Complexity science and 21st Century Issues

London School of Economics 25-26 March 2004

Professor Scot -Kirkpatrick, professor in the School of Engineering and computer science, Hebrew University and Exystence Network of Excellence.

"From the scale-free Web to Avogadro-scale Engineering"

I very much liked Lord May's frame work of the Tycho Brahe / Kepler / Newton stages of understanding of phenomena, but I think I disagree with his feeling that we don't have Newtons anymore and that little is going on at the foundations of science. Whilst there may not be any Newtons we still do our best to find foundational ideas about how complex models can be analysed and how predictions can be founded more soundly for very large and very random systems.

If we take a Brahe stage in understanding the Internet then the statistics do not make much sense and are just collections of data. The analytical or Kepler stage is finding patterns without asking where the patterns came from and for the Internet this was pretty much about casting the degree of distribution of things in a $\log_n/\log_n$ plot and discovering parallel distributions in the connectivity of things at various levels. A very important step in understanding how this comes about was the observation that long scale phenomena, (i.e. those that repeat themselves over many scales) are a natural consequence of things that grow with a preference for joining an existing network but retain characteristics of their own. When you put together a lot of things that are not very affected by their neighbours you do not get particular patterns of distribution. When you aggregate things in a particular way and the pattern at any point has a strong influence on what comes next you get phenomena that are transported down to smaller and smaller scales. That observation is admittedly not like the Newtonian theory of gravity, but it is behind a lot of the work going on today to understand very large networks that are continually growing.

So I'm going to talk about physical networks and information networks. The project 'Evergrow' is focusing principally on the physical Internet and to a lesser extent on the web of information on the Internet. The physical Internet today has at least a billion hosts and a comparable number of robots (the elementary points at which paths branch). At the next level of hierarchy are the autonomous systems (ASs) and there are at least 10 thousand of them today. Like the hosts they appear and disappear so the number grows more slowly than the informational content, but everything is changing pretty fast. The timescale or presence of a specific host may be minutes and the timescale of registration for long term existence of hosts is days.

When you look at the information levels on the Web, there are several orders of magnitude of things and if you count down to character or byte there are several orders of magnitude more. But if you take the view that 10^4 kilobytes is approximately a web page we're talking about a billion things linked in a network that has scale free characteristics and if you put on your Kepler hat you can find power law characterisations of the content.

Genetic expression data from the automation of molecular biology experiments also operates with about 10^9 things. The number of items in the human genome is about on that scale and each of these systems present enormous problems of interpretation. In the case of genetic expression data we'd like to have tools that go beyond the Kepler stage of finding patterns or groupings to the Newton stage of understanding the code so there is a very active process going on at all stages of the analysis. I'm not going to talk in detail about the tools used in genetic expression today but I will talk about the techniques that appear to underlie the recent remarkable successes in understanding these networks which are called survey propagation and belief propagation. These are techniques which go beyond just finding patterns and give us a sense of how we can accurately create side information locally in a distributive way to tell us what is going on in complex networks.

Up to the level of 10^6-10^9 objects, the techniques that have been most successful have been those for solving well defined fixed problems such as finding optima. If you're a mathematician you always want to find a global optima, even if it is only one out of a very large number of almost global optima. If you're an engineer you want to know whether there is or isn't a solution better than some threshold of acceptability. We now know how to characterise a problem and solve it for very large scale systems, even up to about the billion elements. We know, for example, how to optimise the layout and predict the performance of microprocessors in large systems. We know how to send an army into a strange country and get most of the soldiers out again have most of the weapons arrive where they're supposed to, but it is the dynamism of large information networks that people are now trying to deal with. What seems to be happening is that an optimisation problem is becoming one of control, of managing a changing situation where the objective is not to find the best solution but to provide corrective information and directional information so that the system stays within an optimal domain of its phase space.

So what I'm going to talk about are new methods for optimal design on a scale of a billion or so and how the dynamics changes them. What is happening in the engineering of these systems is that they become more dynamic and they are essentially open. Using a physical analogy for modelling the circuits on a chip for example, can be carried out by imagining that they are atoms with springs between them and there is a classical physical method called simulated or virtual annealing. If you want to create a design for a billion circuits in a microprocessor you start with a list of what those circuits are and how they're connected to each other, which ones drive other ones and you imagine them as atoms connected by little springs. If the system has to run fast then the atoms have to be close and you can't have long wires across the chip. If the chip is to run cool you have to pull things apart and make room for the wires to go in between and so keep hot things away from each other. So there is a fundamental frustration between these two objectives. The cost of having things too close or too far apart can be written down as a function of the energy of the system.

It's an engineering problem so as soon as you find an arrangement of the atoms that is good enough you stop and make the resulting chip. And it

works. The trick that comes from physics and that makes it possible is to throw in simulated temperature, which is like sloshing things about with energy from outside. So physical methods such as modelling things as atoms with forces representing trade-offs is a sort of shortcut and a powerful technique in systems for which there is no easily recognised best solution. It may involve many compromises and the trick is to discover whether any of the compromises is good enough to proceed with an engineering process. These methods are being opened up in a surprising number of ways but they are limited. There are other ways of improving the optimisation method though they provide fewer proofs of whether they might work. Game theory is one and I'll talk about how that has completely moved away from physical analogies in a moment.

A system's properties are often characterisable by the extremes of its parameter space. If we put all the circuits in a microprocessor chip right in the centre, then the springs will be totally relaxed and that piece of the objective function will be totally satisfied but it will run too hot. We can either do a characterisation of the solutions or a characterisation of the problems. Designing a microprocessor involves a set of problems that focus a very specific goal but dealing with a logistical problem such as the best ways of using aircraft in an airline has a whole range of parameter and variable space in which problems may be explored and solutions found. If you talk about logistics problems there's the size of the problem such as the number of aircraft in the airline, there's the connectivity, there's the extent to which things have to connect to one another, and there are the constraints which couple the variables.

Earlier we had a discussion about layers which link together. If your aim is to get more people into your retail stores then where the stores are will be linked to how easily customers can get from one place to another. So constraints on one level affect constraints on another. Constraints may be heavy or light depending on the size and complexity of the system. The problem of say designing a strategy for getting to work is very small in a city if you have lots of highways and cars. On the other hand if you have only one bus and everyone has to go on it then that's the other end of the parameter space and it's obvious that it's going to take a long time for everybody to get to work. So whether people all have cars or whether they all have to go on the bus are two extremes that we can characterise: one of them is easily solved, the answer is 'yes' everybody gets to works on time. The answer to the question, 'Is everyone going to get to work on time?' for the bus situation is 'no' they are not going to get to work on time so let's see how long it is going to take them.

Thinking about situations which have the answer 'yes' to a question at one extreme and 'no' at the other extreme leaves open the question of whether we're going to go through a wide range of the answer being sometimes 'yes' and sometimes 'no' or is it going to change dramatically from 'yes' to 'now' at some value of the parameter in between. In simple systems that are studied in physics such as a substance's solid, liquid and gas phase states there are combinations of variables which will enable the answer 'It is a gas' or 'It is a liquid' but in between there is a phase change. That's something that occurs at a macro-scale and it's pretty dramatic but it tells us nothing about what is

happening at the molecular level. Engineering at the Avogadro scale is knowing what the molecules are doing at the micro-level.

Avogadro's number (6.022×10^{23}), is the number of molecules that it takes to give the molecular weight of a substance in grams. So when you have an Avogadro's number worth of stuff and its state goes from 'gas' to 'liquid' with a continuous change of a parameter it must do something dramatic in between. In that vicinity where it's hard to know whether the answer is going to be 'yes' or 'no', this kind of physical argument shows you that phase boundaries are where it's hard to tell the right approach to solving a problem.

Large complex systems are hard to deal with in the regime of phase change and often it is that regime we are most interested in wanting to apply automatic methods of planning. I'm not going to go into the details, but there are now theorems that give some confidence, though are not a complete answer, to understanding behaviour of large systems that go from 'yes' to 'no' across a parameter space, but approaching this problem is difficult for all algorithms we can think of. What I want to talk about now is some of the techniques which have been used in logistical problems. These are a set of randomised techniques that work extremely close to a well characterised phase transition and are my basis for thinking we can take the ideas and generalise them to a wider class of very large dynamic systems

Let me first explain what I mean by the term 'logically satisfied'. If you want to characterise a logistics problem which could be something as simple as making sure that aeroplanes or crews or spare parts all get to their destinations at the right time, you have a bunch of variables which might be about aeroplanes and their location and other things that go on the aeroplanes and so on. So you have a bunch of logical statements which I'll call 'clauses' because that's the term usually used. One clause on each of the constraints on the systems might be some short statement that a couple of things have to be true and then a positive outcome will be the sum, the logical AND of all the clauses of the constraints that have to be true. So then you can write down in normal conjunctive a solution to the problem that you're trying to solve. The difficulty will be that you have a huge long string of these little clauses and ideally all of the clauses must be satisfied for a successful solution. One of the cleanest ways around this is break the problem down with OR functions. You take statements about some little piece of the whole system and the requirement is that at least one of the statements has to be true. It's a three set model using Boolean variables and we have a large set of 'closers' which boil down to groups of three. So what makes it an analysable model is that we have every clause involving Boolean variables selected at random and each of them has half a chance of being required to be true and half a chance of being required to be false. So the problem is accessible to mathematical analysis, though we cannot really be sure that we haven't thrown out the problem we started with. What makes it tricky is that though if the number of clauses (N) is very small then we can have some confidence that we have a solution, where N is extremely large we can have one chain of clauses implying that X is true and another chain implying that X is false and there is no solution. We then have to start looking for approximate solutions with some minimal number of constraints of variables and this becomes a rich and interesting mathematical challenge.

In a simplified three set model a phase change occurs a little bit above four clauses for a variable, and a lot of powerful tools have been used to give insight into exactly how the phase change comes about. We find that solutions come in groups or clusters and that from one solution other nearby solutions can be found though there may be solutions not near the cluster. If the clusters are small and separated it's a different problem from when the clusters are large and you can move relatively easily from one solution to something else that a subsidiary measurement shows is preferable. We've learned a lot by this analysis about the relative ease or hardship in solving problems and whether or not there are gains to be made by redefining the problem.

I got interested in this model about 10 years ago working with Bart Selman at Cornell. We were studying 'three set' using the best algorithmic tools available at that time which were powerful search techniques that exhaustibly assigned values to variables and back tracked if a conflict was found. This meant that eventually you found a solution but sometimes you had to back up and start over, so they were slow. They could also prove that there was no solution if you backed up to where you began, showing you had exhaustively explored a piece of the phase space and proved there was no assignment of the subset of variables touched in your search that would satisfy the overall formula. We were able to solve problems involving approximately 300 variables and 1200 constraints and characterise them statistically to show all the characteristics of a phase transition. We could identify critical ranges of fluctuations and power laws. Finite size scaling, which is a powerful tool for recognising the power of the fluctuations, gave evidence of a phase transition, in finite size systems away from the point where the change actually occurs and the spread of these phenomena as systems get smaller is something we also know how to predict. We could show all this using the best tools of the early 90s, but methods have now come along which have pushed the boundary up from 300 variables. Faster computers enable us to deal with exponentially large problems and we now routinely solve problems of the order 10^7 or 10^8 variables (?) and we are getting good statistics on problems with a million variables. It's a heuristic method that uses distributive computation. What happens with survey propagation and belief propagation is that they take a hard problem of Boolean variables (1 or 0, up or down spin, aircraft present or not present etc.) and they replace it with probabilities, (e.g. the probability that the aeroplane will get to a particular airport) and they collect in a softer way, reasonable local assumptions from which solutions can be built. This appears to be a powerful technique that is generalisable.

You can find a solution by softening the Boolean variables in this way but a second approach is to use a random walk method to search around in variable space. It's a technique called 'Walksat' that is quite blind but there is some overall control which keeps it moving towards a possibly better solution. The traditional exact methods have a cost which is exponential with N, whereas the cost of the random walk technique is linear. There comes a point however at which simple linear methods break down, though over the last few years we have moved that point of breakdown very close to a phase transition.

What is needed from an engineering point of view is a means of constructing soft local estimators of what the variables are going to do. A typical way of softening the binary variable is to say that it can be 1 or 1 minus

the probability that it's going to be 0. Calculation of that is a long standing tradition in artificial intelligence called 'belief propagation' and the useful variation of that is finding variables that fall into several classes. There are classes of variables that are always 1 in a particular group or cluster of solutions and classes of variables that are always 0 and of course those which change. The tricky bit is if you set one right and another wrong you never get a solution. Still having separated the variables into the three classes you can then settle yourself into some space in which there is a subset of possible solutions. Ten years ago one of the complicated methods that I used was one in which the values of each variable was asserted one after the other and then you had to back the whole thing up when a mistake was discovered. But if you assign probabilities to a variable such as, 'almost certainly 1' or 'almost certainly 0' relative to a group or cluster of solutions you can transform the formula to something simpler. It moves the problem out to where the random walk method can take over because a survey propagation method will stop once it thinks it's found all the clearly assigned variables (all the variables that are always 1 and those that are always 0). In effect it says, 'the rest are sometimes 1 and sometimes 0 so why don't you go and ask my friend the random walk to help you find any of the resulting solutions'. The method works and I believe it is generalisable.

Someone asked about 'renormalisation'. In effect this transforming the problem to a simpler set of variables based on the insight that some of them have a clearer role than others. This is different from the change of scale envisioned in the kind of renormalisation that you use in physics though it's still a systematic transformation from hard problem to easier one. What I have described as 'survey propagations' are 'belief propagations' in a slightly richer space based on some assumptions about the nature of the solutions such as that they are clustered and that you decide on the things that can be carried out and those that should be left for later. The work I'm currently doing suggests that you can mix the two and come up with transformations which are a linear combination of survey and belief propagation methods and will if used carefully give you the best of both worlds. The surveys tell you things about solutions in particular regimes whereas solutions that cluster together as beliefs spread across a wider range of solutions. It an interesting question whether this method can be generalised to a wider range of problems in logistics and artificial intelligence simply by generalising the working equations.

OK, we've been talking about soft methods that build on speculations about the role of different variables and whether some are more important than others but there's one other technique that comes from physics which I think will be used more widely in the future and that is the 'cavity' method. Some problems are too complicated to solve by inverting the matrix. I mean a simple example of that is where you have and equation such as $ax = y$ and you know 'a' and 'y', you simply invert the equation as $x = y/a$ to solve it. Some problems are too complicated to do that even using iteration, so what we need to do is take some point, take some trial solution to the equation and then move it through the system to see if it works. However systems like the Internet involve networks that are like trees that branch of forever and sending a solution out on one path takes a long time even given the 'small world'

characteristic of the Internet today. Where you have recurring constraints it should be possible to send the solution out on one path and have it return on another. What the cavity method does is to focus on one point and ask what information brings the system to this point and what happens if we remove this point? So suppose we have a clause concerning this point or variable and a set of equations which results in that clause, then can we solve the set of equations that works in to that clause and most directly affects this particular variable and then pass it out to other clauses and do the whole thing again? So what this means is that there is a principled strategy beyond belief and survey methods that involves information in a cavity around a point and continually updating your estimate of what happens at that point. That is I think as far as we've got today using a physical analogy. Some of the ideas are 50 years old and some of them are still in flux, but I think we now know a lot about how to optimise at any one point in time, large complex systems by using 'iteration of soft estimate' methods. The cavity method can be made rigorous under some fairly sensible assumptions and is a powerful way of pulling things together.

But what about the Internet's growth? It has no well defined dynamic like a physical system. I mean in a physical analogy of a system, say of atoms connected by springs then if I move one atom with my finger, its motion will be felt by the atoms around it. These will react and their reaction will get back to the atom I moved. In other words I will feel the force of the other atoms backlog. That's not the culture of the Internet. Motivation is selfish. People try to discover as well as possible how best to attain an immediate objective and they go and do it. Although there's certainly a lot of altruism and people who try to be good citizens there are people who try to be bad citizens because it amuses them. And there are people in between, who just want to run a business and don't care who gets in the way. I don't think spammers, for example, are trying to bring the internet to its knees, they just think they'll make more money shortly before everybody stops using e-mail. The Internet has grown up over a period of some 30 years. Its engineering standards are set by the Internet engineering task force, its economic incentives are regulated by bodies in every country in the world but its growth depends on selfish imperatives. How could we reflect this in a model?

I'm going to sketch for you a really nice set of ideas that started with Christof Poppa Dimetrio Burton (?). It has been evolved by his students and is now studied by a number of people. It starts by making selfishness the core of a model for the physical network which consists of the links and wires along which the packets of information flow. The first question you have to ask is 'Who pays for a network to come into existence?' And the answer is individuals who choose to link themselves to other individuals with the plausible objective of sending e-mails to gather information from all places in the world in which they have an interest. So they pay for a piece of wire from the phone company and the service that goes with it. Once you've assigned the cost of building the network to the individual participants then each person is going to make decisions about adding links to their network (or getting rid of links) depending on whether it helps them accomplish what they're setting out to do. This is the extreme antithesis of the focus on energy that the physical method requires. I don't feel the other atoms pushing back when I make a decision to purchase a link in the Internet. So whereas in a physical network

system I may ideally get a global equilibrium based energy cost, for the Internet that would be replaced by a Nash equilibrium, which represents the point at which nobody has an incentive to change because the cost is an optimum for all people in it.

That's the outcome if we look at its evolution in terms of 'game theory' though it's often the case in game theory that a social equilibrium is not the same as a Nash equilibrium. Poppa Dimetrio suggested that the first thing to think about is the price of anarchy or selfishness, and that is the ratio between the sum of the cost incurred by the individuals at the Nash equilibrium and the cost of a social equilibrium which represents the best value for all people in the network. It is frequently the case in game theory that the social equilibrium is not the Nash equilibrium. There are lots of simple models that can show you what happens and it seems that it is a Nash equilibrium for the present Internet

The model that I'm going to talk about was refined from Poppa Dimetrio's ideas by one of his students and what we've learned is that in the worst scenario, the Nash equilibrium is some way away from the global optimum, but that focuses on the price of anarchy (the worst thing that would be stable) and the best that you can do is probably an inappropriate characterisation because under plausible assumptions both the selfish equilibrium and the global equilibrium are quite sharply distributed though their ratio is much closer to each other than the worst case analysis would suggest.

So here's the model. Each link is purchased only once at a cost of X dollars. Once purchased everybody can travel in either direction across the links and each site makes decisions to purchase or get rid of links based on the cost of the links plus the sum of all distances and there are various models for distances. It's easiest to think about distances being the number of hops you have to go over but there are more realistic models that people are starting to use and if a site is not connected and the distance is taken to be infinity you might want to avoid that. You might assume that everybody buys no more links than they have to and because they make their decisions asynchronously they're not really able to care very much about optimising the whole. They're just doing business as well as they can.

It's the cost of links that will determine the kind of configurations that you get. There is a bound for small cost links and a bound for large cost links. If the cost of a link is small enough the model is that every pair of members of the network has a link between them. In other words if the links are cheap enough everyone will buy as many as they think they might need and pretty soon someone has bought every possible link. This is then the situation where all distances are one and is an optimum for distance. The other extreme is when the links are extremely expensive and the best solution is a star configuration. Links are purchased between one site and all the other sites and that's a pretty good solution to the distance problem. What actually happens in modelling this decision making process (and a similar thing that happens in global optimisation with finite temperatures) is that something more interesting than the completely linked configuration and a lot more robust than a star tends to happen when you assume selfishness with limited information.

The practical problem in simulating a Nash equilibrium is that each site has to decide whether to add or remove N links, so it has 2^N situations to

consider. A more plausible model is the simple adding or subtracting of one link and that's something that probably could be assessed by the individual owner of a site in the time that he has to make the decision before somebody else changes the situation around him. So I think that's a good model. We augmented it in a way that is a little bit more powerful and not significantly more expensive and that was to consider trading one link for another. So allowing that replacement move you get very good solutions. Typically you get the global optimum as the Nash equilibrium except in the intermediate range of moderate strength where you get these open stars. What's happening there is that there's a core of three sites replacing a single site in a star and every site makes a link to two of them. So the graph is two bonds between the three, or sometimes three bonds between the three in the centre and you can get from any one point to any other point in no more than two steps.

Overall in real terms the world has spent twice as much money on links in the Internet as it would have done if it had built a star but the payoff is that we've got a far more robust system. Remember the discussion in the first talk about how in scale free networks it's easy to attack their most connected points and blow them to bits? That of course is something that needs to be defended against . I don't think we want networks for any engineering purpose that are as fragile as the star and so one question that emerges is whether we need to devise strategies to force networks to go against their selfish imperatives and build in redundancy. In global optimization I can get things like this by allowing everybody some flexibility to occasionally spend a little extra money to purchase extra links and the end result will be that there are always multiple paths between any pair of sites. It's a much more reliable system because of redundancy against local failures.

It also happens when you have limited information, even with this strong kind of selfish model, you don't need to cook the mechanisms of the game theory in elaborate ways because the nature of real processes are that people will not play an ideal game and as a result it leaves you with more redundant and reliable systems. But when costs are high some sites may be quite happy to be just a few steps from a centre which carries a large amount of traffic. So we could have a stable system with just a moderate level of branching in which the cost is separated in an acceptable way though it might have a single point of failure. If we compare global optimization with selfish optimisation what we find is a cumulative distribution of Nash equilibria. What we can show for a group is that sites that have a pair of neighbors will also have a high probability of a pair of paths between any other sites in that group and therefore achieve a redundancy that gives stability. Where global optimization preserves robustness, then all but a small fraction of the sites may have a pair of neighbors and therefore two ways to start the path between themselves and any other site. If we have three neighbors there's the chance of three independent paths between a pair of sites and so this would be a case where the selfish optimization model now fails quite badly. So there are some interesting questions in the things that lead to robustness and redundancy in these models.

What I would like to suggest is that there are at least two different ways to achieve redundancy: one of them is through a group plan, a consensus that everybody will manage themselves to include redundancy as part of their

optimization. But what is more likely to work is a model simulating human nature and imperfect information and a finite time to make decisions. Things therefore are not perfectly optimized and the less optimized solutions would appear to give better overall network properties.

Now my interest in trying to measure, manage and understand the edges of the network was that I started with a map of the Internet which was like a tree. This approach was one of searching outwards from a single point and finding a whole galaxy and keeping track of the path to all the other known points. That approach gives a tree, but the Internet is not a tree it is a mesh. At the AS level there are shared links between different service providers which allow all sorts of messages to flow sideways without having to go through Washington DC. The services that I think are going to transform the Internet and form a more personal environment are services like 'peer to peer' delivery. We'll get a spreading out to regular non scale free net workers exploiting the links around the edge of the network. Our objective in Evergrow is to go out there and find those links, and mapping exercises have begun to show up the dark matter in the Internet. But a further question to ask is how things like wireless technologies are affecting the density of connections at the edge of the Internet. These are going to have a big influence on the kind of services that we deliver for information networks around the edge of the web in the next five or ten years. So we can try as hard as we can to come up with mechanisms that make people behave but the things that will characterize the Web are the finiteness of information horizons, the finiteness of decision times and the fact the Internet is changing so fast that we can't block it for very long.

I said I would talk about Avogadro scale engineering but I participated in a workshop where it was said that we were going to talk about Avogadro scale engineering. However my conclusion at the moment s that nobody wants to do what I would think of as Avogadro scale engineering and that is tackling the web and tackling large information networks in all of their local complexity. What people want to do at the moment involving 6×10^{23} molecules is to make medicines on a large scale using cellular manufacturing processes. That's aiming at one product which is put into capsules large enough to sell as a medicine. And quantitative computing is using very complicated machinery to answer one problem at a time. So the nano-scale approach to things we might do in the future is, at the moment, focused on a single outcome and it's not scale free. I also think the biomedical context is a cultural force here which cannot accept the risk of a wide range of outcomes. This is because if what you're making is pharmaceutical you've got to get it right, because you can't make medicines that kill people even if eventually Darwinian pressures will eliminate them from the process that you're trying to design. So there's a certain risk aversion in the early attempts to do Avogadro scale manufacturing. As a result of that it's being engineered in a way that is not what you would call flexible or large scale logistic problem solving. So I don't think we have to worry about the Avogadro scale engineering right now.

Questioner 1: Have you considered in your network model, the case where different nodes have different utility for their links and that heterogeneity expresses itself in heterogeneous networks?

Answer: Yes, heterogeneity is considered. The first place you put it is in distances. I mean saying that all distances are 1 is obviously making no differentiation, but if you want to think about peer to peer services at the edge of the network, one way to do it is to come up with a chunk of real network or a plausible model of real network with delays and then place your sites at spots in the real network and ask them to find pairs in the underlying network and share information with them. That makes the distance problem far more realistic and we're learning a lot from tackling problems in that way. The second thing is that there are two things I would change in the individual incentives model: the first is you might want to think that a link is a partnership between the two ends and the cost is shared and I think you'll see immediately more rational decisions made, or I should say more group favored decisions. And the second thing is the cost of making a link ought to have some relationship to the number of links already present at the destination site. Both of those have been factored into the simulations we've done.