Part 2: nature-inspired design for artificial systems

Saffre & Halloy, 2005

Plan of the presentation

- **Introduction**
- **Complex(?) networks**
- **Swarm-based IDS**
  - Context and basic concepts
  - ADDICT (demo)
- **Self-organised Service Orchestration**
  - SelfService
    - Concepts and tools
    - Pervasive SelfService (demo)
    - Extended SelfService (demo)
  - Embryo
    - Background
    - Latest news (demo)
- **Conclusions**

ULB

BT

Introduction

Saffre & Halloy, 2005

What is this all about?

- We are seeing a headlong rush towards a world stuffed full of smart devices, ambient intelligence and pervasive computing.
- In this world, coping with physical constraints is less of a challenge than making sense of the total mess that is the “network economy”.
- The business community is aware of this: it has a compelling vision of what *can* be achieved, but it is very confused when it comes to *realise* that vision.

ULB

BT

Saffre & Halloy, 2005

Take-away message

- The vision is: all of that "smart stuff" should be able to help us provide useful services that meet end users' demands, in real time, when and where they arise.
- The truth is: almost everybody accepts that, but very few people have actually started thinking about *how* to make it happen!
- The solution is to embed enough autonomy (self-*) into the "smart stuff" that it can organise itself into an "*Adaptive Service Ecosystem*" - that adjusts automatically and invisibly to changes in demand and policy.

ULB

BT

Saffre & Halloy, 2005

Expected benefits

- More agile computing assets, capable of responding adaptively to unique, changing and unpredictable user demands.
- More robust software, capable of self-diagnostic and of actively and autonomously seeking to avoid “unsafe” configurations.
- Reduced cost of ownership (i.e. a direct and highly desirable consequence of increased robustness).
- Reduced “downtime” (ibid).

ULB

BT

Saffre & Halloy, 2005

Practical applications

- Service deployment: module-based applications could greatly benefit from “on-the-fly” adjustment to unpredictable usage patterns (i.e. “who needs what service, where and when?”).
- Resources accounting and allocation: “on-demand” utility computing (i.e. seamless Grid) requires real-time balancing of the offer and demand, which could be achieved via unsupervised negotiation between potential collaborators.
- Self-organising ad-hoc networks: social differentiation (specialisation) and/or decentralised radio spectrum management (e.g. via cross-inhibition) can enhance usability and/or longevity.

ULB

BT

Saffre & Halloy, 2005

Autonomic principles

- The trend towards self-configuration, self-protection etc. championed by IBM is widely referred to as “autonomic computing”.
- Though finding its origin in “pure” research (biologically inspired systems), it has gained so much momentum and widespread endorsement that many implementations now exist.
- However, they are mostly “node-centric” (as opposed to “network-centric”), which means that they do not explicitly take into account group dynamics.
- This is potentially a serious flaw, as applying autonomic principles creates the perfect conditions for complex system behaviour (many interacting units making autonomous/selfish decisions on the basis of locally available information).

ULB

BT

Saffre & Halloy, 2005

Self-organisation

- *By definition*, a system composed of units making autonomous decisions based on locally available information can only be "driven" to a desirable state via self-organisation.
- This requires engineering the reasoning and decision-making engine running on individual units so as to promote the emergence of the "right" collective behaviour.
- In turn, this means adapting the predictive techniques of *natural* complexity science (both analytical and numerical) to meet the needs of *artificial* complex systems designers.
- It seems relatively trivial *in principle*, but experimental validation requires prototype implementation, which is a serious issue.

ULB

BT

Saffre & Halloy, 2005

Key tasks/milestones

- Identify relevant and specific causes of complex behaviour in artificial systems (e.g. in dynamic networks, especially overlays).
- Gain a thorough (i.e. not “anecdotic”) understanding of how they combine to affect global response.
- Learn how to use this improved knowledge to make probabilistic predictions about the evolution of complex artificial systems.
- “Reverse-engineer” the process leading to desirable system state(s) to “discover” the right local rules.

ULB

BT

Saffre & Halloy, 2005

Why we (the Telcos) care about AC

- **It's an opportunity:** autonomic computing has the potential to revolutionise services via self-organisation of software components.
- **It's a threat:** we want to mitigate the risk that network operators are left with bandwidth as their single asset/product.
- **Bottomline:** we are very keen to contribute our expertise to the development of autonomic ICT solutions, be recognised as key players in the field and, ultimately, have a share of the corresponding market!

ULB

BT

Complex(?) networks

Saffre & Halloy, 2005

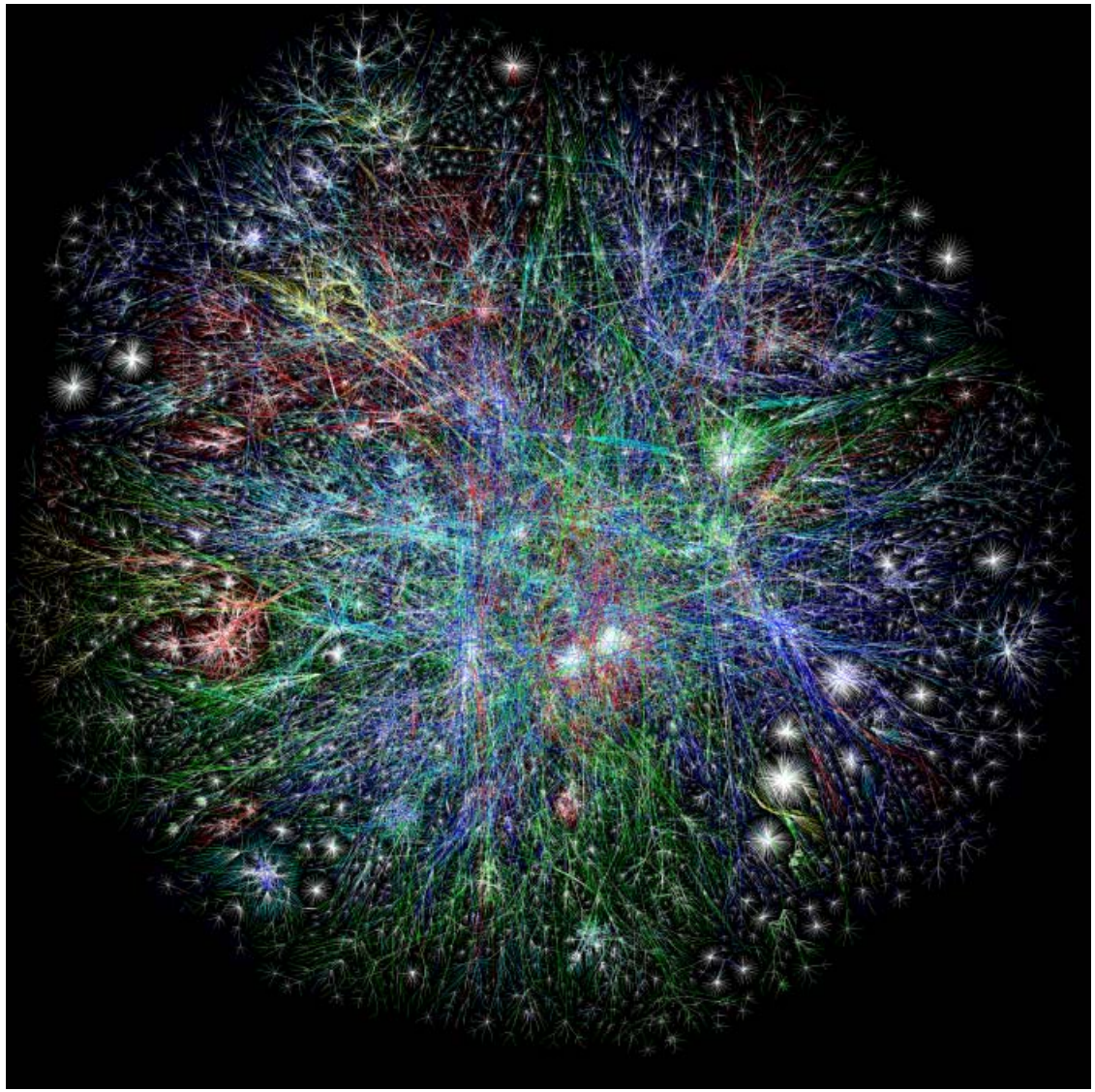
Are *today's* networks complex?

- Only some of them!
- Complex doesn't just mean large and complicated, it means exhibiting non-trivial global behaviour as a result of *unsupervised* local interactions between system constituents.
- In many ways, engineers are trained to "fight" complexity, i.e. to constrain system behaviour and find ways of enforcing central control.
- And there is also some measure of confusion in so-called "complex networks" science.

ULB

BT

Saffre & Halloy, 2005



http://www.nd.edu/~networks

Saffre & Halloy, 2005

Allow me to explain…

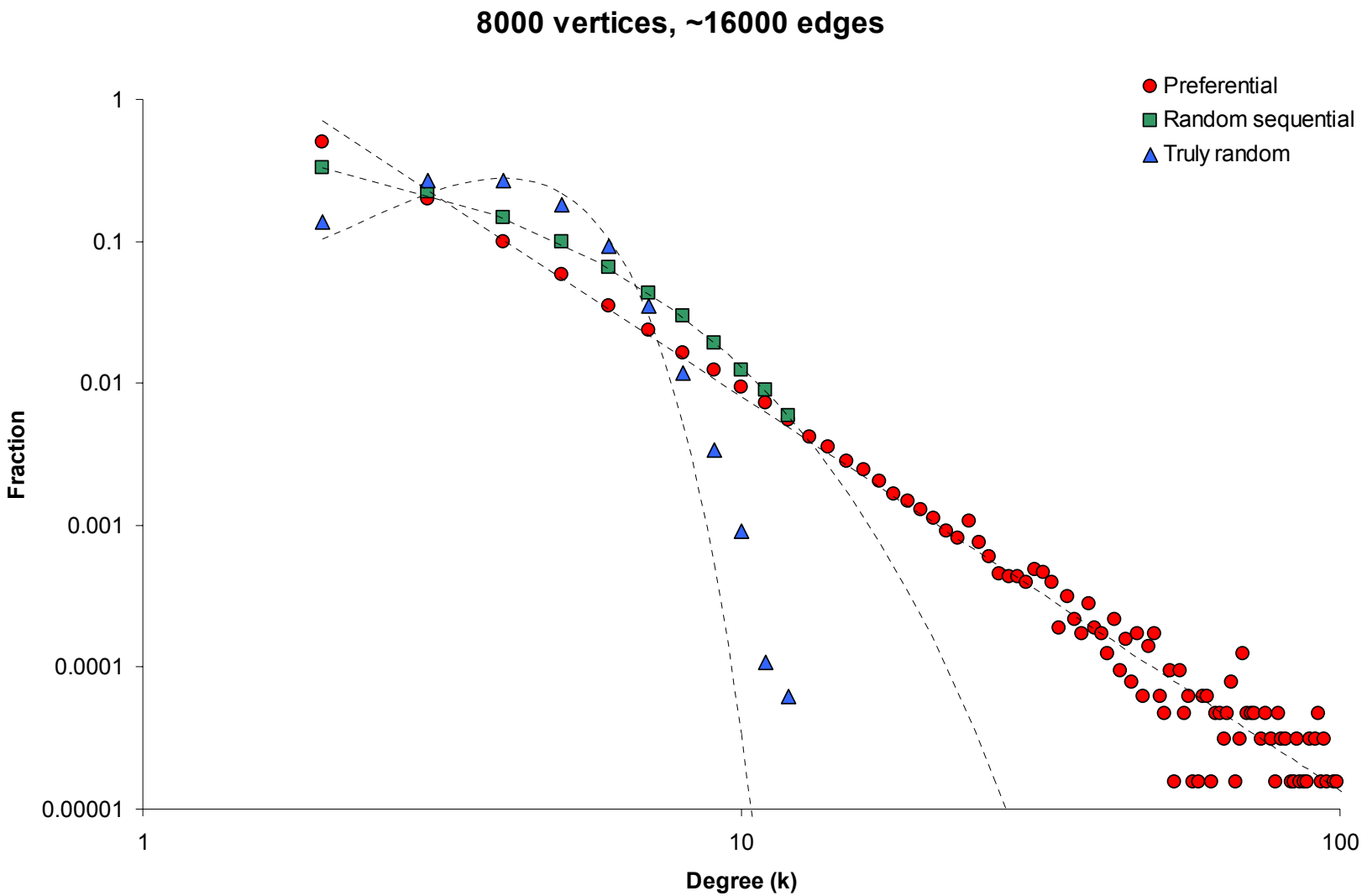
- A lot of so-called “complex” networks are only marginally so!
- Admittedly, it is possible to promote (and maintain) some global topological properties through local decision-making, which amounts to a form of emergence.
- However, many models make a lot of (hidden) assumptions!
- For example, the famous “preferential attachment rule” only generates scale-free topology if growth is sequential and newcomers have complete information on network state.

$$P_{i,n+1} = \frac{k_i}{\sum_{j=1}^{n} k_j} \neq \frac{k_i}{\sum_{j=1}^{N} k_j}$$

ULB

BT

Saffre & Hallov, 2005



ULB

BT

Saffre & Halloy, 2005

But more importantly…

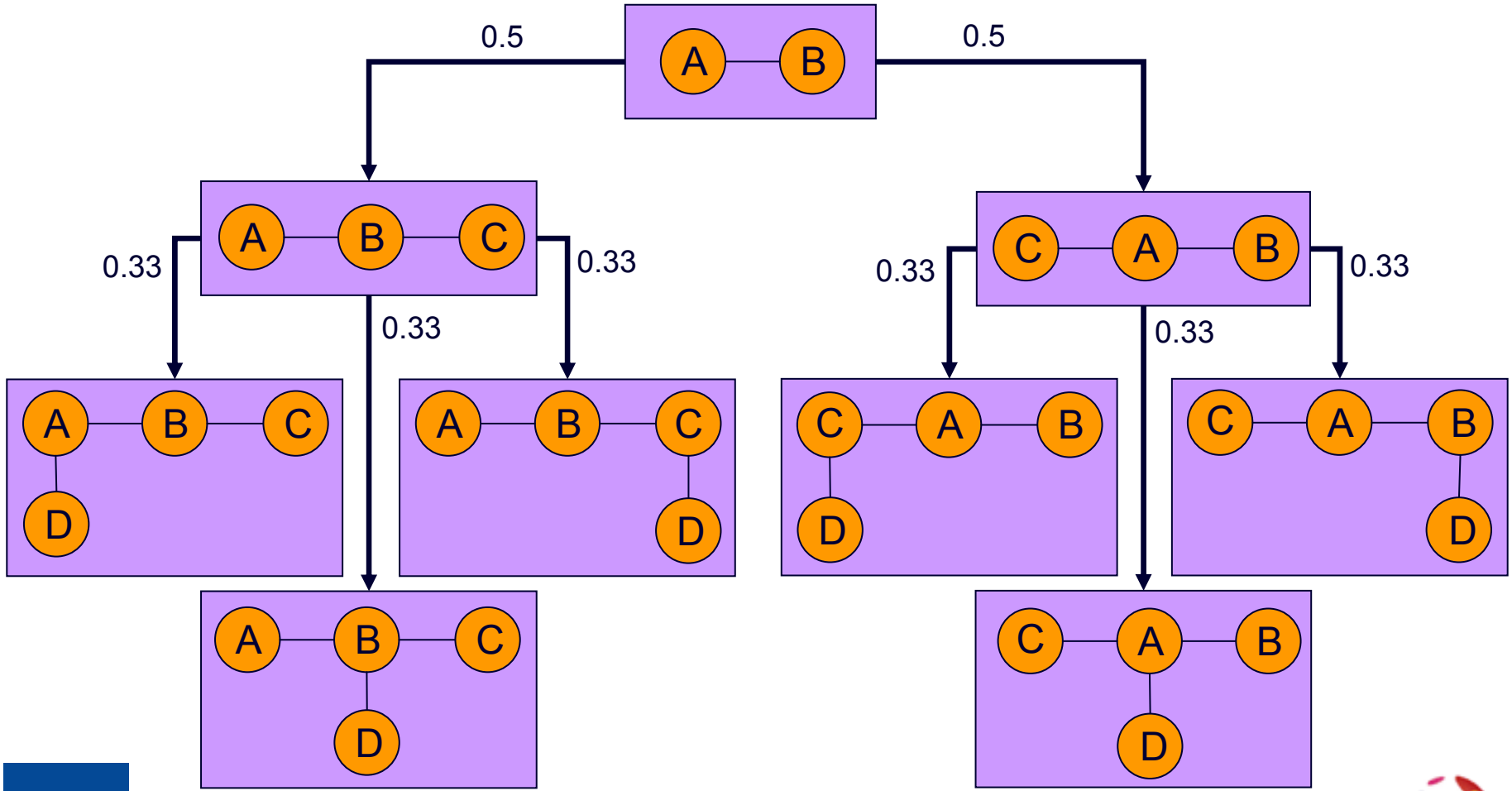
- There's nothing "magical" or even "surprising" in the way these degree distributions emerge.
- Generally speaking, once local rules are known and interactions understood, complex systems are eminently predictable (from a probabilistic point of view).
- As far as complex networks are concerned, global properties can be thoroughly explained by applying good old combinatorics, as they merely reflect the probability distribution of having a given degree.

ULB

BT

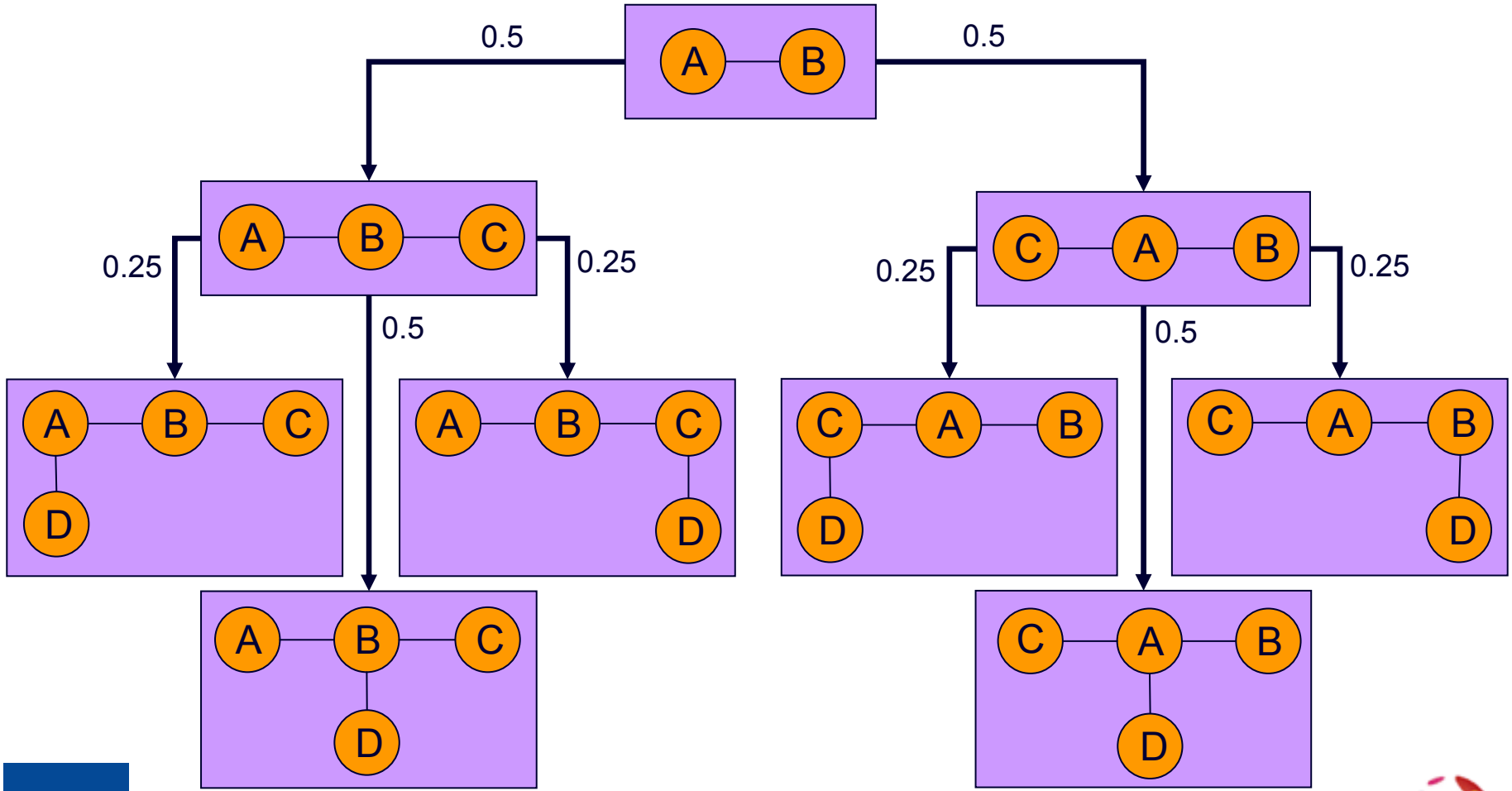
Saffre & Halloy, 2005

Example: random sequential



Saffre & Halloy, 2005

Example: preferential sequential



ULB

BT

Saffre & Halloy, 2005

Result:

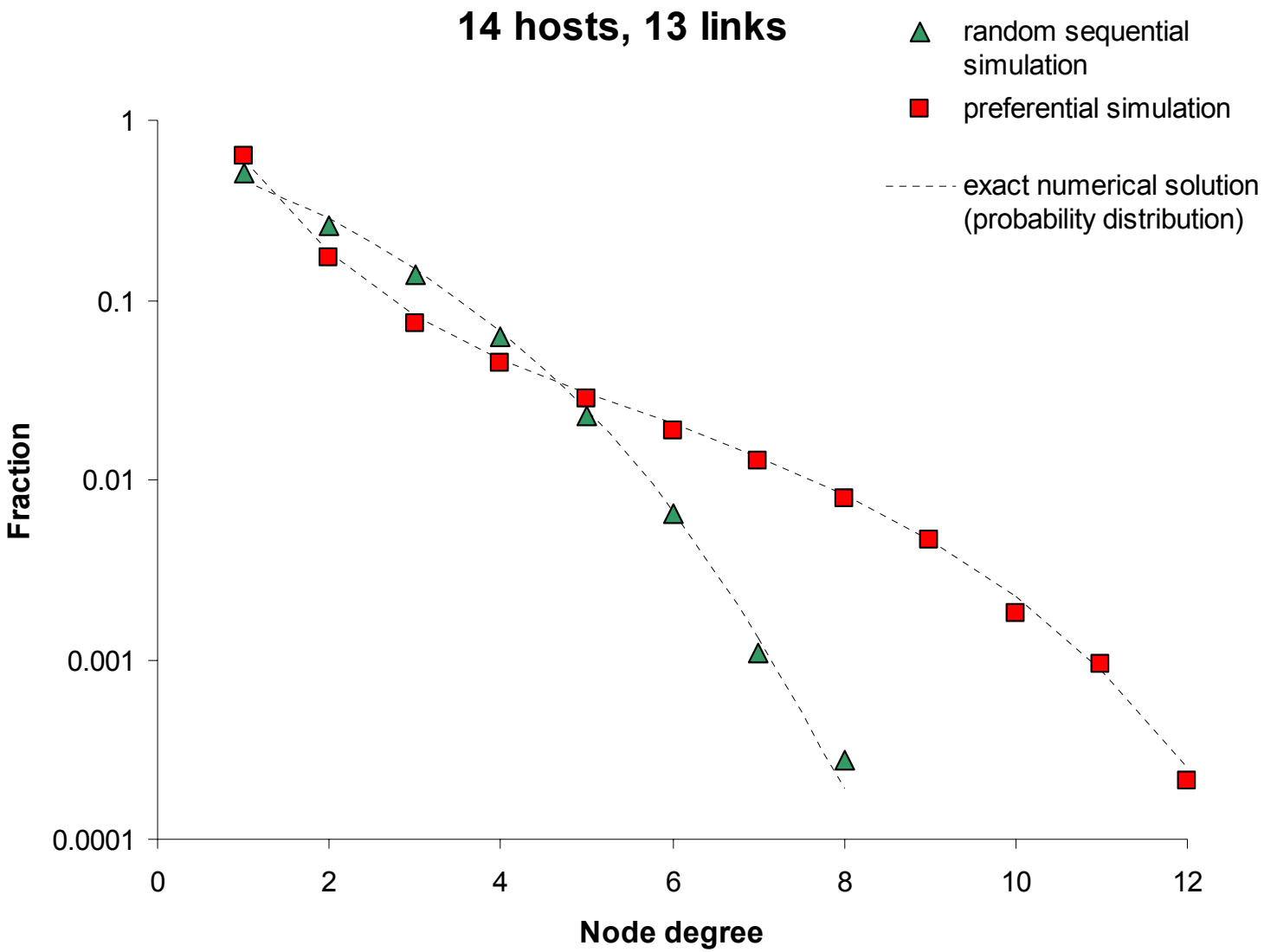
k_A	k_B	k_C	k_D	random	preferential	random	preferential
3	1	1	1	0.17	0.25	0.33	0.5
1	3	1	1	0.17	0.25		
2	2	1	1	0.33	0.25	0.67	0.5
2	1	2	1	0.17	0.125		
1	2	2	1	0.17	0.125		

“Fat tail” starts here…

ULB

BT

Saffre & Halloy, 2005



ULB

BT

Saffre & Halloy, 2005

Will *tomorrow's* networks be complex?

- Very likely!
- Networks are becoming so dynamic and complicated that the only viable management option is to *make* them complex…
- Because we have no choice but to gradually switch from centralised to decentralised control, we are effectively sowing the seeds of complexity.
- We must learn to live with and take advantage of the emergent properties arising from the interaction of many system constituents, not try to counter them.

ULB

BT

Swarm-based IDS

Saffre & Halloy, 2005

Intrusion Detection and Response

- Key topic in network security!
  - How do you know that you're being attacked?
  - How do you identify opening breaches?
  - When intrusion is in progress, how do you contain the threat?
  - Can you devise a system-wide collective response that will outrun the attacker?
- HIDS (Host-based Intrusion Detection System)
  - Scalable, but tends to miss macroscopic attack patterns.
- NIDS (Network-based Intrusion Detection System)
  - Detects macroscopic attack patterns…
  - But typically not in real time (~ "forensic" tool)!

ULB

BT

Saffre & Halloy, 2005

One solution could be (loosely) based on mimicking nest defence

- Contemporary networks are like a termite mound…
- Pretty hard to break into by probing at random…
- But permanently under repair and/or undergoing transformations.
- In short: there's always a weak spot!

ULB

BT

Saffre & Halloy, 2005

“My LAN is my castle…”

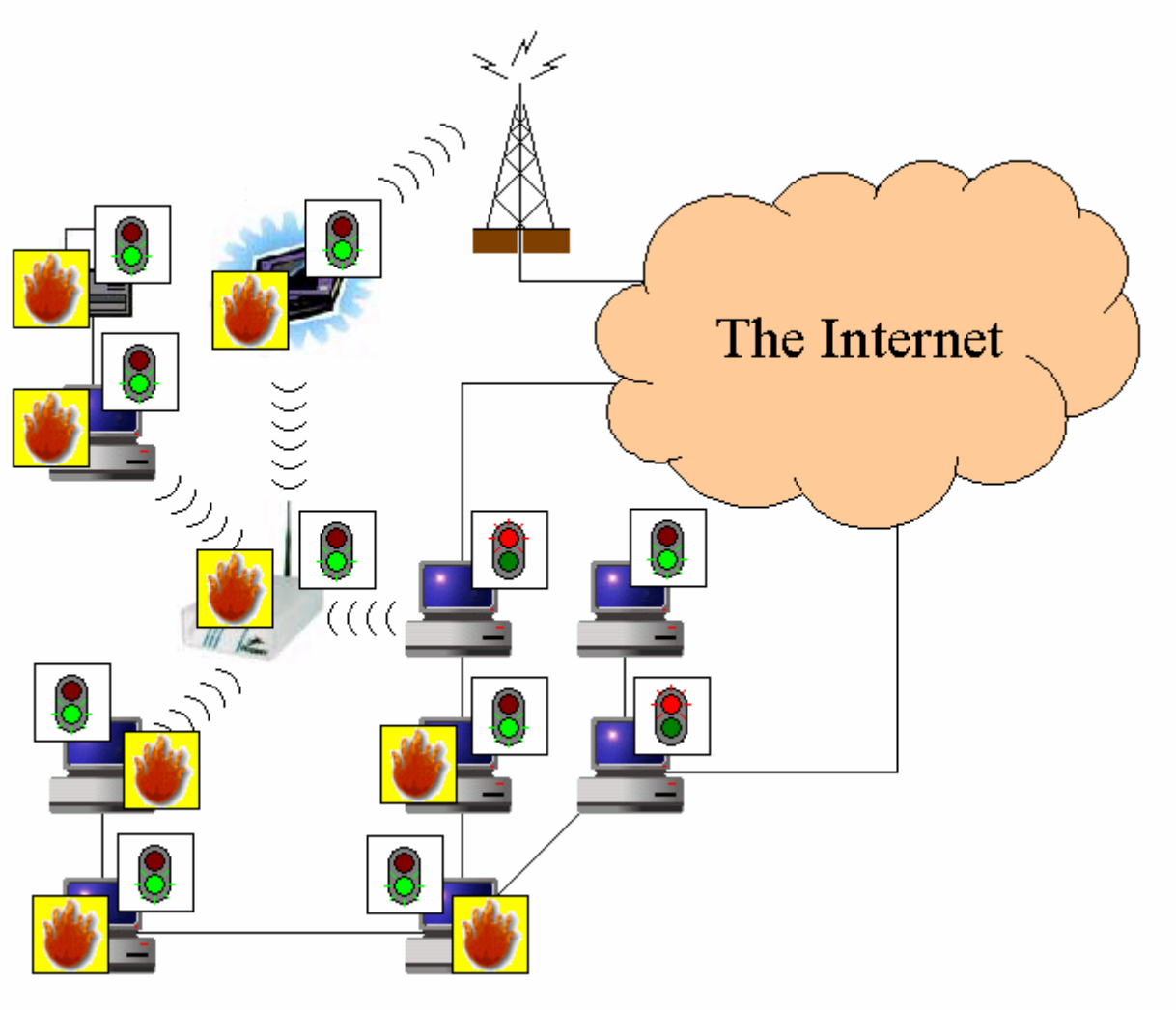
- The problem with dynamic architectures is that they can’t be efficiently protected by static fortifications.
- The rapid take-up of “plug-and-play” and wireless access points has created a volatile situation, sometimes referred to as “the disappearing perimeter”.

ULB

BT

Saffre & Halloy, 2005

The disappearing perimeter



Saffre & Halloy, 2005

“Once more, onto the breach…”

- How do termites react to a similar situation?
  - Soldiers run to fill the gap.
  - Workers seal the compromised area, protecting the deeper chambers.
- This requires a way of (collectively):
  - Detecting a breach in the outer wall.
  - Building a new defensive layer to protect the “inner sanctum”.

ULB

BT

Saffre & Halloy, 2005

In network security terms, that means:

- Detect abnormal activity
  - Port-scanning
  - Repeated (failed) log-in attempts
  - Illegitimate requests from authenticated users/hosts
- Take actions to circumvent the intruder
  - Update security stance of personal firewalls on exposed hosts
  - Revoke legitimate privileges from misbehaving users
  - …

ULB

BT

Saffre & Halloy, 2005

Drawing inspiration isn't copying

- Mobile agents (i.e. the most obvious analogy) aren't a suitable way of enforcing security in contemporary networks.
  - They are just too "heavy" and too slow…
- In clear:
  - A "wall" = a host on the highest ("paranoid") security stance.
  - A "breach" = any unknown user/host, unusual traffic, forbidden process…
  - "Workers", "soldiers" = non-existent!

ULB

BT

Saffre & Halloy, 2005

ADDICT
Adaptive Defence for Dynamic ICT

- The best (only?) way to conduct in-depth intrusion detection and response in a dynamic context is to consolidate host-based monitoring data across the network.
  - Combines the advantages of HIDS & NIDS (scalability + ”bird’s eye” view).
  - Individual hosts share “pre-digested” information *locally*.
- Inhibitory signalling
  - If you’re only interacting with identifiable, well-behaved and “happy” devices, relax.
  - If (some of) your neighbours are identifiable, well-behaved, but “worried”, be suspicious…
  - If (some of) the devices that you’re interacting with are unidentified and/or misbehaving, be worried!

ULB

BT

Saffre & Halloy, 2005

Definitions and model

- “Identified”: probably means that a “colony tag” (i.e. collective, encrypted signature?) is needed.
- “Well-behaved”:
  - runs (nothing but) authorised applications and up-to-date security software…
  - only makes requests that are appropriate for this “colony member”.
- “Beacon”: encapsulated identity/behaviour data + alert status (sent periodically).

ULB

BT

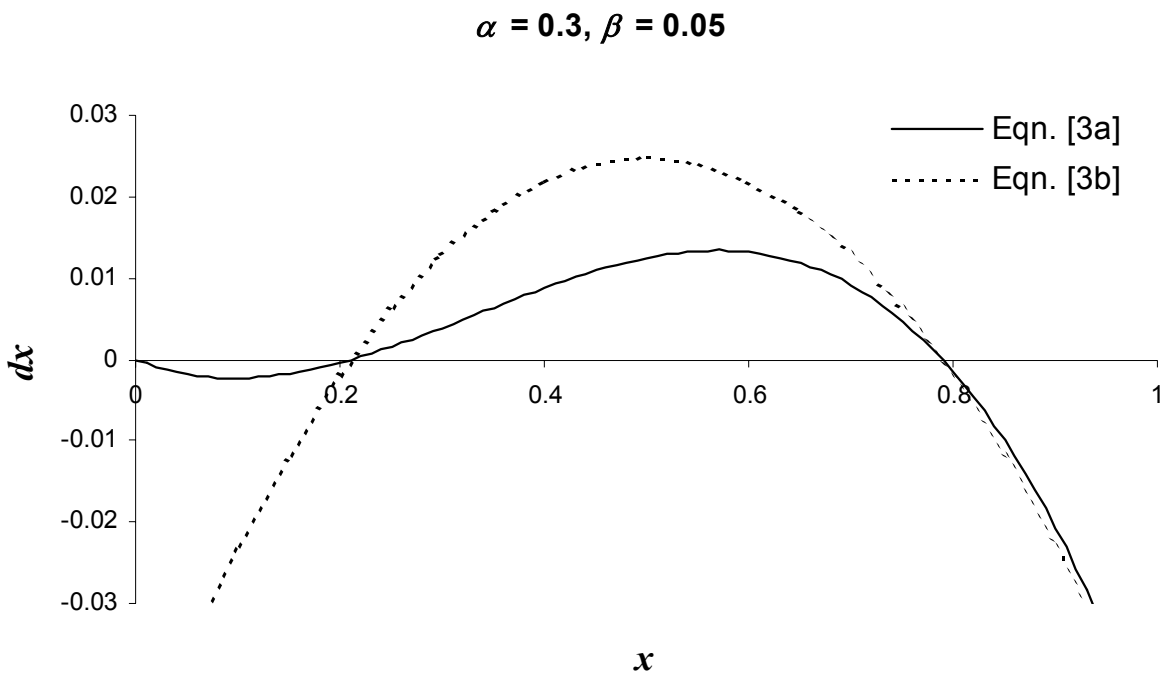
Saffre & Halloy, 2005

Definitions and model (2)

$$\frac{dx}{dt} = \frac{x(1-x)}{N}\left(N - n + \alpha \sum_{i=1}^{n} x_i \right) - \beta x$$

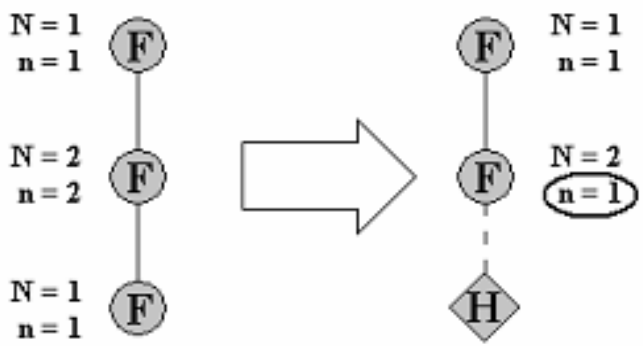
Saffre & Halloy, 2005

Analytical solution - example 1 “all friends” scenario



Saffre & Halloy, 2005

Analytical solution - example 2 “one friend, one foe” scenario



- Already slightly more difficult...
- Solution obeys:

$$\frac{dx}{dt} = x(1-x)\alpha y - \beta x = 0$$

$$\frac{dy}{dt} = \frac{y(1-y)}{2}(1+\alpha x) - \beta y = 0$$

That is:

$$y = \frac{\beta}{\alpha(1-x)}$$

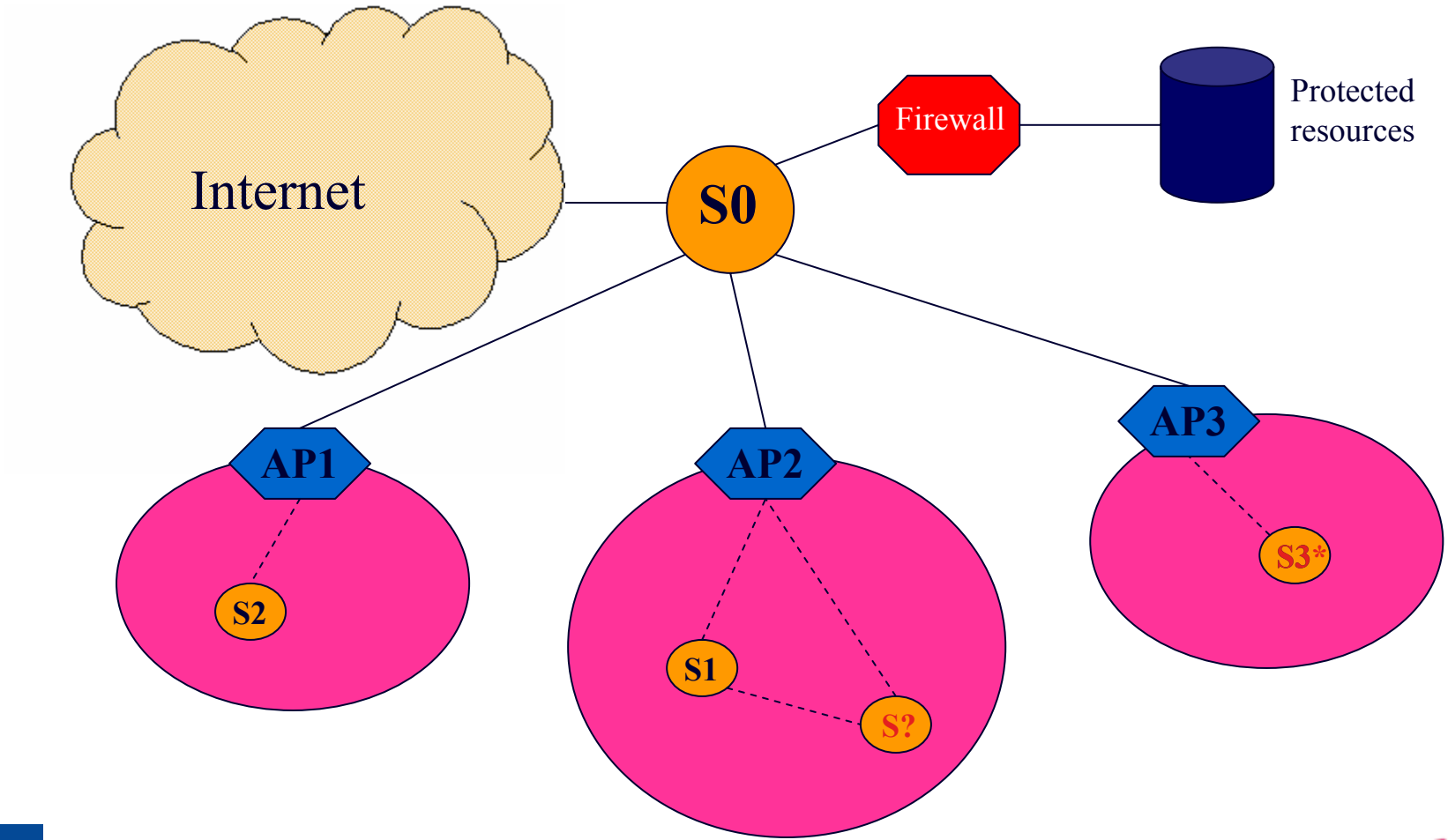
$$x = \frac{\alpha + \beta - 1 \pm \sqrt{(\alpha - \beta + 1)^2 - 4\beta(\alpha + 1)}}{2\alpha}$$

ULB

BT

Saffre & Halloy, 2005

Application scenario: Wireless LAN



Saffre & Halloy, 2005

Without ADDICT

- When unknown station S? attempts to connect to access point AP2:
  - It either succeeds (open system, null authentication) or fails (shared key).
  - In the first case, S? can at the very least free-ride on the Internet.
  - In neither case is S1's behaviour modified by S?'s presence.
- When authenticated station S3 starts to "misbehave" (→S3*):
  - Nothing happens!

ULB

BT

Saffre & Halloy, 2005

With ADDICT

- In essence, more room for flexibility…
  - Even if S? is denied access to the WLAN, the attempt will trigger an alert signal on AP2 and the warning will be passed to S1.
  - If S? is voluntarily granted access (e.g. AP2 is in an area where legitimate visitors are common), you may still want to temporarily build up the security settings on S1.
  - In either case, the actions taken could prevent (in)voluntary damage caused by the intrusion (eavesdropping, virus infection, sniffing…)

ULB

BT

Saffre & Halloy, 2005

With ADDICT (2)

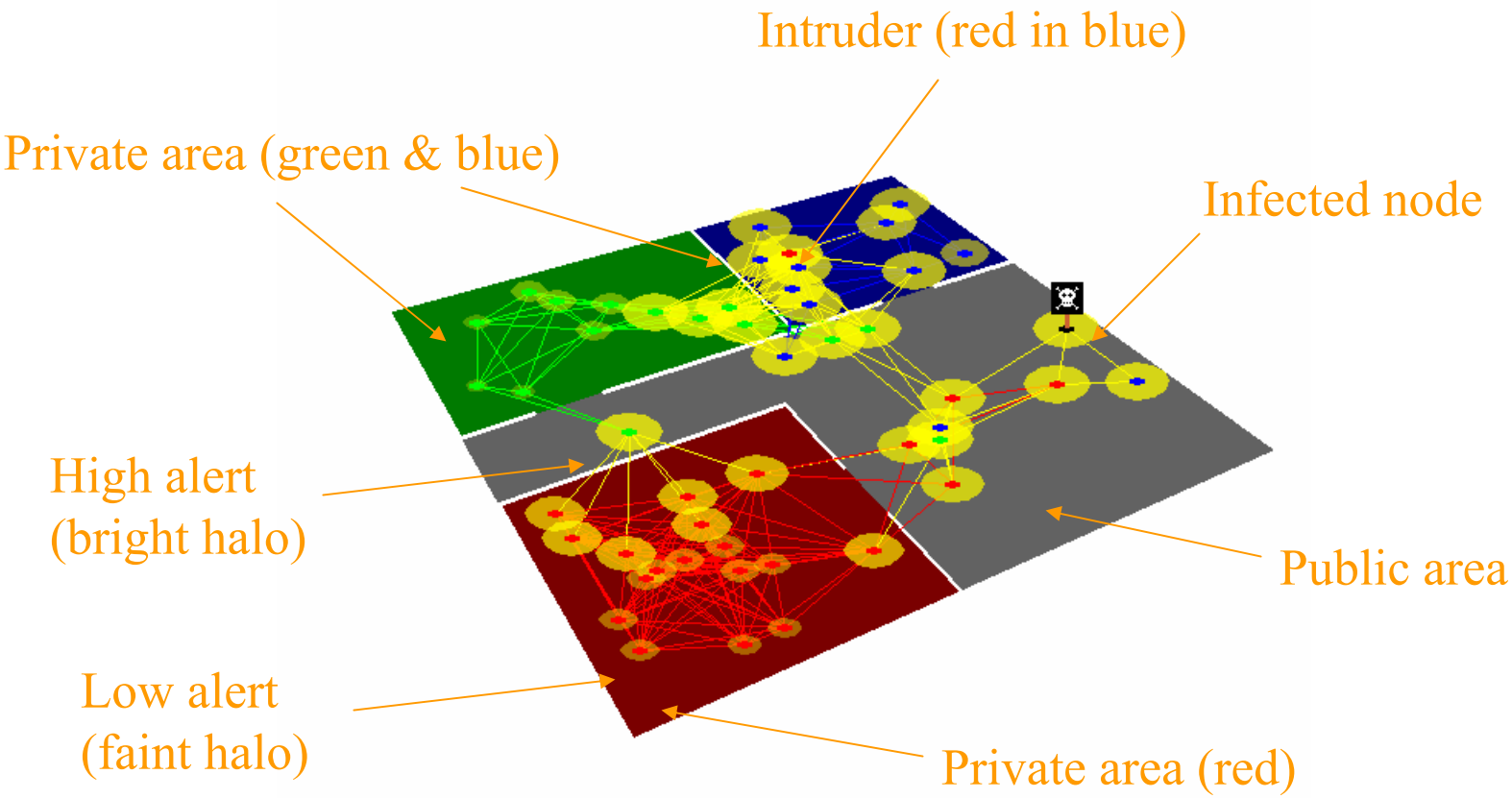
- Dealing with S3→S3*
  - Could be (e.g.) because S3, after successful authentication, turns on a forbidden application.
  - Shows up in the beacon signal.
  - Even if S3* is impersonating S3 (stolen authorised ID), this could expose the forgery.
  - Again, it triggers an alert signal on AP3, which can propagate through S0.
  - Actions could involve (e.g.) neighbouring APs starting to use stronger encryption/authentication.

ULB

BT

Saffre & Halloy, 2005

The ADDICTed office



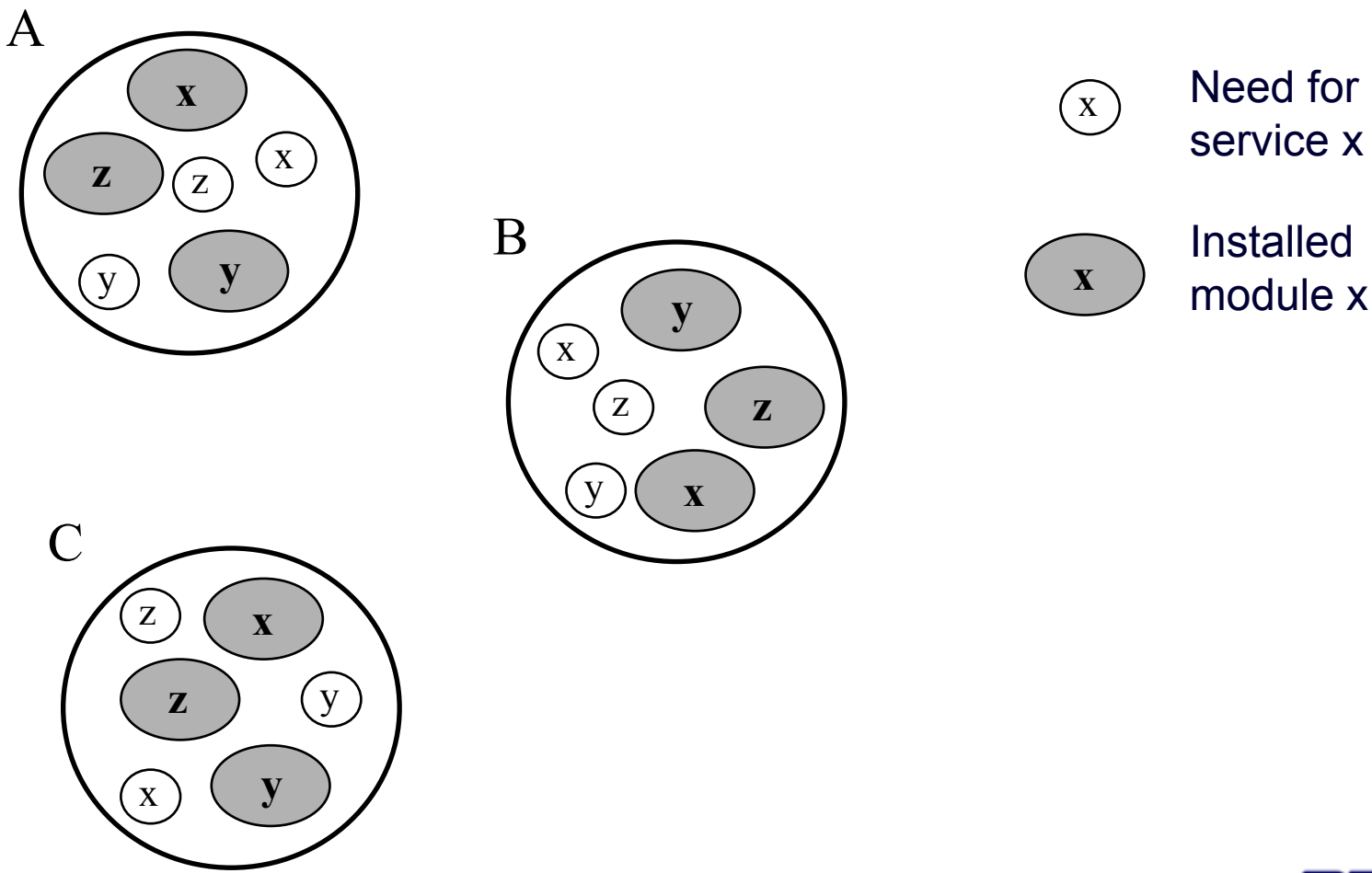
ULB

BT

Self-Organised Service Orchestration

Saffre & Halloy, 2005

“Alone in the world”



Saffre & Halloy, 2005

Ups and downs

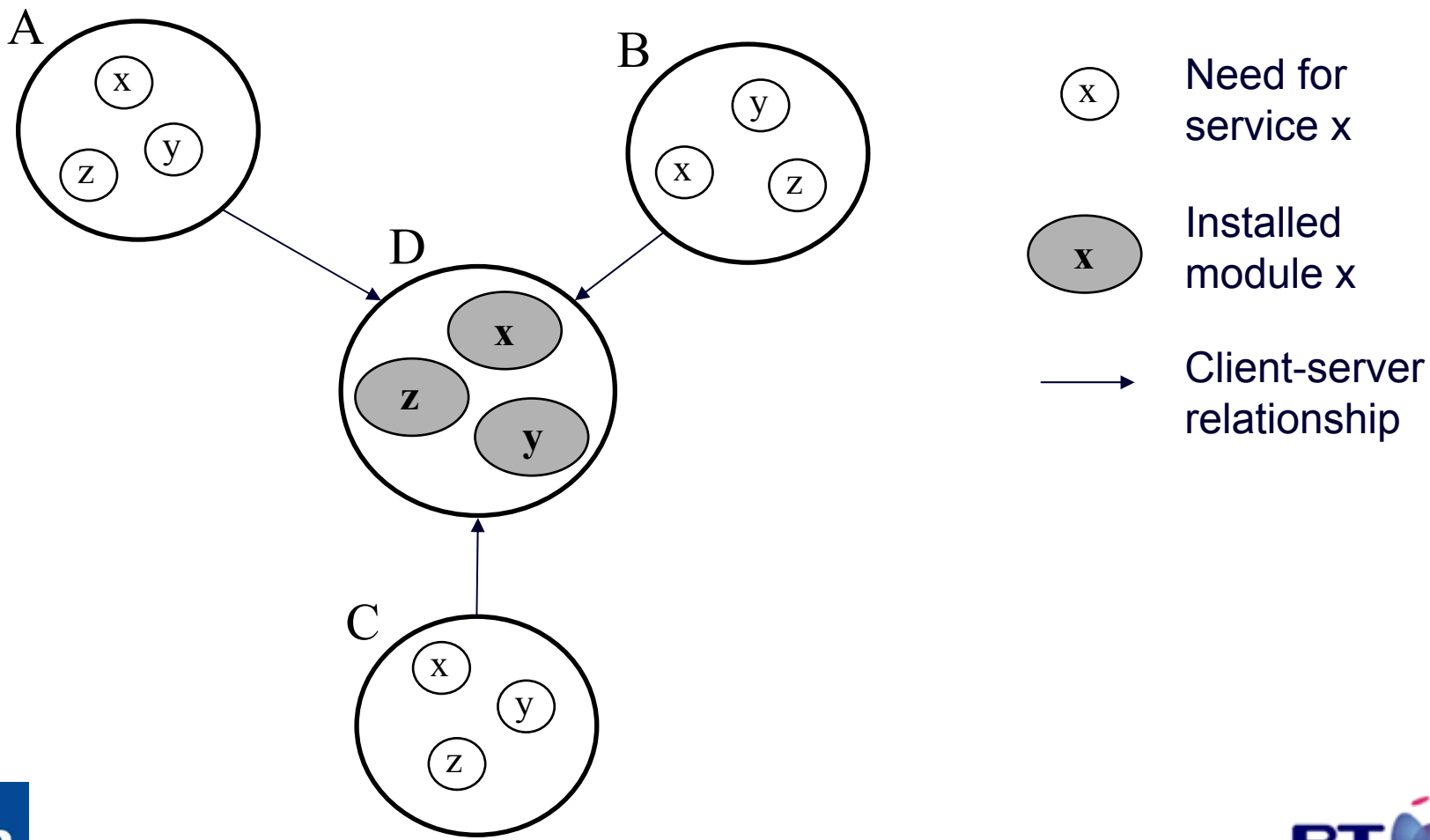
- Highly robust to node or network failure.
- End user has total control.
- Need a lot of onboard power (good for hardware manufacturers!).
- Need many copies of every application (good for software manufacturers!).
- Need virtually no ICT infrastructure (*not* good for service providers!).
- Amazing waste of resources (“99% idle time” syndrome).

ULB

BT

Saffre & Halloy, 2005

“Thin client”



Saffre & Halloy, 2005

Ups and downs

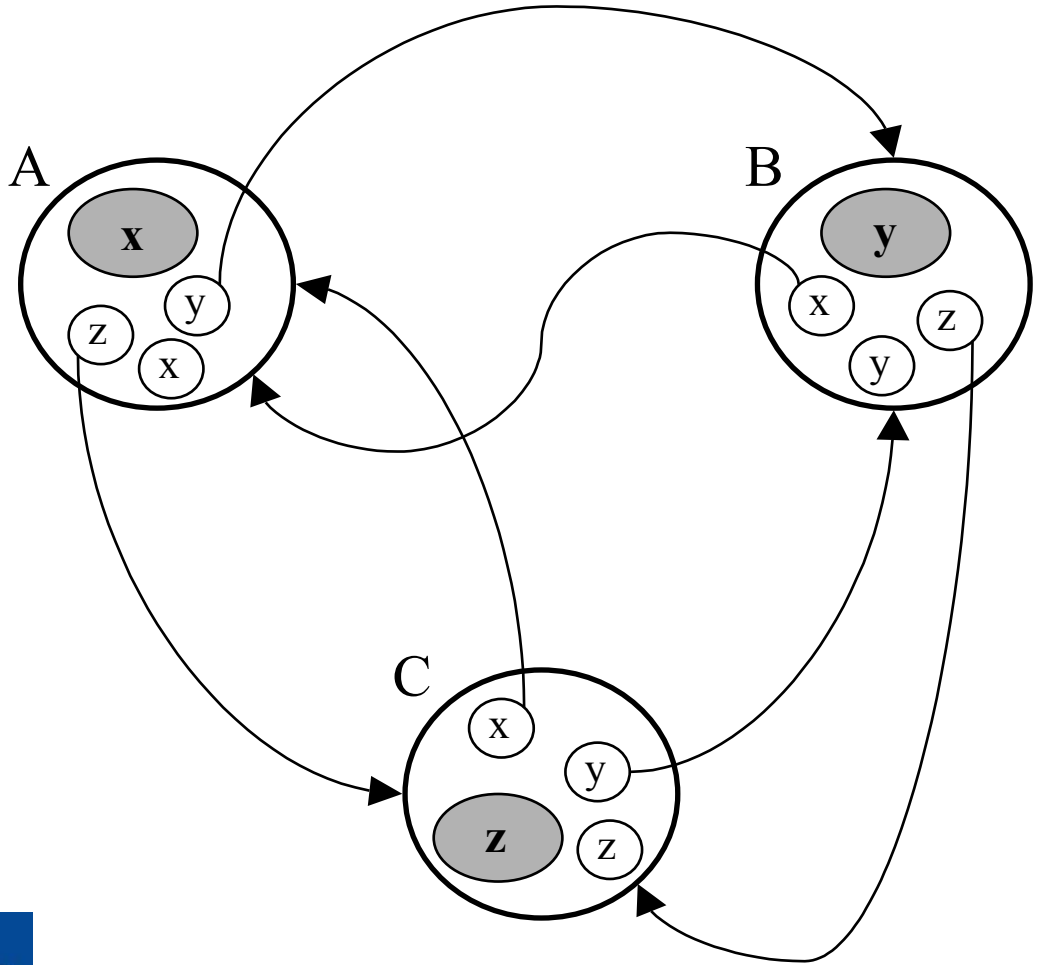
- Extremely brittle (single point of failure!)
- Administrator has total control.
- Mixed picture for the hardware industry (need powerful servers, but only low-end PC's)
- Mixed picture for the software industry (depends on license management).
- Mixed picture for ICT providers (network services are paramount, but risk of bottlenecks and QoS degradation).

ULB

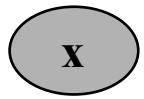
BT

Saffre & Halloy, 2005

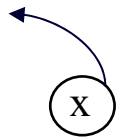
“SelfService”



Need for service x



Installed module x



Service-specific client-server relationship

Saffre & Halloy, 2005

Ups and downs

- Intermediate robustness (no single point of failure, but problems will tend to be “non-local”).
- End user is back in charge (decides what to install).
- Advantages to the hardware/software industry?
- Interesting model for sharing resources (i.e. P2P utility computing).
- A clear step towards pervasive ICT and a great opportunity for service providers, *if* we can deliver!

ULB

BT

Saffre & Halloy, 2005

Why is that a challenge?

- The difficulty is of course to ensure adequate service coverage, in terms of accessibility, reliability, latency etc.
- This has to be achieved without central control or planning, otherwise:
  - It won't scale.
  - We'll lose many of the benefits (in terms of agility and adaptability).
- *In the IT community, there is a chronic lack of interest for preliminary studies on system properties!*

ULB

BT

Saffre & Halloy, 2005

"SelfService"

- Objectives:
  - To support reliable, fault-tolerant access to a sub-set of services, handpicked by users on an individual basis.
  - To reduce the need for installation/running of the corresponding software modules on local devices used as access points.
  - Without having to rely on dedicated servers.
- Underlying hypothesis:

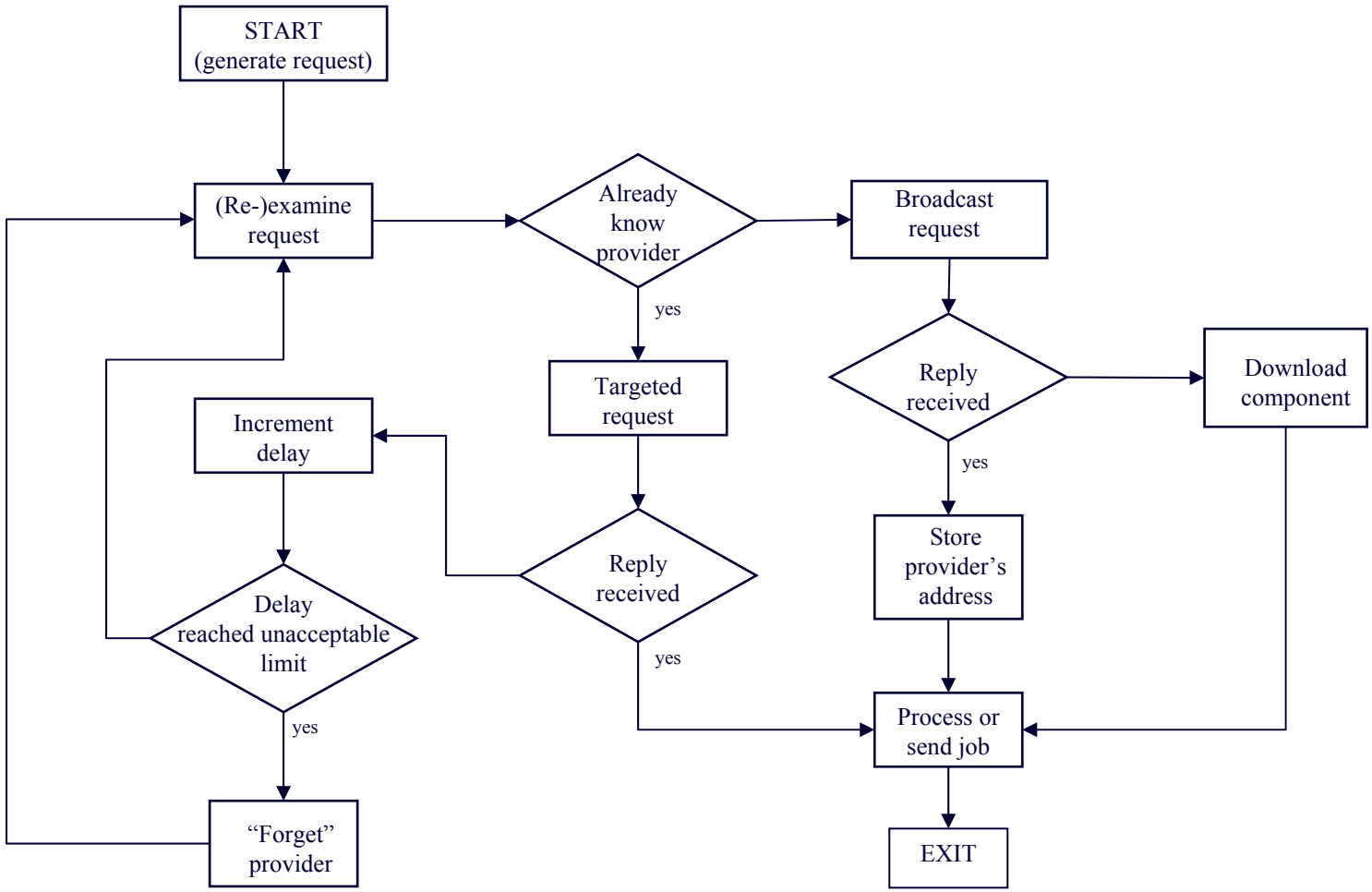
  There are unpredictable but consistent patterns of activity, which can be used to select stable partnerships (e.g. "device X, *hosting* service S1, is able to *provide* it to device Y for 80% of business hours").

ULB

BT

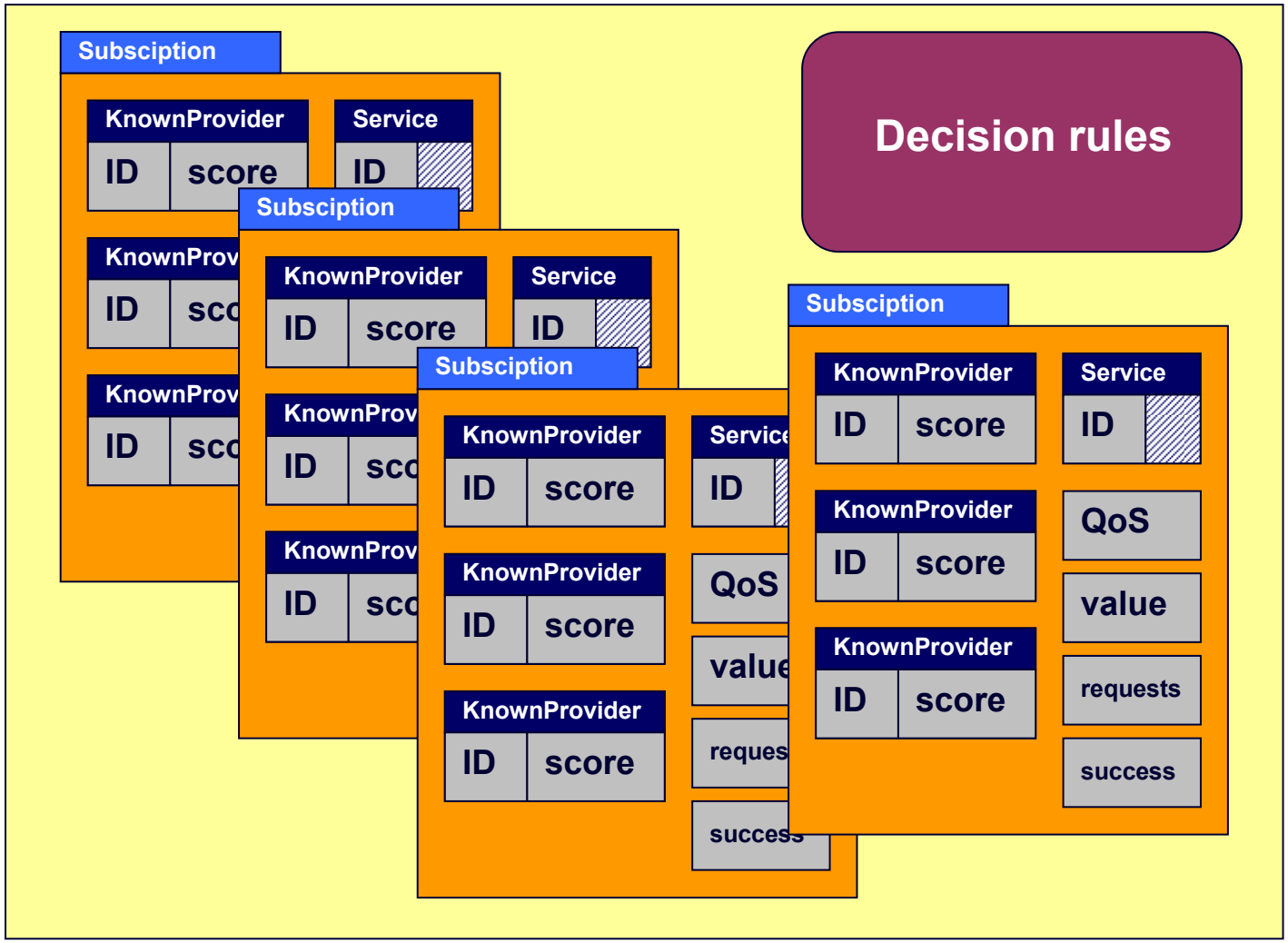
Saffre & Halloy, 2005

Experimental algorithm



Saffre & Halloy, 2005

Locally maintained information

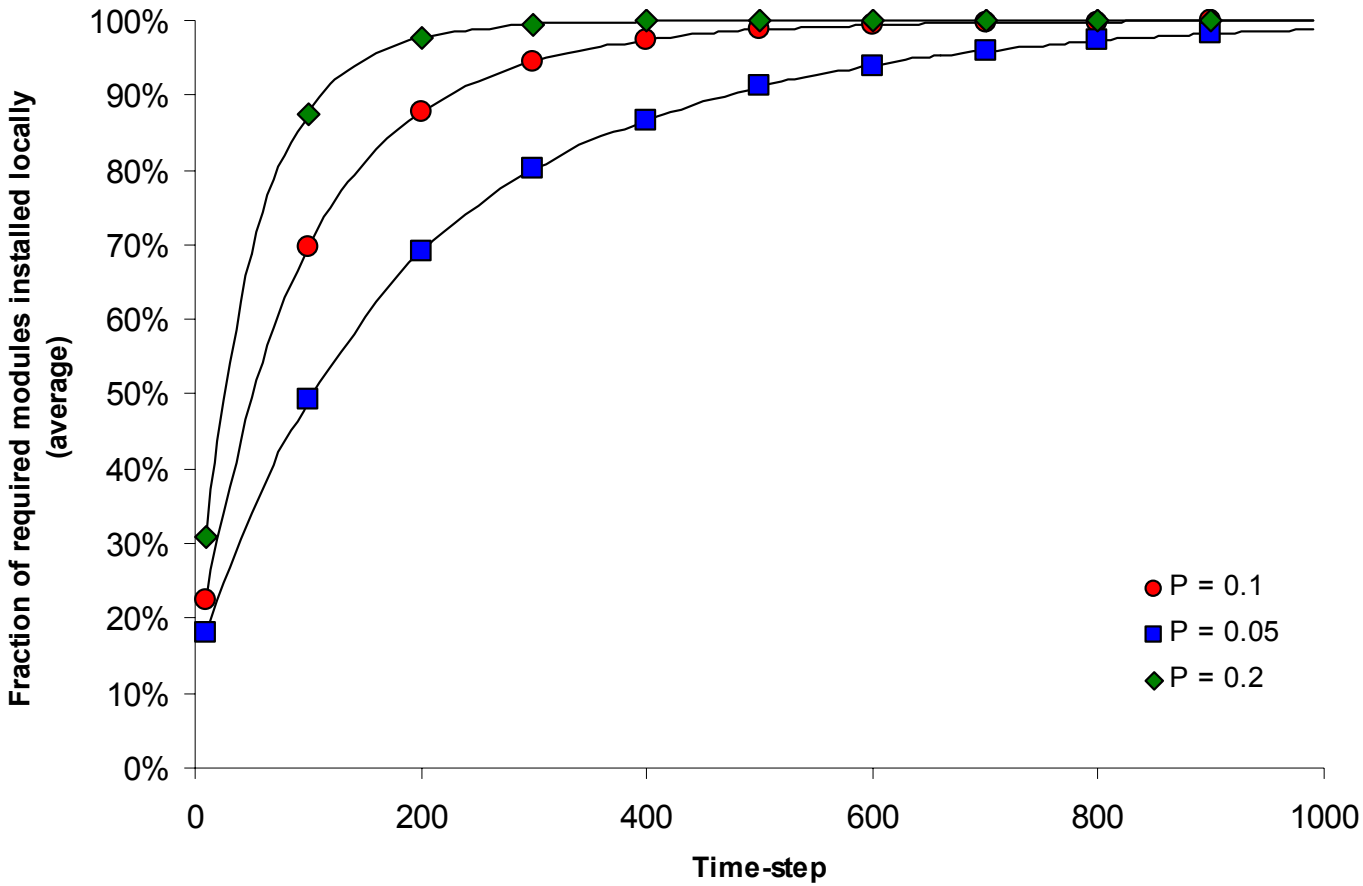


ULB

BT

Saffre & Halloy, 2005

“Alone in the world” (benchmark)

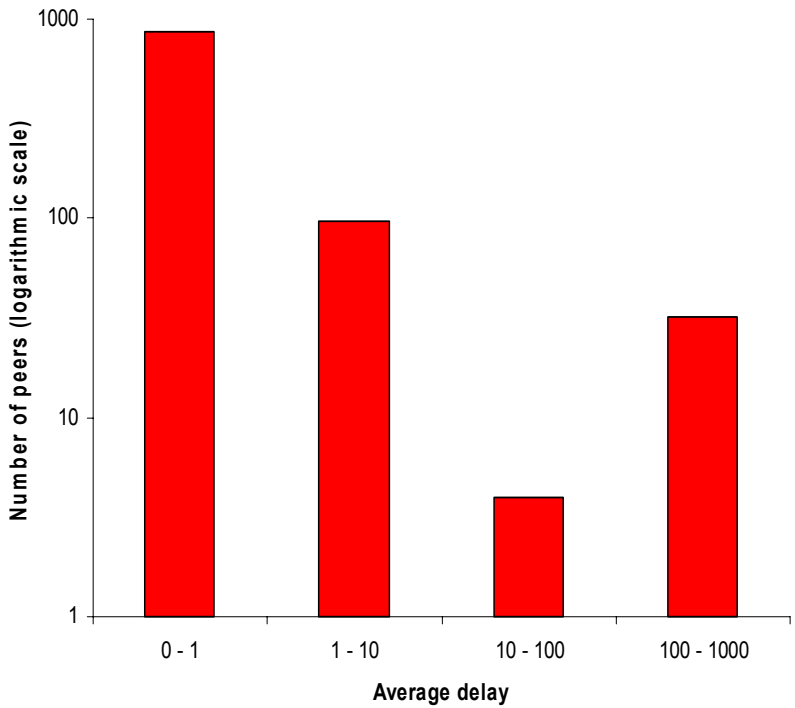


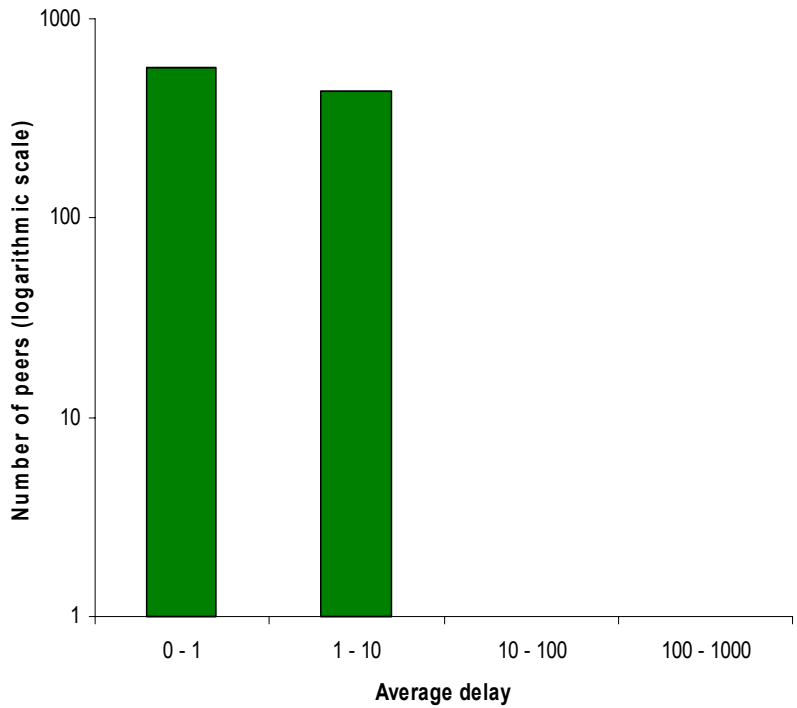
ULB

BT

Saffre & Halloy, 2005

Load balancing



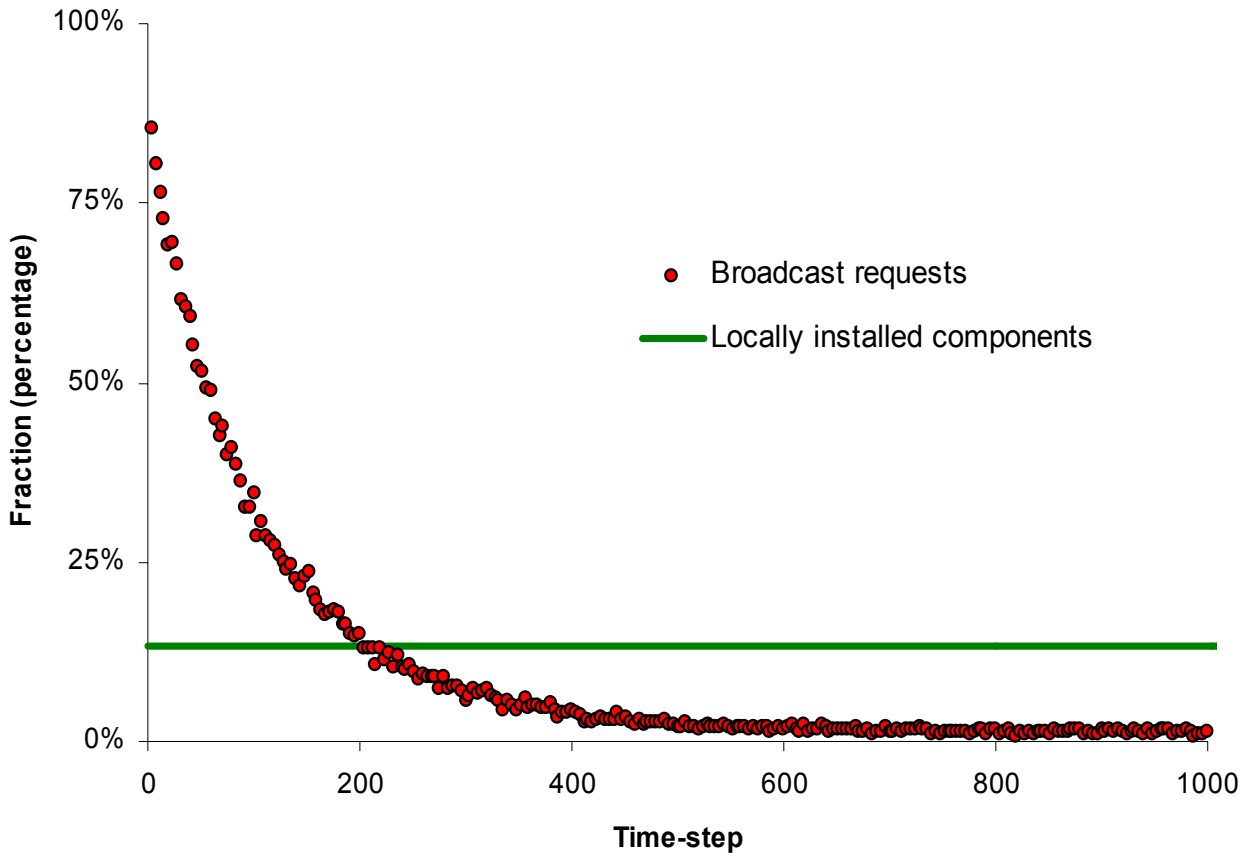


ULB

BT

Saffre & Halloy, 2005

Broadcasts & downloads (~bandwidth)

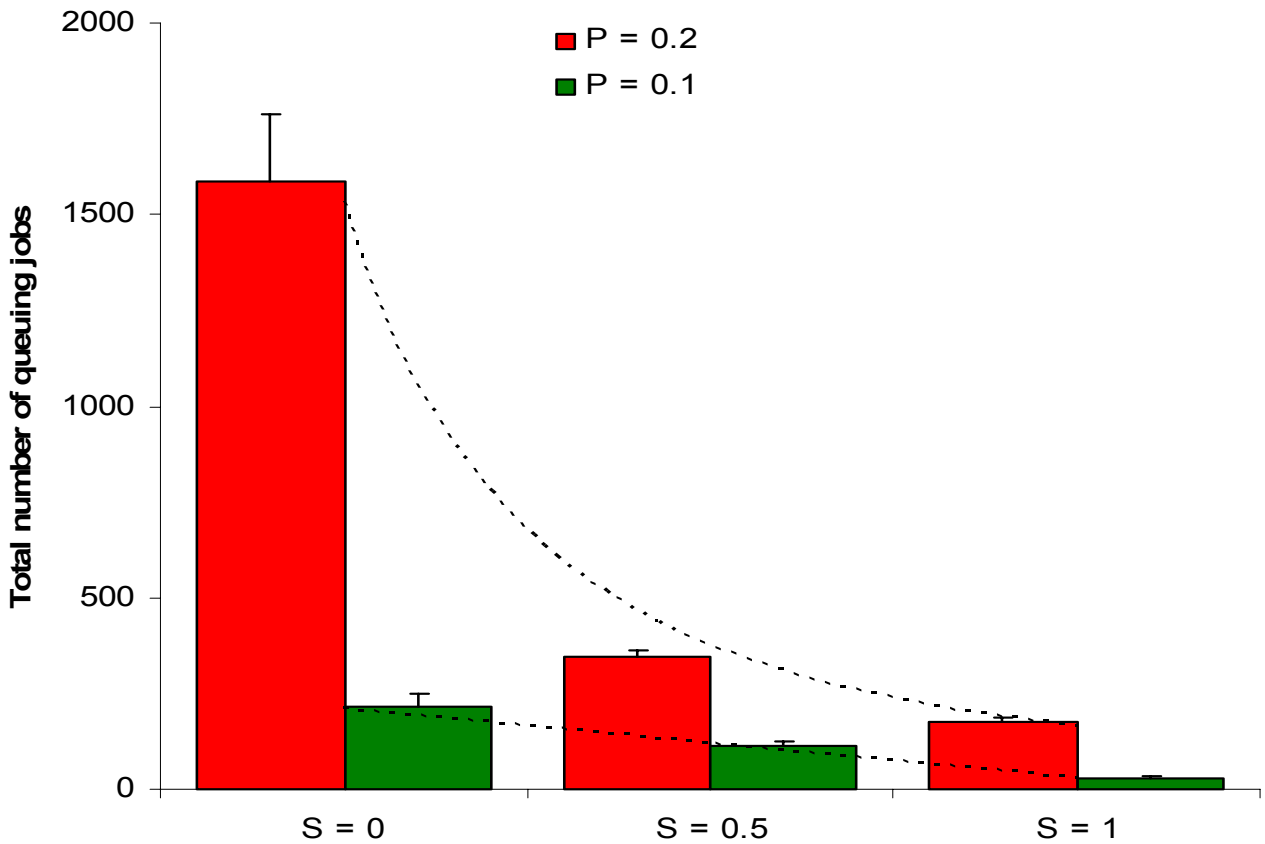


ULB

BT

Saffre & Halloy, 2005

Queue build-up



Saffre & Halloy, 2005

Sanity check

- These were all Monte Carlo simulation results.
- But there are also mathematical tools to help us understand and predict:
  - How such a system will evolve
  - Whether it will reach steady-state
  - How much time will be spent (statistically) in every possible configuration
- Many of these tools tend to quickly become intractable when complexity increases, but they can still play a vital part in validating simulation.

ULB

BT

Saffre & Halloy, 2005

Simplest case: 2 devices, 2 modules

- Rules:
  - If device d_i has module j installed, then it will accept a request for service s_j with probability

    $$P_{ij} = 1/\sigma_i$$

    where σ_i is the number of modules installed on device d_i.
  - If d_i does not have module j installed, it will accept a request for service s_j with an arbitrarily fixed probability

    $$P_{ij} = q = 0.1$$

    In this case d_i will have to change its configuration in order to provide the service.
  - Every time that a device doesn't accept a request, it makes an attempt at delegating it to the other.

ULB

BT

Saffre & Halloy, 2005

Simplest case: 2 devices, 2 modules

- Rules (cont'd):
  - If the request is accepted by the other device, then if the originator has module *j* installed, it can choose to uninstall it and will do so with probability

$$R_{ij} = \sigma_i / m$$

    where *m* is the number of participating devices (here *m* = 2).
- Finally, we make two assumptions:
  - a request bounces back and forth until it is eventually accepted
  - each device receives a similar number of request for each of the *n* = 2 services.

ULB

BT

Saffre & Halloy, 2005

System states

- When $m = n = 2$, although there are $2^{mn} = 16$ possible configurations, when taking symmetry into account, one is left with only seven meaningful states:

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

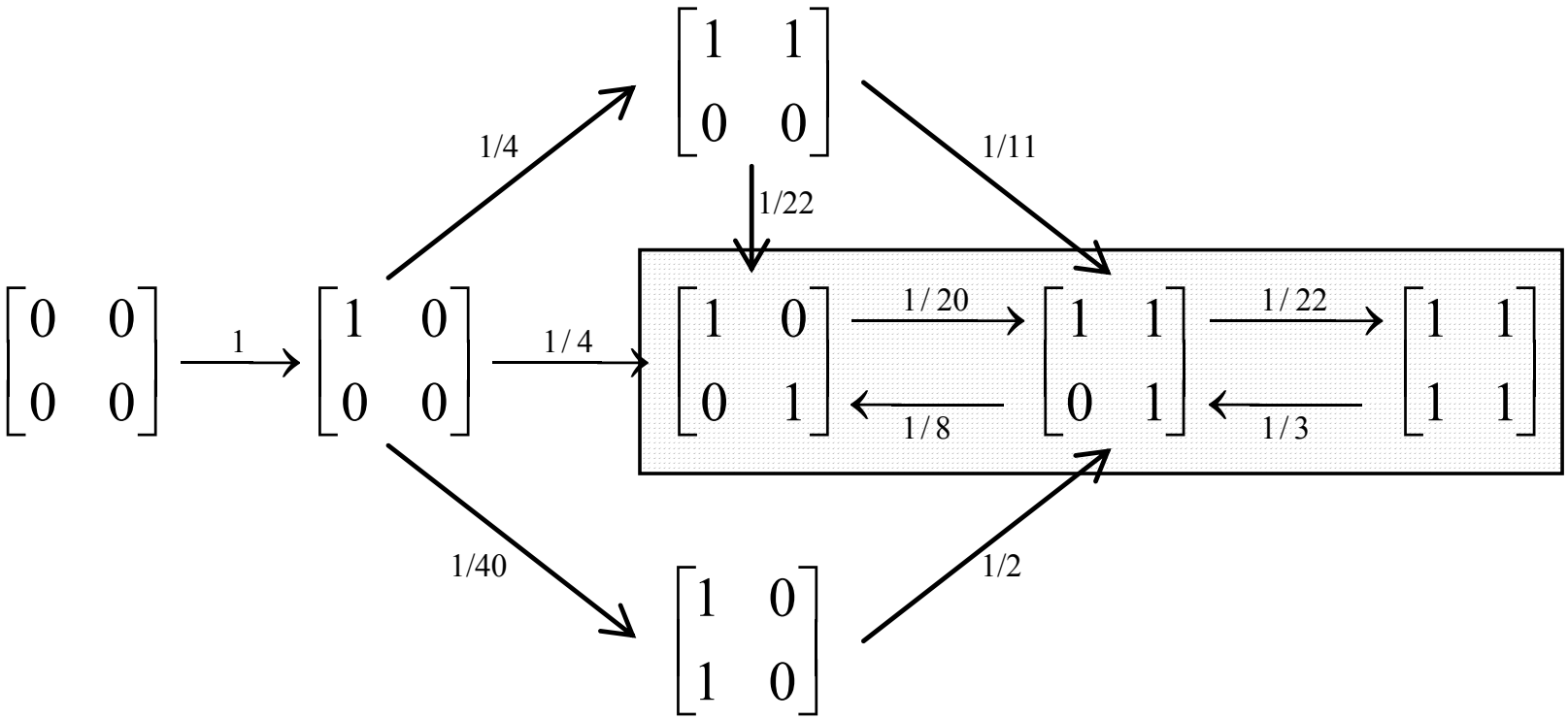
- Given the local decision rules, it is possible to fully predict the rate of transition between them.

ULB

BT

Saffre & Halloy, 2005

Markov chain



Saffre & Halloy, 2005

Markov chain

- Finding the invariant distribution, one can predict that the system should be in

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$ 0.6875 of the time

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$ 0.275 of the time

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$ 0.0375 of the time

- By repeating the same procedure for other parameter values (local rules) we can use the analytical solution to verify that simulation results contain no artefact.

ULB

BT

Saffre & Halloy, 2005

Back to Monte Carlo

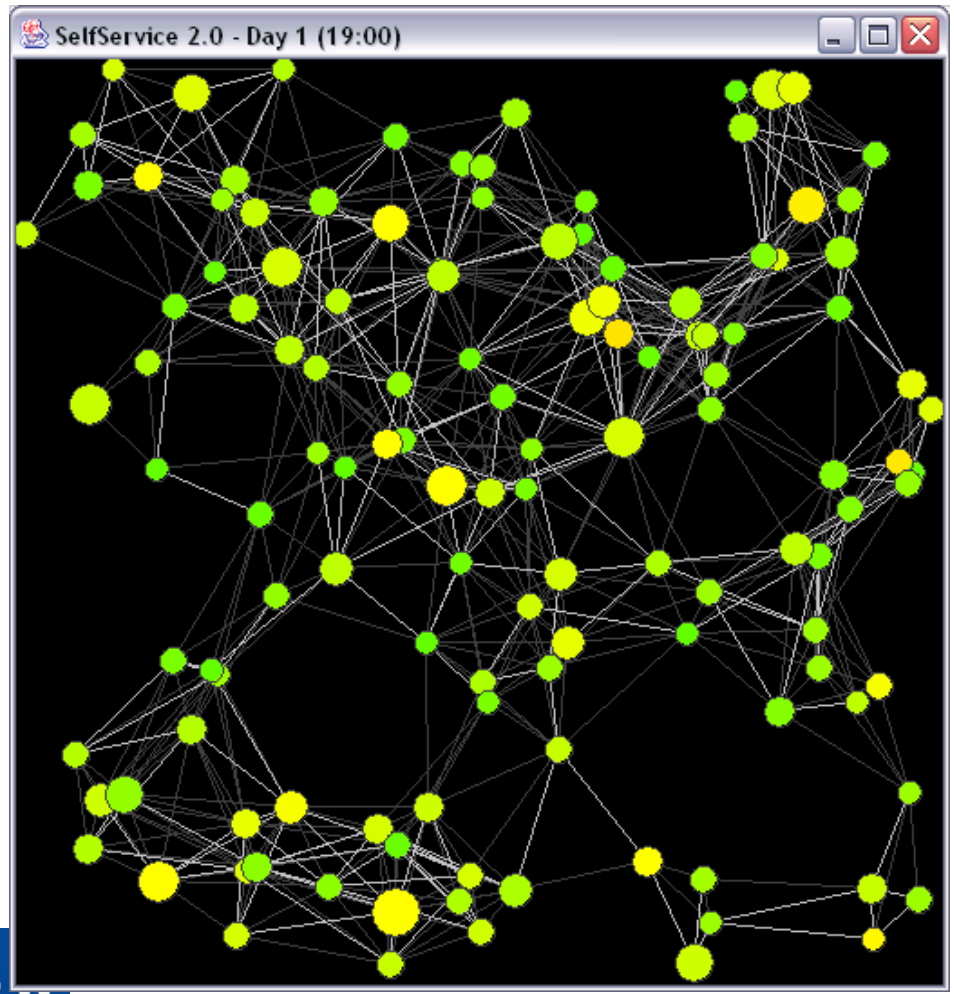
- Validating numerical tools using analytical projections is essential, as it is very easy to produce stable yet *erroneous* (or biased) simulation code!
- Only when the correctness of the simulation engine is beyond reasonable doubt can it be confidently used to produce results independently from the mathematical framework…
- Which (unfortunately?) is made necessary by the fact that the analytical approach doesn't scale.
- For example, if for $m = n = 3$, there are still "only" 36 meaningful states, for $m = n = 4$, there are already 317, and even identifying them among the $2^{mn} = 65536$ possible configurations becomes far from trivial!

ULB

BT

Saffre & Halloy, 2005

Pervasive “SelfService”



- Mobile nodes (pda’s?).
- Regular, but unpredictable, daily activity cycles.
- Colour code = QoS.
- Size = number of modules installed.
- Grey links = in range.
- White links = identified opportunities for co-operation.

ULB

BT

Saffre & Halloy, 2005

It's not all about software modules

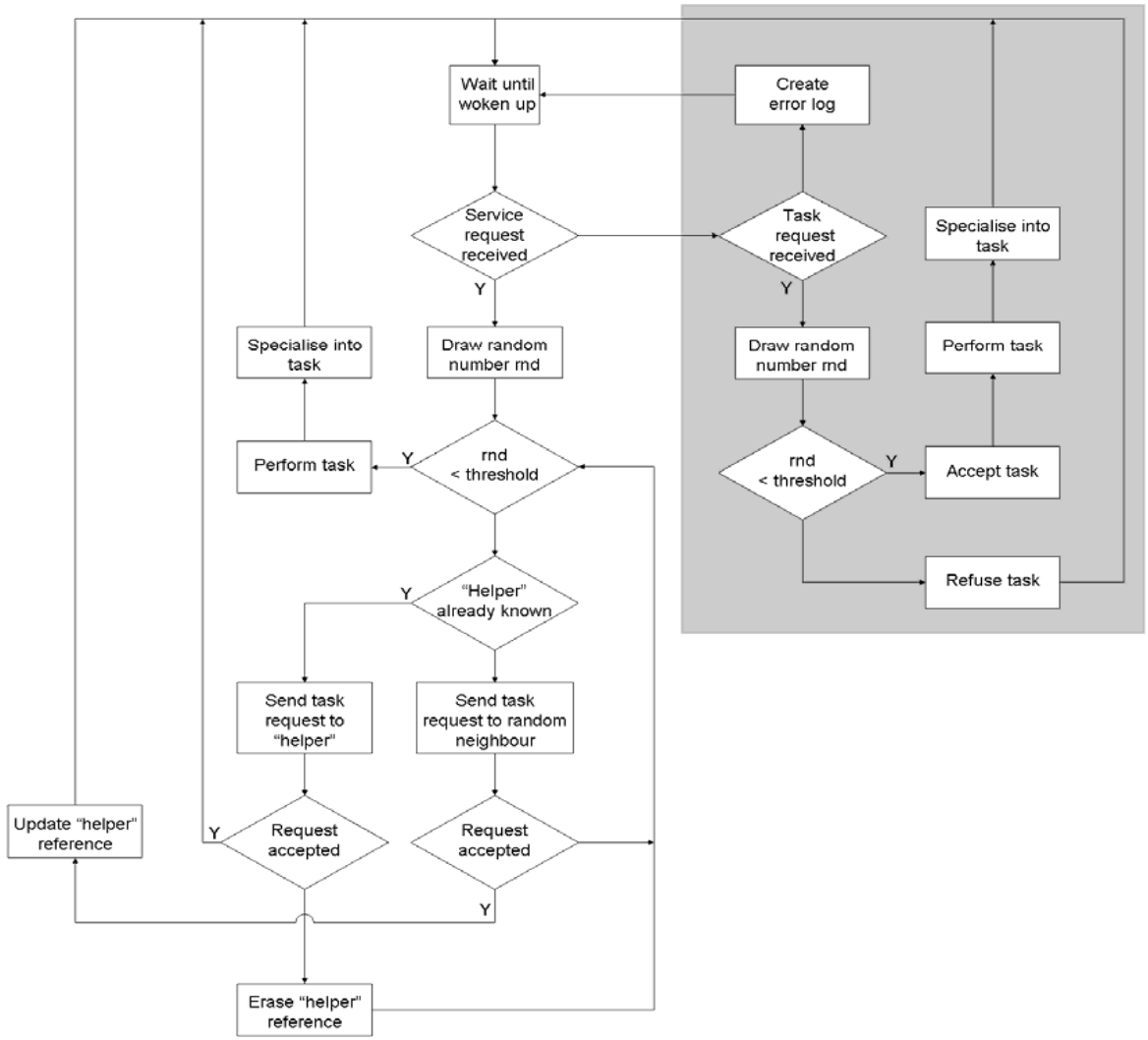
- There is also the opportunity to fine-tune individual machines' "internal state" (i.e. differential allocation of resources like storage, processing power etc.)
- Internal state can affect task-specific performance (with a possibility of conflicts when the same machine is expected to perform a variety of tasks).
- The same principles of self-organised differentiation can be applied to maximise the *collective* efficiency of a community of devices.

ULB

BT

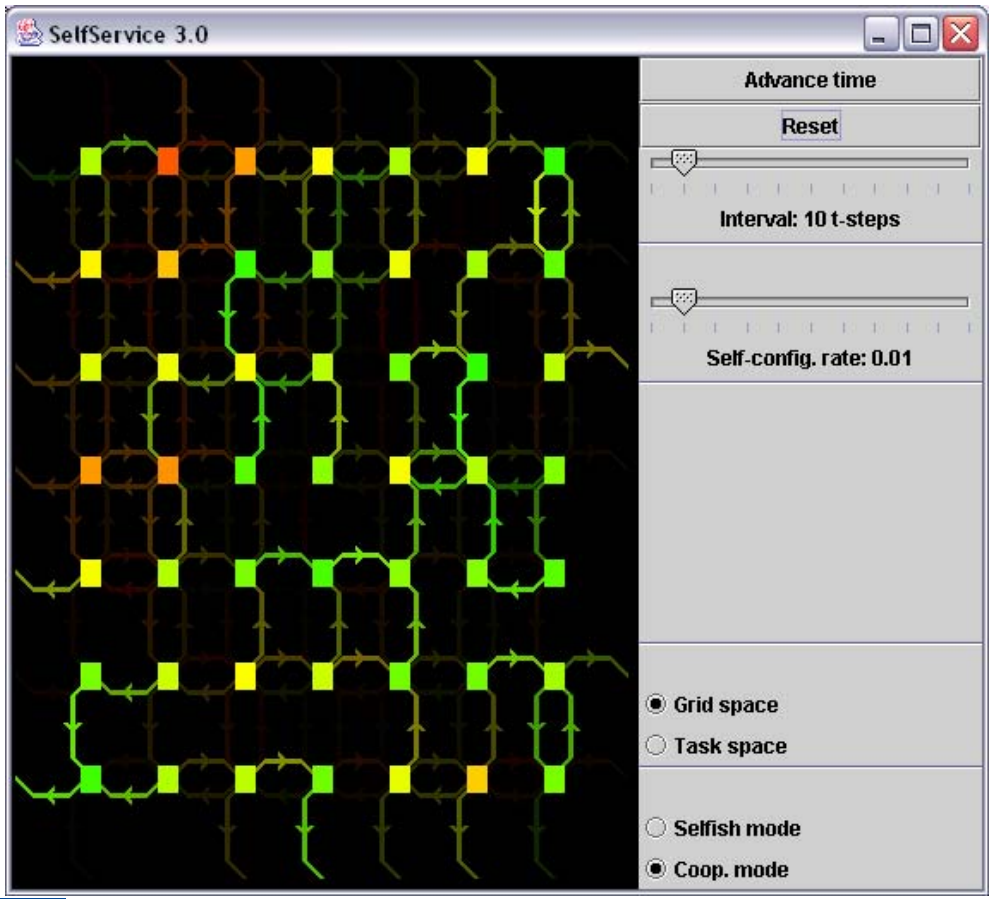
Saffre & Halloy, 2005

Collective differentiation



Saffre & Halloy, 2005

“SelfService”: an artificial ecosystem(?)



- Efficiency proportional to level of specialisation.
- Job acceptance prob. proportional to efficiency.
- Co-operation limited to first neighbours.
- Colour code:
  - QoS (nodes)
  - Delegation success (links)
- Brightness:
  - Frequency of delegation

ULB

BT

Saffre & Halloy, 2005

Embryo

- Let us go back to the future and the full-blown "*Adaptive Service Ecosystem*", where network topology *and* service characteristics are dynamic.
- Key features/properties:
  - All components have some "intelligent" autonomic capabilities
  - The relationships between them (esp. who is connected to whom in the overlay) are in permanent turmoil
  - Multiple (hopefully modular!) applications coexist within the same service ecosystem
  - Which modules make up which application is constantly revised: new applications appear, others die out, new (old) modules are added to (removed from) existing applications
  - Demand is highly variable and requires "on-the-fly" reallocation of resources
  - There is no common information repository and no control centre can cope with the rate of change in the system
- Can we make it work and how?

ULB

BT

Saffre & Halloy, 2005

Embryo's ancestor: T-Man

- Gossip-based, self-organizing P2P network protocol developed by Jelasity and Babaoglu.
- Nodes "find" and "migrate" to their target location in the system by gradually refining their global knowledge through local messaging, and by rewiring accordingly.

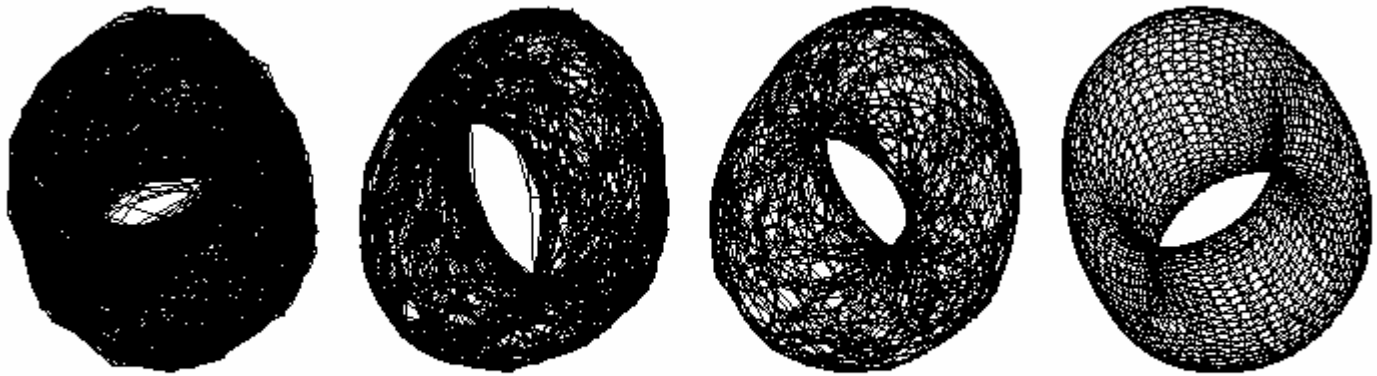


Figure 2. Illustrative example of constructing a torus over $50 \times 50 = 2500$ nodes, starting from a uniform random topology with $c = 20$. For clarity, only the nearest 4 neighbors (out of 20) of each node are displayed.

http://www.cs.unibo.it/pub/TR/UBLCS/2004/2004-07.pdf

ULB

BT

Saffre & Halloy, 2005

So what else do we need?

- Basically, in T-Man, the target location of a node is defined by some internal property (which could be a constant or a variable) that doesn't change as a function of the node's neighbourhood.
- But what if a node's *objective* transcends its internal state, and it can *adjust* its own characteristics in its (selfish) pursuit of that objective, so as to *complement* what its neighbours have to offer?
- Then a node's target location (equivalent to its objective) varies over time, and since it's a function of its neighbours' and its own internal properties, it can potentially be reached by rewiring, by changing state or by a combination of both.
- This is the complex situation that Embryo is trying to deal with…

ULB

BT

Saffre & Halloy, 2005

Why “Embryo”?

- Because it’s not unlike what’s happening during the morphogenesis of multicellular organisms.
- It all starts with a handful of identical stem cells, which progressively multiply, specialise and “migrate” (i.e. change relative positions during development).
- Eventually, the process gives rise to a living being, whose existence depends on highly specialised and interdependent organs, each made of many cells belonging to various differentiated types.
- And of course this extraordinarily complex orchestration all happens in the absence of any conductor, through a combination of local mechanisms ranging from cell adhesion to cross-stimulation/inhibition.

ULB

BT

Saffre & Halloy, 2005

What does it mean (in ICT terms)?

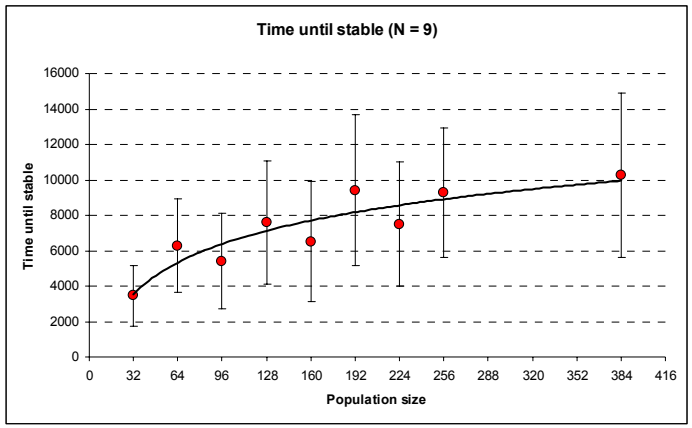
- An organism = a “service ecosystem”
- An organ = an application (involves multiple services)
- A cell type = a service (corresponds to a software module)
- A cell = a node (can be a physical device or a virtual entity) providing a point of access to an application
- A group of neighbouring cells = an overlay network connecting service instances
- Cell differentiation = instantiation of a particular service
- Cell migration = rewiring within the overlay
- Cell signalling = gossiping between neighbouring nodes

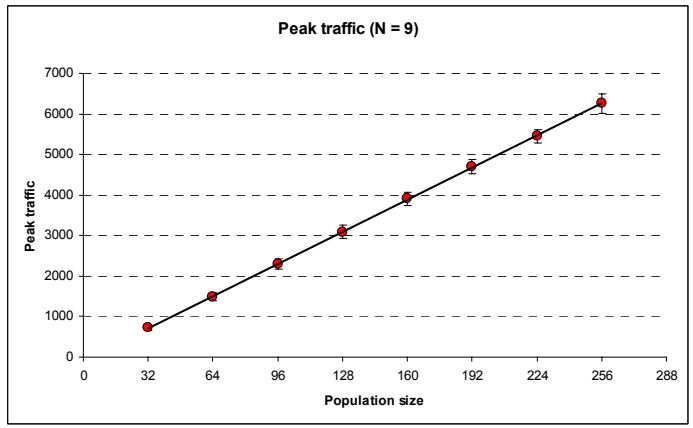
ULB

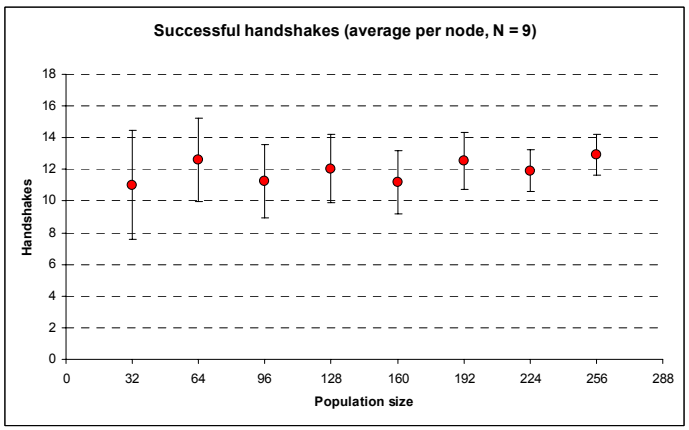
BT

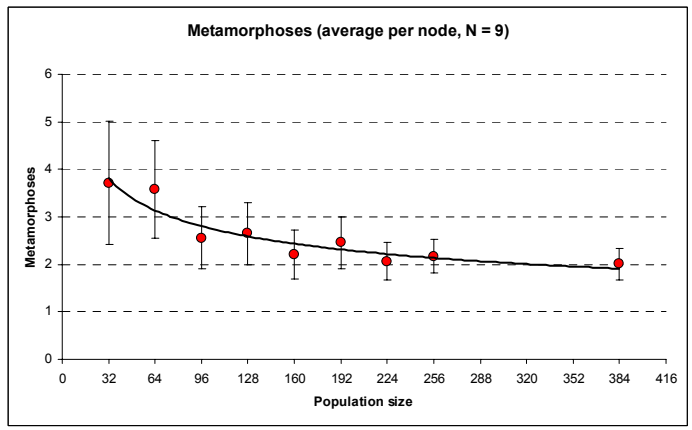
Saffre & Halloy, 2005

Embryo: some simulation results



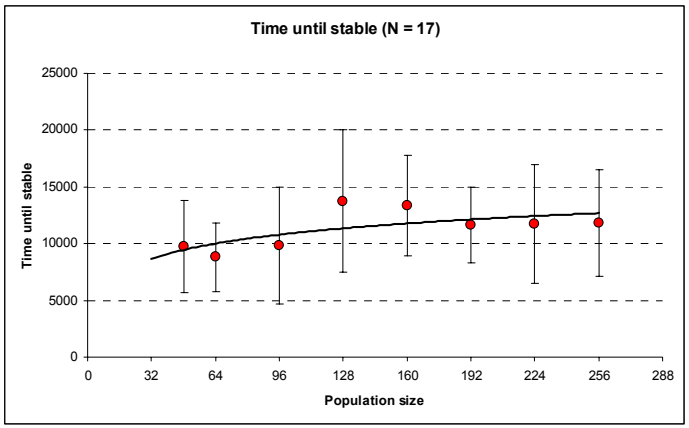


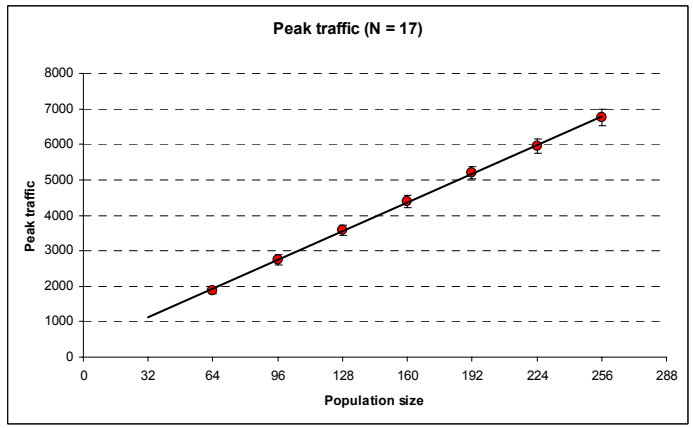


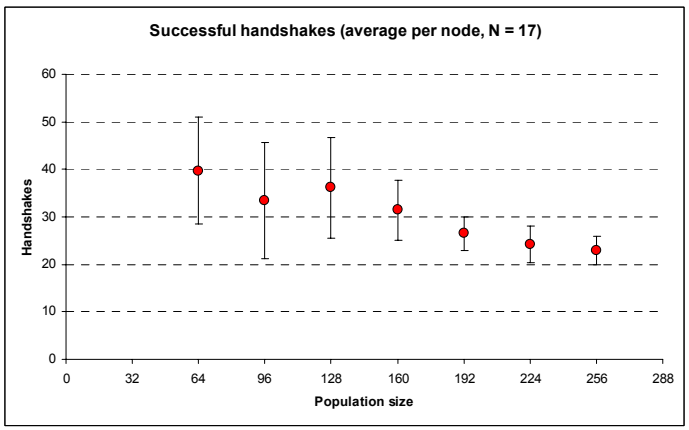


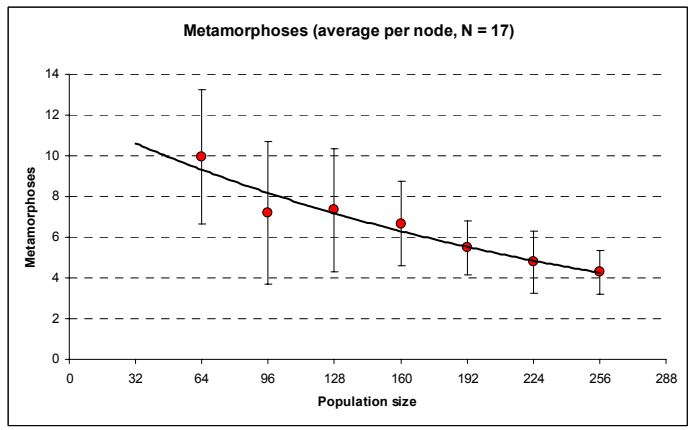
Saffre & Halloy, 2005

Embryo: some simulation results (2)







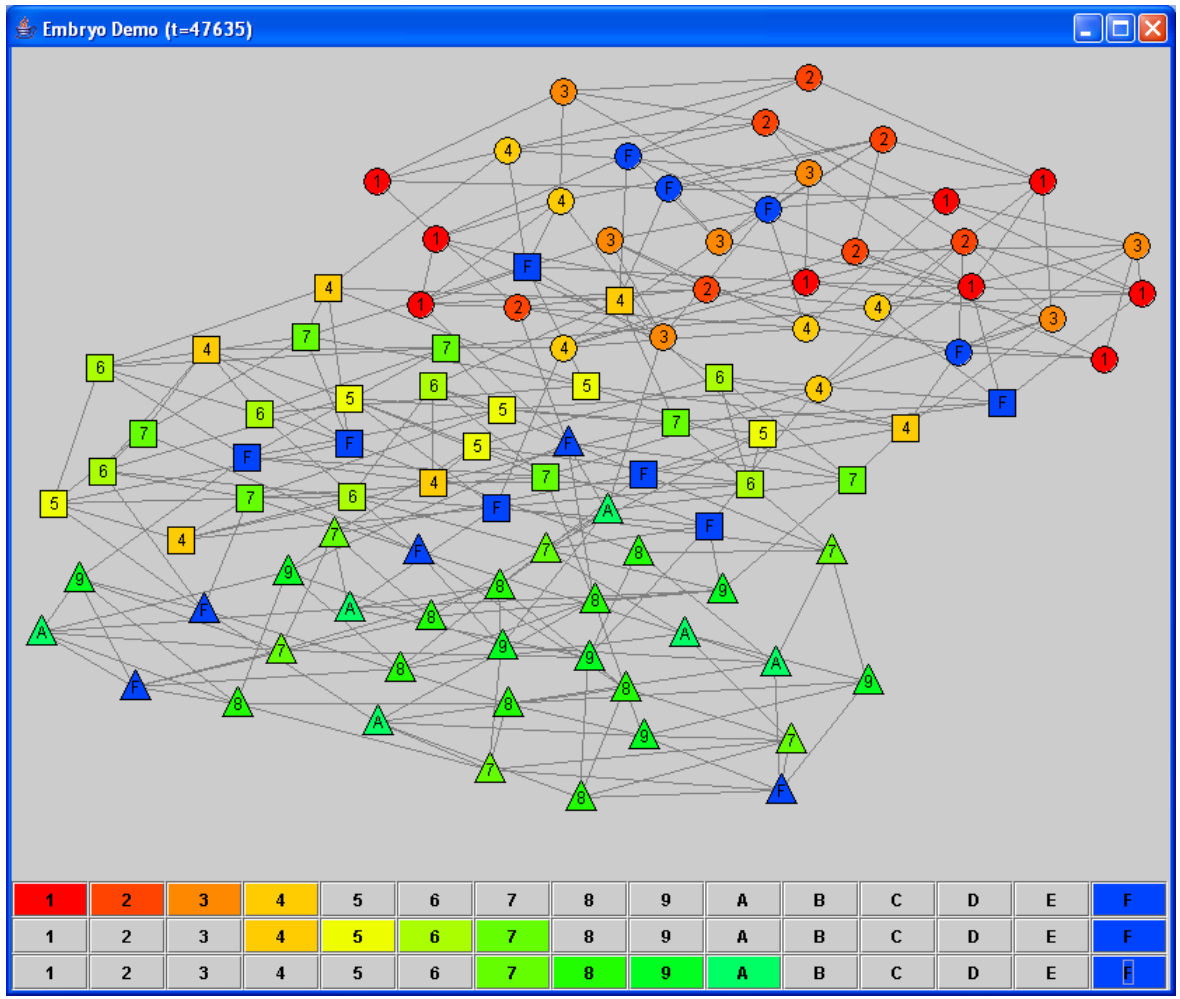


ULB

BT

Saffre & Halloy, 2005

Embryo demo



Saffre & Halloy, 2005

Conclusions

- The borders between:
  - Autonomic computing/communication
  - Networking and services
  - Pervasive computing
  - Resource sharing
  - Software design and engineering

  are fading rapidly…
- Because they all share the same problem: less control, increased complexity, poor understanding of artificial systems emergent properties.

ULB

BT

Conclusions (2)

- The corresponding industries have increasingly overlapping markets with, e.g., Telcos and IT companies now competing to provide ICT services.
- The race is on, and whoever can demonstrate that they've "cracked the complexity nut" *in practice* will have a decisive advantage.
- Because they will be able to offer *new* ICT solutions that are *predictably* and *reliably* efficient in the unpredictable and unreliable world of the "network economy".

Publicity time…

- 1st International Conference on Autonomic Computing (ICAC): New York, USA - May 2004
- 2nd ICAC: Seattle, USA - June 2005
- *3rd ICAC: Dublin, EU - June 2006… be there!*

www.autonomic-conference.org

- Autonomic Communication Forum (ACF)
- Federates the majority of European academics and industrialists active in the field of autonomic ICT

www.autonomic-communication.org

- Should you wish to keep track of our work over the next 3 years, the CASCADAS project website is a good place to start

http://netmob.unitn.it/cascadas/

ULB

BT