lesser, 1983) D. D. Corkill and V. R. Lesser. "The Use of Meta-Level Control for Coordination in a Distributed Problem Solving Network". In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'83)*. Pages 748--756. 1983.

(Decker and Lesser, 1992) K. S. Decker and V. R. Lesser. "Generalising the Partial Global Planning Algorithm". *Journal of Intelligent and Cooperative Information Systems* 1(2). Pages 319--346. 1992.

(Decker and Lesser, 1993) K. S. Decker and V. R. Lesser. "A One-shot Coordination Algorithm for Distributed Sensor Networks". In *Proceedings AAAI'93*. Pages 210--216, 1993.

(Decker and Lesser, 1995) K. S. Decker and V. R. Lesser. "Designing a Family of Coordination Algorithms". In *Proceedings of the first International Conference on Multi Agent Systems (ICMAS'95)*. Pages 73--80. 1995.

(Decker, 1996) K. S. Decker. "TAEMS: A Framework for Environment Centred Analysis and Design of Coordination Mechanisms". In *Foundations of Distributed Artificial Intelligence* edited by F. M. P. O'Hare and N. R. Jennings. Pages 429--448. John Wiley & Sons 1996.

(Dijkstra, 1959) E. W. Dijkstra. "A Note on Two Problems in Connection with Graphs". *Numerical Mathematics* 1. Pages = "269--271". 1959.

(Durfee and Montgomery, 1990) E. H. Durfee and T. A. Montgomery. "A Hierarchical Protocol for Coordinating Multi-agent Behaviours", In *Proceedings AAAI'90*. Pages 83--93. 1990.

(Durfee and Lesser, 1991) E. H. Durfee and V. R. Lesser. "Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation". *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-21**(5). Pages 1167--1183. 1991.

(Durfee and So, 1997) E. H. Durfee and Y. So "The Effects of Runtime Coordination Strategies within Static Organisations". In *Proceedings IJCAI'97*. Pages 612--618. 1997.

(Ford and Fulkerson, 1962) L. R. Jr. Ford and D. R Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ. 1962

(Fox, 1981) M. S. Fox. "An Organisational View of Distributed Systems". *IEEE Transactions on Systems, Man and Cybernetics* **SMC-11**(1). Pages 70--80, 1981.

(Frei and Faltings, 1998) C. Frei and B. Faltings. "Dynamic Hierarchy of Intelligent Agents for Network Management". In *Proceedings of the Second International Workshop on Intelligent Agents for Telecommunications Applications (IATA'98)*. Edited by S. Albayrak and F. J. Garijo. Pages 1--16. Springer (as LNAI-1437). 1998.

(Garey and Johnson, 1979) M. R. Garey and D. S. Johnson. "Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman and Co, 1979.

(Gasser, 1992) L. Gasser. "DAI Approaches to Coordination". In "Distributed Artificial Intelligence: Theory and Praxis". Edited N. M. Avouris and L. Gasser. Pages 31--51. Kluwer 1992.

(Gibney and Jennings, 1998) M. A. Gibney and N. R. Jennings. "Dynamic Resource Allocation by Market-Base Routing in Telecommunications Networks". In *Proceedings of the Second International Workshop on Intelligent Agents for Telecommunications Applications (IATA'98)*. Edited by S. Albayrak and F. J. Garijo. Pages 102--117. Springer (as LNAI-1437). 1998.

(Grosz, 1996) B. Grosz. "Collaborating Systems". *AI Magazine* **17**(2). Pages 67--85. 1996.

(Guichard and Ayel, 1994) F. Guichard and J. Ayel. "Logical Reorganisation of DAI Systems". In *Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures*

and Languages (ATAL'94). Pages 118--128. Springer Verlag (as Lecture Notes in Artificial Intelligence 890). 1994.

(Gymtrasiewicz, Durfee, Wehe, 1991) P. J. Gymtrasiewicz, E. H. Durfee and D. K. Wehe. "The Utility of Communication in Coordinating Intelligent Agents". In *Proceedings AAAI'91*. Pages 166--172. 1991.

(Hayzelden and Bingham, 1998) A. L. G. Hayzelden and J. Bingham. "Heterogeneous Multi-Agent Architecture for ATM Virtual Path Network Resource Configuration ". In *Proceedings of the Second International Workshop on Intelligent Agents for Telecommunications Applications (IATA'98)*. Edited by S. Albayrak and F. J. Garijo. Pages 45--59. Springer (as LNAI-1437). 1998.

(Ishida, Gasser, Yokoo, 1992) T. Ishida, L. Gasser and M. Yokoo. "Organisation Self-Design of Distributed Production Systems". *IEEE Transactions on Knowledge and Data Engineering*, 4(2). Pages 123--134. April 1992.

(Jennings and Mamdani, 1992) N. R. Jennings and A. Mamdani. "Using Joint Responsibility to Coordinate Collaborative Problem Solving in Dynamic Environments". In *Proceedings AAAI'92*. Pages 169--275, 1992.

(Jennings, 1995) N. R. Jennings. "Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems Using Joint Intentions". *Artificial Intelligence* 75. Pages 195--240, 1995.

(Jennings, Corera, Laresgoiti, 1995) N. R. Jennings, J. M. Corera and I. Laresgoiti. "Developing Industrial Multi-Agent Systems". In *Proceedings of the first International Conference on Multi-Agent Systems (ICMAS'95)*. Pages 423--430. 1995.

(Jennings, 1996) N. R. Jennings. "Coordination Techniques for Distributed Artificial Intelligence". In *Foundations of Distributed Artificial Intelligence* edited by F. M. P. O'Hare and N. R. Jennings. Pages 187--210. John Wiley & Sons 1996.

(Jennings, Sycara, Wooldridge, 1998) N. R. Jennings, K. Sycara and M. Wooldridge. "A Roadmap of Agent Research and Development". *Journal of Autonomous Agents and Multi-Agent Systems* 1. Pages 275--306. March 1998.

(Kazovsky, Benedetto, Willner, 1996) L. Kazovsky, S. Benedetto and A. Willner. "Optical Fiber Communication Systems". Artech House, Boston, 1996.

(Kosoresow, 1993) A. P. Kosoresow. "A Fast First-Cut Protocol for Agent Coordination". In *Proceedings AAAI'93*. Pages 237--242. 1993.

(Kraus and Wilkenfeld, 1991) S. Kraus and J. Wilkenfeld. "The Function of Time in Cooperative Negotiations". In *Proceedings AAAI'91*. Pages 179--184. 1991.

(Kraus, Wilkenfeld, Zlotkin, 1995) S. Kraus, J. Wilkenfeld and G Zlotkin. "Multi-agent Negotiation under Time Constraints". *Artificial Intelligence* **75**. Pages 297--345. 1995.

(Kuwabara, 1996) K. Kuwabara, T. Ishida, Y. Nishibe and T. Suda. "An Equalibratory Market Based Approach for Distributed Resource Allocation and its Applications to Communication Network Control". In (Clearwater, 1996). Pages 53--73. 1996.

(Laasri et al, 1992) B. Laasri, and H. Laasri, S. Lander and S. Lesser. "A Generic Model for Intelligent Negotiating Agents.". *International Journal of Intelligent and Cooperative Information Systems* **1**(2). Pages 291--317. 1992.

(Lee, Hluchyj, Humblet, 1995) W. L. Lee, M. G. Hluchyj and P. A. Humblet. "Routing subject to Quality of Service Constraints". *IEEE Network*. July/August 1995.

(Lesser, Corkill, Durfee, 1987) V. R. Lesser, D. D. Corkill and E. H. Durfee. "An Update on the Distributed Vehicle Monitoring Testbed". University of Massachusetts, Amherst, Computer Science Technical Report 87-111. 1987.

(Lin and Yee, 1993) F. Y. S. Lin and J. R. Yee. "A Real-Time Distributed Routing and Admission Control Algorithm for ATM Networks. In *Proceedings INFOCOM* **12**(3). Pages 792--801. 1993.

(Malone, 1988) T. W. Malone. "Modeling Coordination in Organisations and Markets". In *Readings in Distributed Artificial Intelligence*, edited by A. H. Bond and L. Gasser. Pages 151--158. Morgan Kaufmann 1988.

(Perlman, 1992) R. Perlman. *Interconnections: Bridges and Routers*. Addison Wesley 1992.

(Plotkin, 1995) S. Plotkin. "Competitive Routing of Virtual Circuits in ATM Networks". *IEEE Journal on Selected Areas in Communications* **13**(6). Pages 1128--1135. August 1995.

(Prasad, 1996) M. V. N. Prasad, K. S. Decker, A. Garvey and V. R. Lesser. "Exploring Organisational Designs with TAEMS: A Case Study of Distributed Data Processing". In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS'96)*. Pages 283--290. 1996.

(de Pryker, 1995) M. de Pryker. *Asynchronous Transfer Mode: Solution for Broadband ISDN, 3rd Edition*. Prentice Hall, 1995.

(Sexton and Reid, 1997) M. Sexton and A. Reid. *Broadband Networking: ATM, SDH, and SONET*. Artech House, Boston/London, 1997.

(Shoham and Tennenholtz, 1992) Y. Shoham and M. Tennenholtz. "On the Synthesis of Useful Social Laws for Artificial Agent Societies". In *Proceedings AAAI-92*. Pages 276--281. 1992.

(Siegl and Trausmuth, 1996) M. R. Siegl and G. Trausmuth. "Hierarchical Network Management: a Concept and its Prototype in SNMPv2". *Computer Networks and ISDN Systems* **28**(4). Pages 441--452. February 1996.

(So and Durfee, 1992) Y. P. So and E. H. Durfee. "A Distributed Problems Solving System for Computer Network Management". In *International Journal of Intelligent and Co-operating Information Systems*, **1**(2), 1992.

(Somers, 1996) F. Somers. "HYBRID: Intelligent Agents for Distributed ATM Network Management". In *Working Notes of the Workshop on Intelligent Agents for Telecom Applications - ECAI'96*. 1996.

(Tambe, 1997) M. Tambe. "Towards Flexible Teamwork". *Journal of Artificial Intelligence Research* **7**. Pages 83--124. 1997.

(Vedantham and Iyengar, 1998) S. Vedantham and S. S. Iyengar. "The Bandwidth Allocation Problem in the ATM network model is NP-Complete". *Information Processing Letters* **65**. Pages 179-182. 1998.

(Varga, Jennings, Cockburn, 1992) L. Z. Varga, N. R. Jennings and D. Cockburn. "Integrating Intelligent Systems into a Cooperating Community for Electricity Distribution Management". In *Proceedings Expert Systems '92*. 1992.

(Wang and Crowcroft, 1996) Z. Wang and J. Crowcroft. "Quality-of-Service Routing for Supporting Multimedia Applications". *IEEE Journal on Selected Areas in Communications* **14**(7). 1996.

(Werner, 1989) E. Werner. "Cooperating Agents: A Unified Theory of Communication and Social Structure". In *Distributed Artificial Intelligence, Volume II*, edited by L. Gasser and M. Huhns. Pages 3--36. Pitman 1989.

(Wellman, 1992) M. P. Wellman. "A General-Equilibrium Approach to Distributed Transportation Planning". In *Proceedings AAAI'92*. Pages 282--289, 1992.

(Wellman, 1994) M. P. Wellman. "A Market-Oriented Programming Environment and its Application to Distributed Multicommodity Flow Problems". *Journal of Artificial Intelligence Research* **1**(1). Pages 1--23. 1994.

(Wooldridge and Jennings, 1994) M. Wooldridge and N. R. Jennings. "Towards a Theory of Cooperative Problem Solving". In *Proceedings Workshop Modelling Autonomous Agents in a Multi Agent World (MAAMAW'94)*. Pages 15--26. 1994.

(Yamaki, Wellman, Ishida, 1996) H. Yamaki, M. P. Wellman and T. Ishida. "A Market-Based Approach to Allocating QoS for Multimedia Applications". *In Proceedings: Second International Conference on Multi Agent Systems (ICMAS'96)*. Pages 385--392. 1996.