2-Day Introduction to Agent-Based Modelling

Day 2: Session 8

Exploring Model Collections, NetLogo Lists, Further Resources

Manchester Metropolitan University

Data structures built into NetLogo

Variables can have values in terms of:

- Numbers
- Strings (text)
- Colours
- Booleans (True/False)

But also sometimes more complicated things:

- Lists of the above, [1 2 3] or [red blue red]
- Lists of lists of the above, [[1 2] [“A” “B”] True]

You sometimes need to store and manipulate these more complex structures

Other structures such as matrices, arrays and tables are available as extensions (see extensions section of the NetLogo manual)

List Primitives

Making lists:

- Writing them explicitly: [grey violet]
- Making them as a result of a calculation:
  list (mean [food] of buyers) (mean [food] of buyers)

Extracting from lists:

- First item from a list: first [grey violet]
- All *but* the first item: but-first [“a” “v” “w”]
- An item from a certain position in the list:
  item 3 [4 3 5 6 7]
  (note: NetLogo lists start with position 0!!)

A simulation of a Peer-2-Peer file sharing network

- Computers are joined in a distributed fashion without central servers
- Files in their store can be downloaded by others in the network
- Nodes can send out queries for files they want – these go out to all its network neighbours
- If a node gets a query for one of its files, it sends it back to the originator of the query
- Otherwise nodes pass on queries it gets to its neighbours (unless the query has exceeded the maximum number of hops allowed) thus queries are distributed like gossip

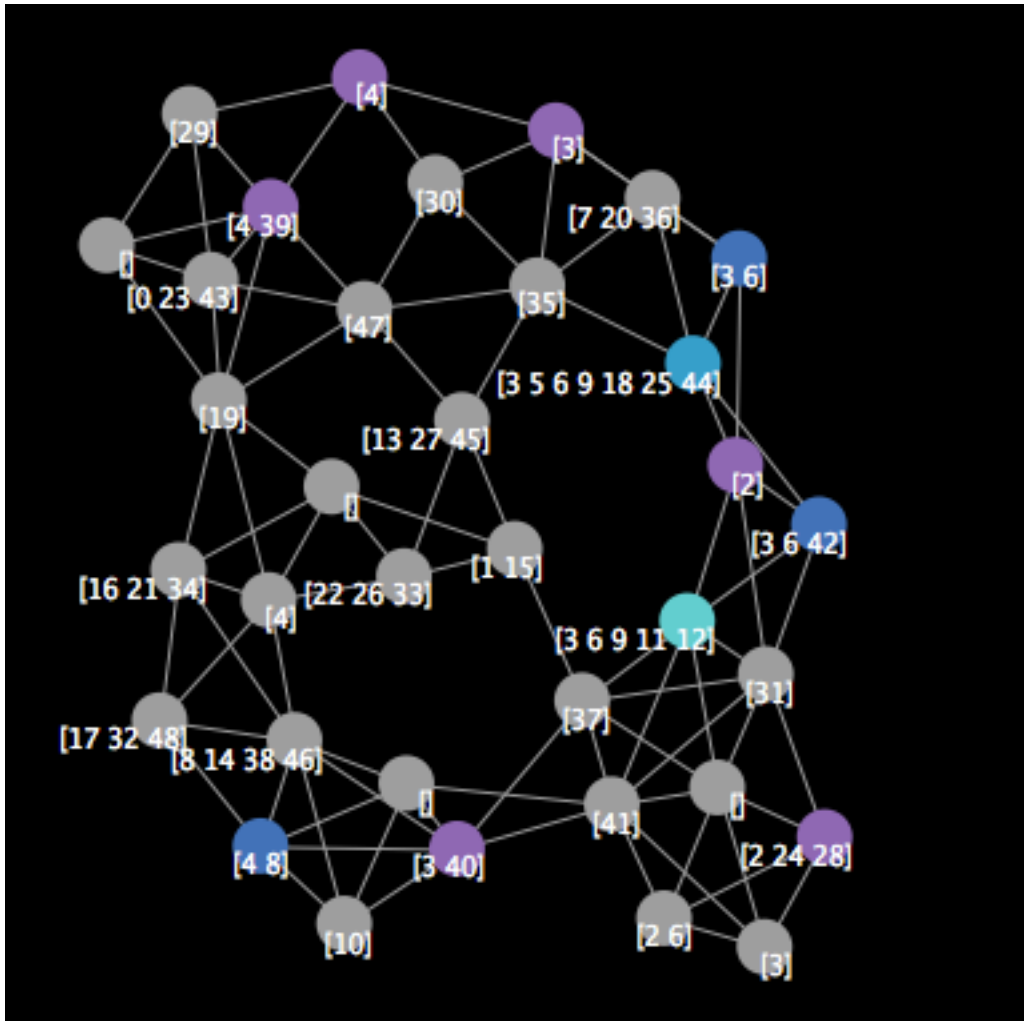
File Collecting Behaviour

- Files are represented by integers: 0, 1, 2, …, up to the value of “max-file - 1”
- These are each distributed to random to start with
- Nodes are each given a base-target (a small integer above 1), it then seeks for numbers that are multiples of this, sending out queries in turn for each these
- As files become copied around it becomes increasingly easy to find them

Use of lists in this simulation

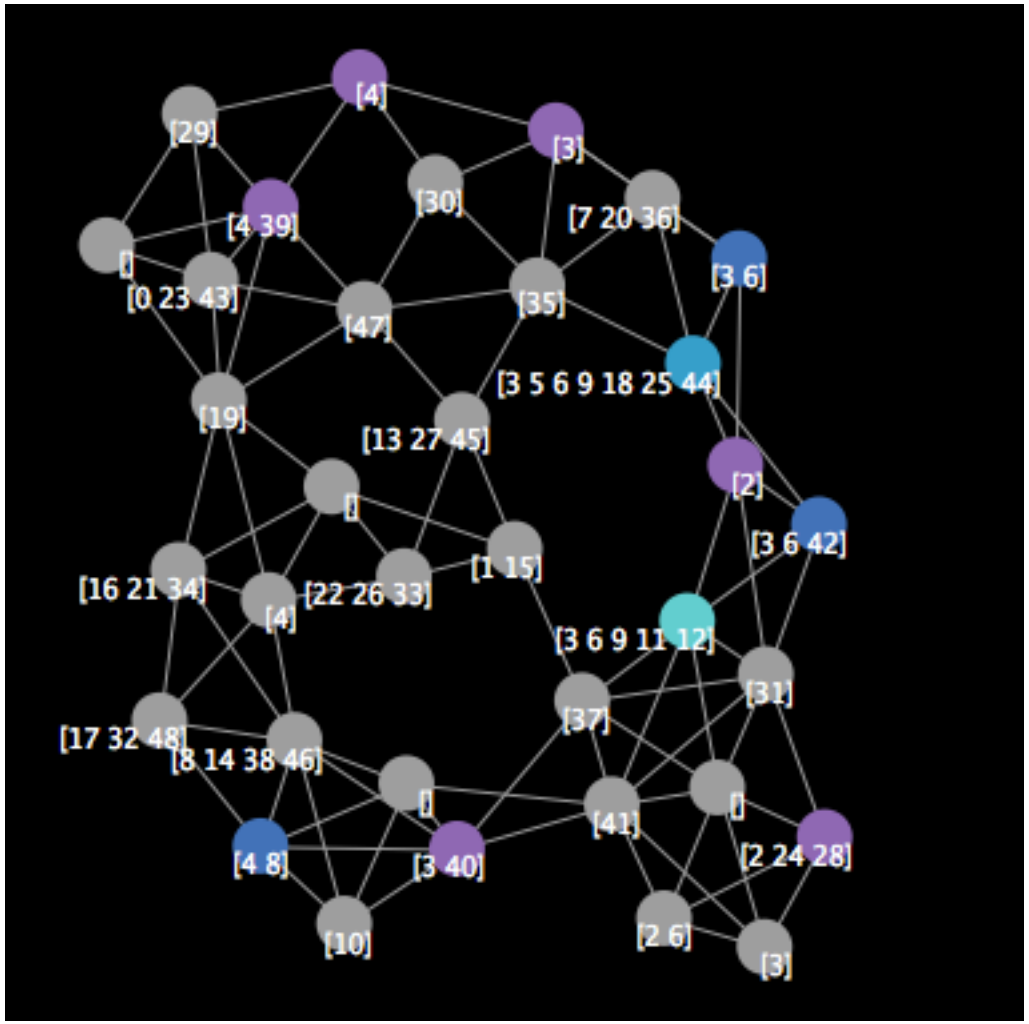
- To collect the files, each node has a list called "file-store", as it gets files it adds it to this (at its front using first put – fput)
  - set file-store fput new-file file-store
- To check if it has a file is uses this
  - member? a-file file-store
- Each node also has a list of queries it has received called "messages"
- When it receives queries from others it adds them to the messages list (using fput as above)
- Then (once a time click) goes through the list of messages (using foreach) and either sends the file (if it has it) or passes on the query (unless it has reached its maximum allowed hops)

The Simulation Display



The Simulation Display

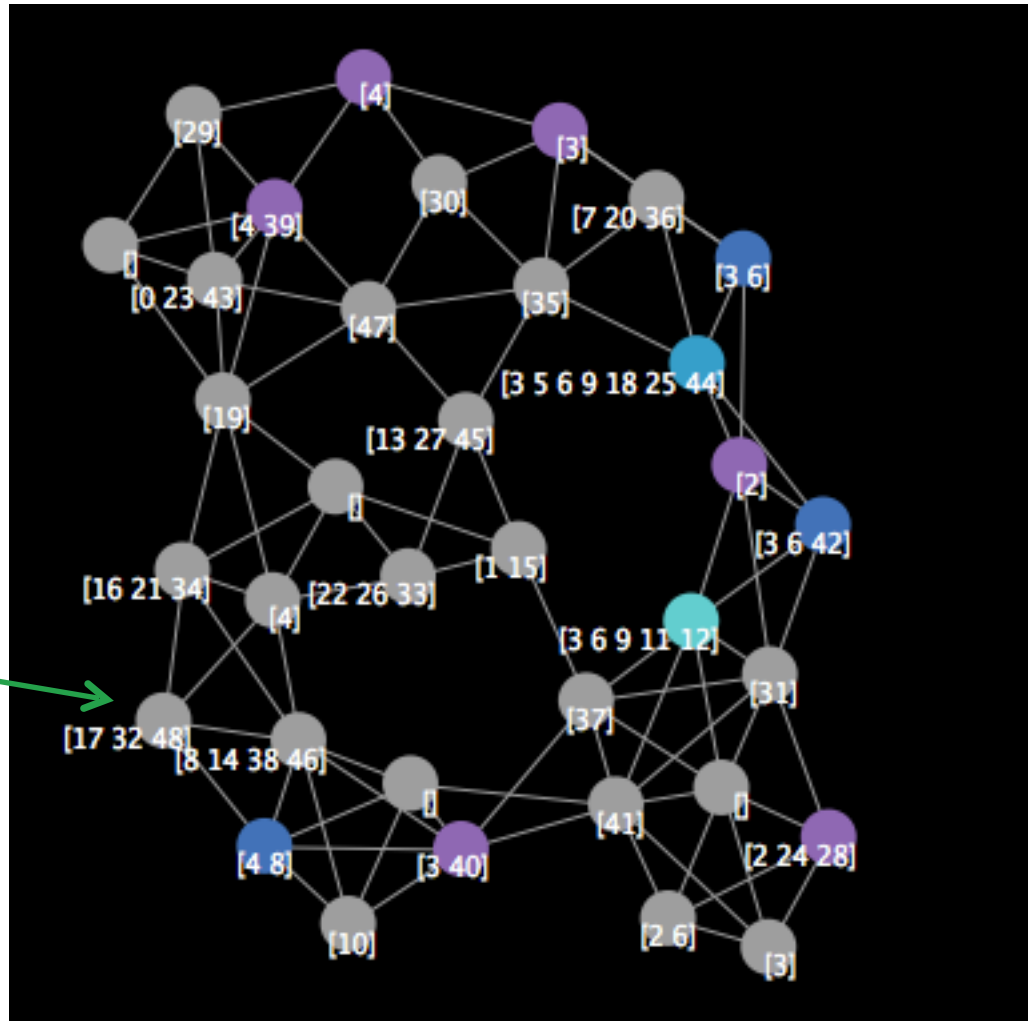
Load the "8-p2p.nlogo" simulation



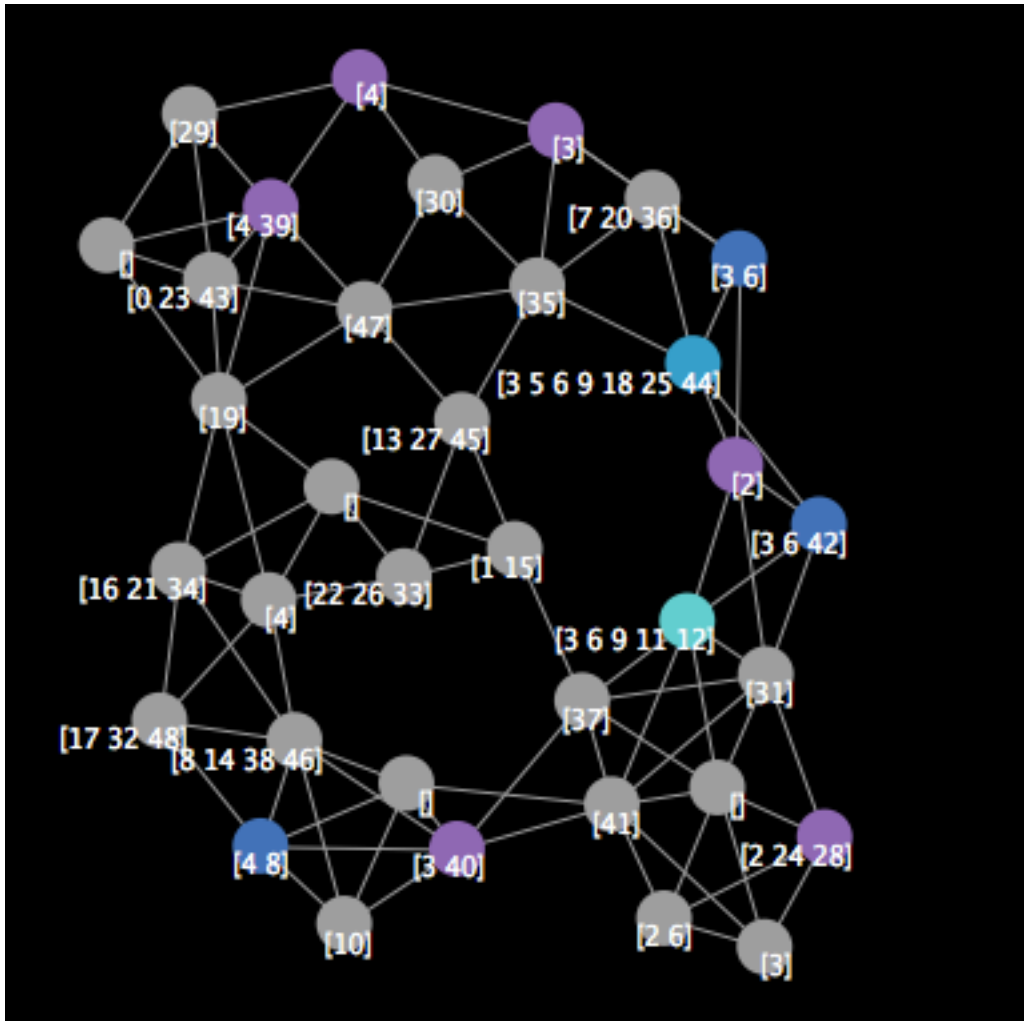
The Simulation Display

A node in the P2P network, grey when it has found none of its target “files”, then blue…green… yellow… red etc. as it finds more.

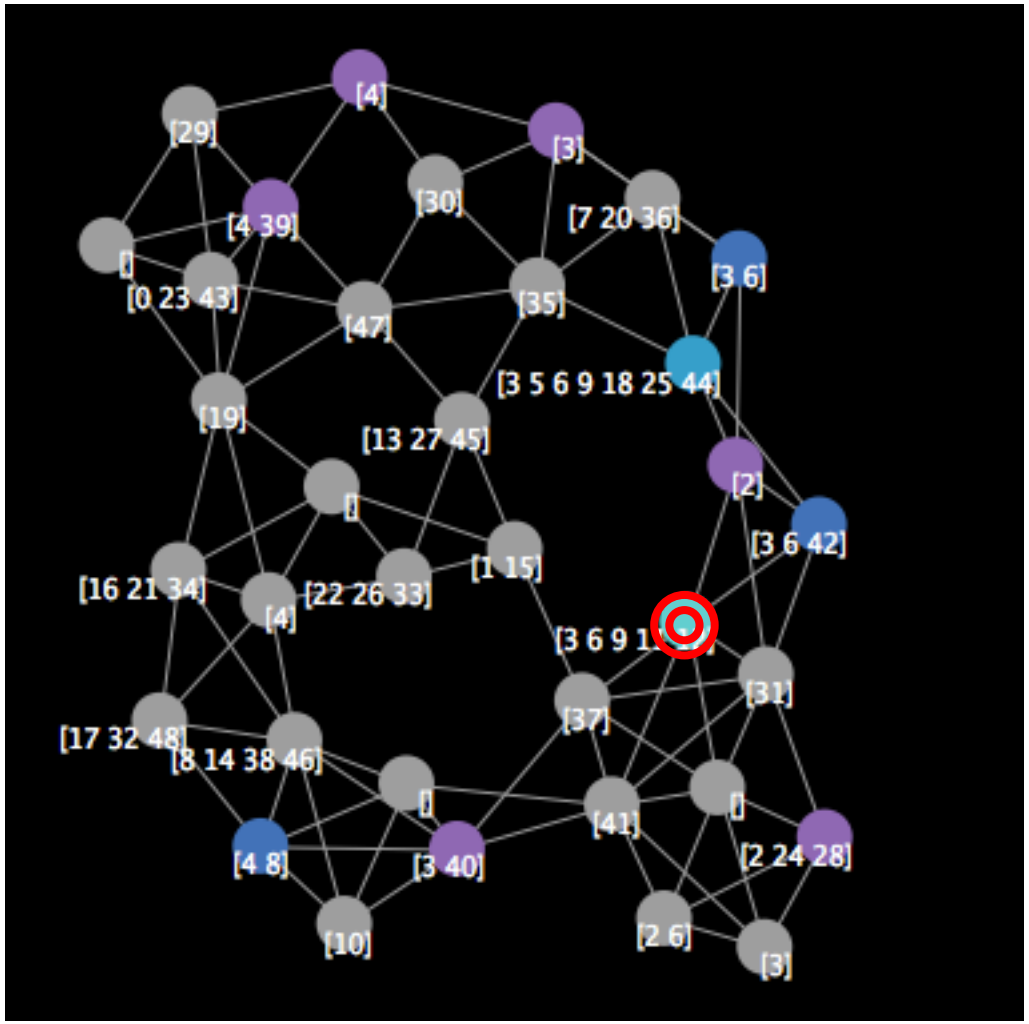
The label of each node shows its list of “files”.



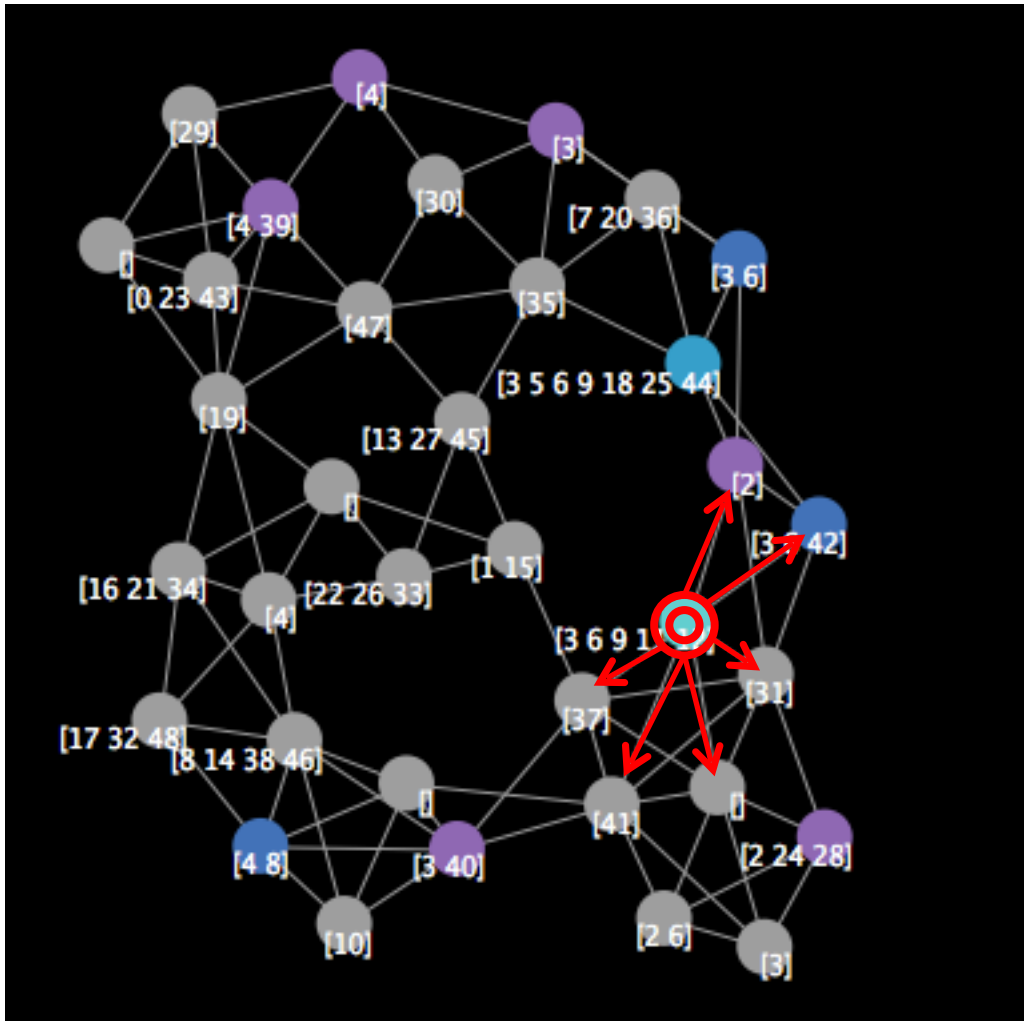
The Simulation Display



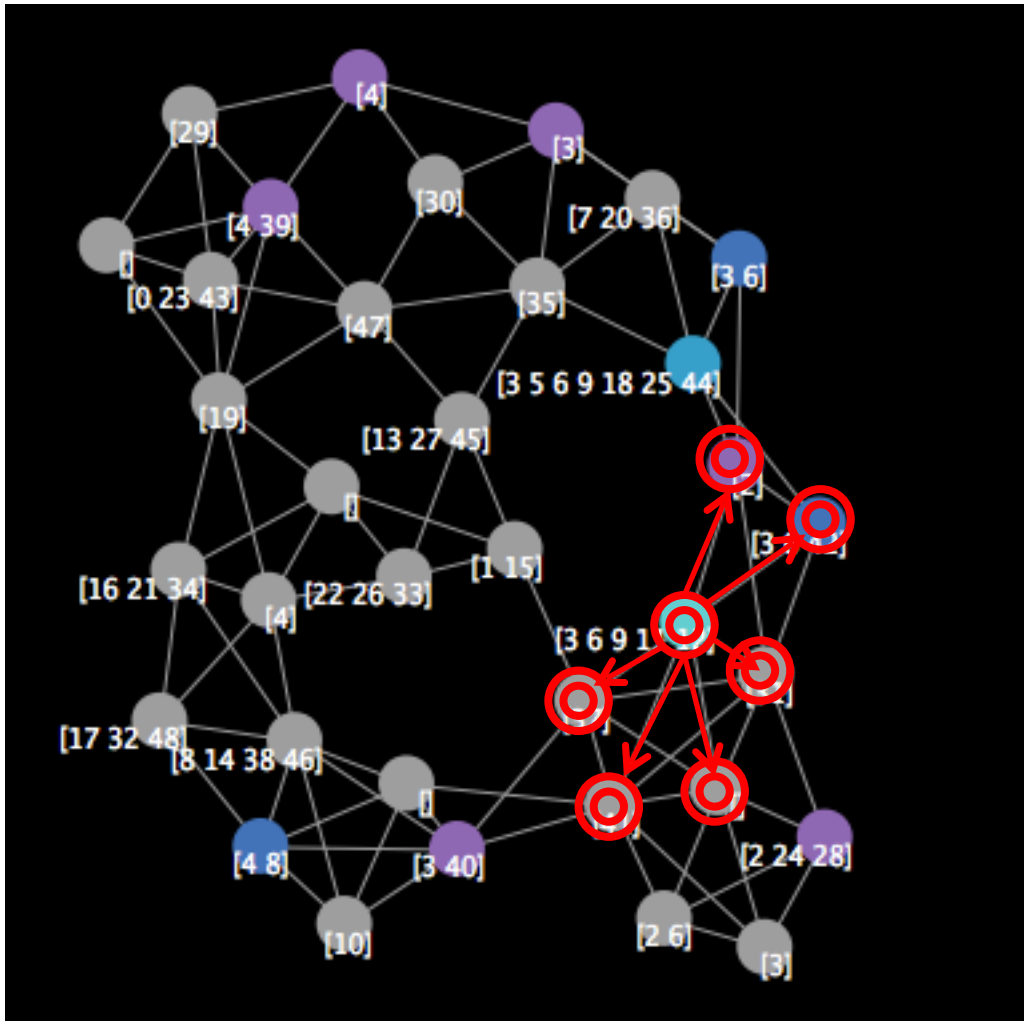
The Simulation Display



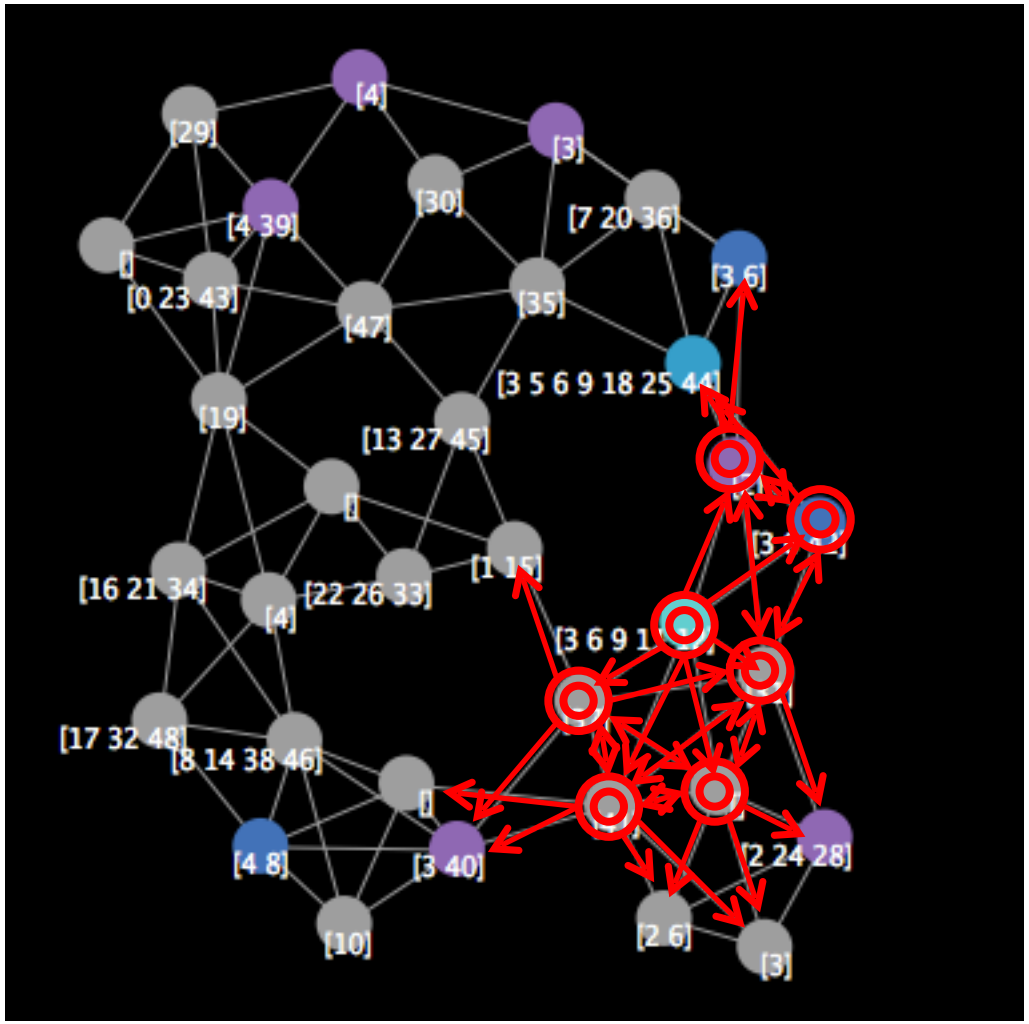
The Simulation Display



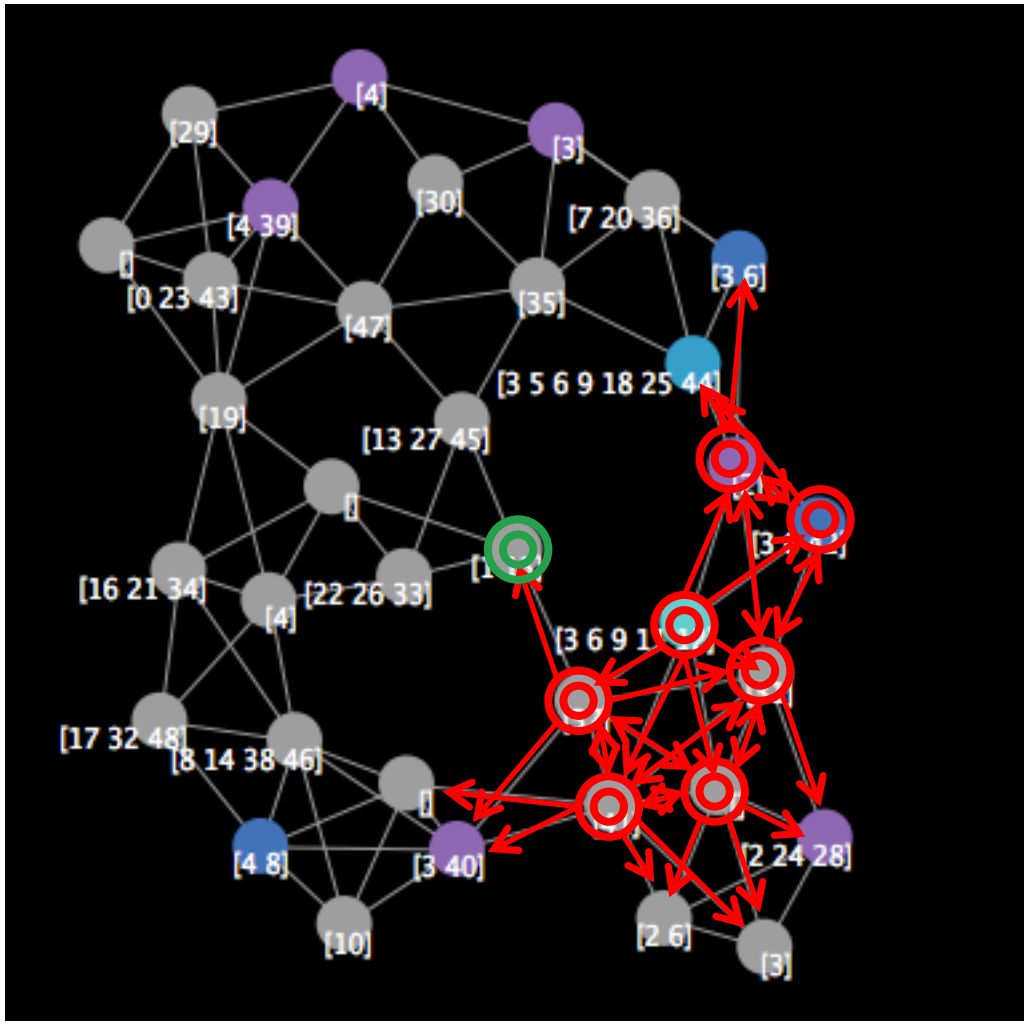
The Simulation Display



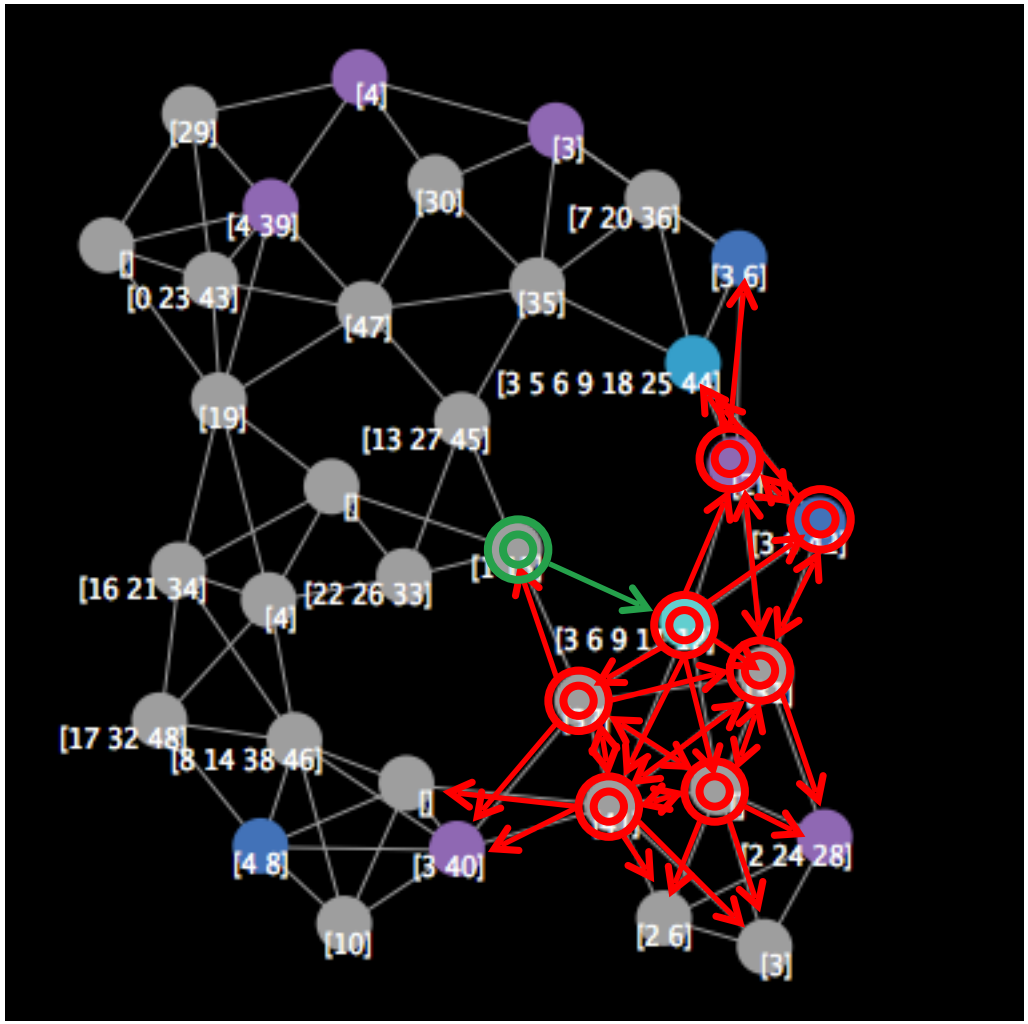
The Simulation Display



The Simulation Display



The Simulation Display



Model parameters

- population – number of nodes
- number-links-each – number of links of each node in the network (both initial number and a target number as the simulation progresses)
- num-hops – maximum number of times a query can be passed on before it "dies"
- max-file – the integer the number representing files go up to
- prob-newquery – the probability each time that each node initiates a new query for the next file in its set that it wants

Things to try

- Can you work out the conditions under which files are effectively shared to all that need it?
- Could you add that some nodes are cooperators (share their files) and some not (pass on queries but never send files)?
- What happens if you add some of the rules from the last example about changing network structure by nodes?

Recap of NetLogo Features Covered

- The Interface: view of world, sliders, buttons, plots, histograms, monitors, info tab, code
- An idea of the general structure of a typical simulation
- Use of agents, different breeds, their local variables, “ask”, using agent sets to select subsets of all agents
- Interacting with other agents in different ways
- Different programming contexts: observer, agent, patch, link, etc. and the effect in terms of kinds of procedure and variables
- Different ways of ‘peeking’ into a simulation to find what is happening
- Construction of networks, changing them and using them
- A little bit about NetLogo lists and their use
- A bit about extracting data from simulations

Resources for further learning
(listed on the course website)

- The *NetLogo* website (tutorials etc.)
- Looking at other models, trying them out, looking at their code for ideas of how to program/organise a simulation (see collections of NetLogo models on the course website)
- The *NetLogo user group* – you can ask for help there when stuck (in moderation)
- The books mentioned on the course website, especially those with NetLogo examples
- The *Journal of Artificial Societies and Social Simulation*, for examples and discussions of how to simulate etc.
- The *European Social Simulation Association*: its summer school, its bi-annual conference, essa@work, its newsletter
- Openabm.org – its model library and announcements
- The SimSoc mailing list for relevant announcements etc.
- BUT, if you can get it, find someone who knows NetLogo programming who can help you when you get stuck (as we all do from time to time – even experts)

The Very End…
Good Luck!

2-Day Introduction to Agent-Based Modelling
http://cfpm.org/simulationcourse

Manchester Metropolitan University