Autonomic Communication

Mikhail Smirnov
Fraunhofer FOKUS, Berlin

Fraunhofer Institute for Open Communication Systems

PGNet 2005, Liverpool John Moores University
27 June 2005

Roadmap

- What is autonomic communication
- Etiquette in autonomic communication

The AC Vision

- Paradigm shift:
  - Towards a self-organizing and context-aware autonomous network infrastructure
- Disclaimer:
  - This talk is about looong-term research
- Reminder:
  - Radical technology change happens within one university cycle

What Industry says

"The obstacle is complexity ... Dealing with it is the single most important challenge facing the I/T industry“

Paul Horn, IBM

“Technology needs to manage itself”

Irving Wladawsky-Berger, IBM

“New EcoSystem is evolving, new ways of interaction, in which network orchestrates"

Joelle Gauthier, Alcatel

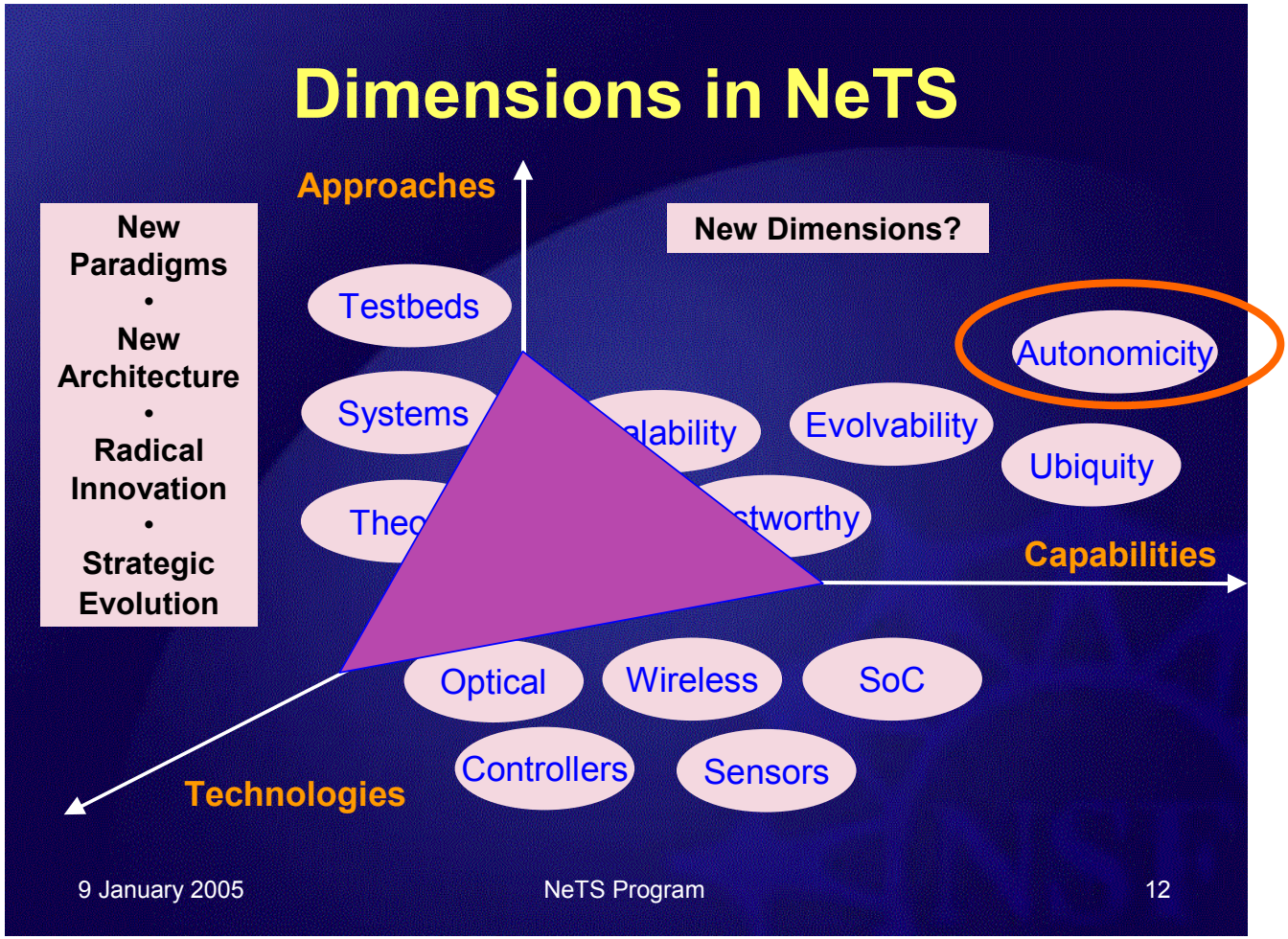
“Service creation at service element”

Graca Carvalho, Cisco Systems

Self – Star

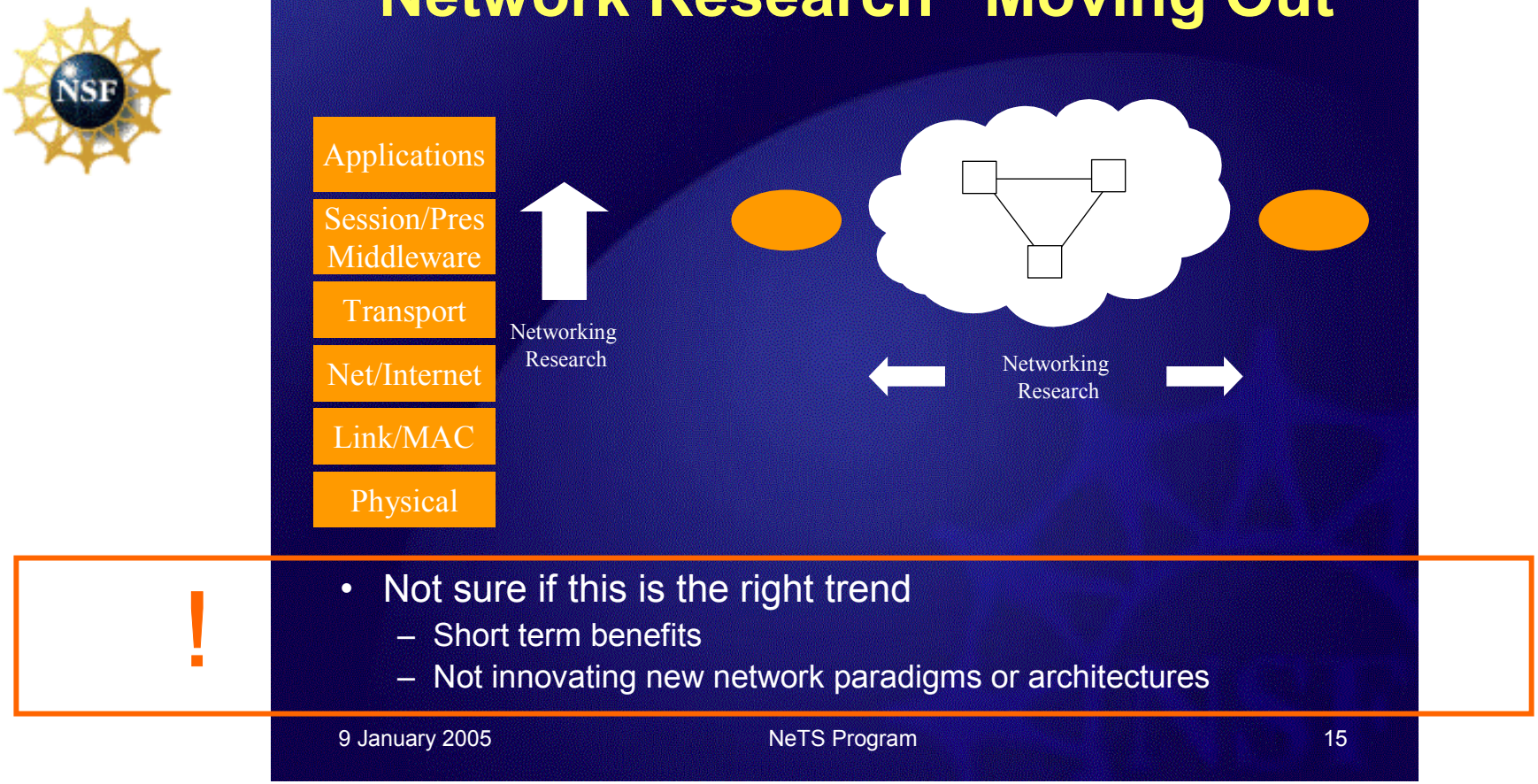
After http://www.research.ibm.com/autonomic/overview/elements.html

What NSF says



Source: Guru Parulkar (NSF), The NeTS Program - **Broadly Defined Networking Research**

Guru‘s Observation



Source: Guru Parulkar (NSF), The NeTS Program

The True Best Effort

Traffic theory (demand ↔ capacity ↔ performance)	flow-aware networking	community of features serving a flow Better intrusion detection
Single QoS metric at feature level for all app. classes	dynamic degradation distribution	community of features serving all flows of the same class Better resource management
Stateless per flow guarantees (short-term traffic management decision making)	measurement-based CAC	community of features empowered by IPFIX/PSAMP Better scalability
Contracted access	mediation	community of features involved in SLA provisioning and monitoring Better fairness

QoS = performance+availability+security

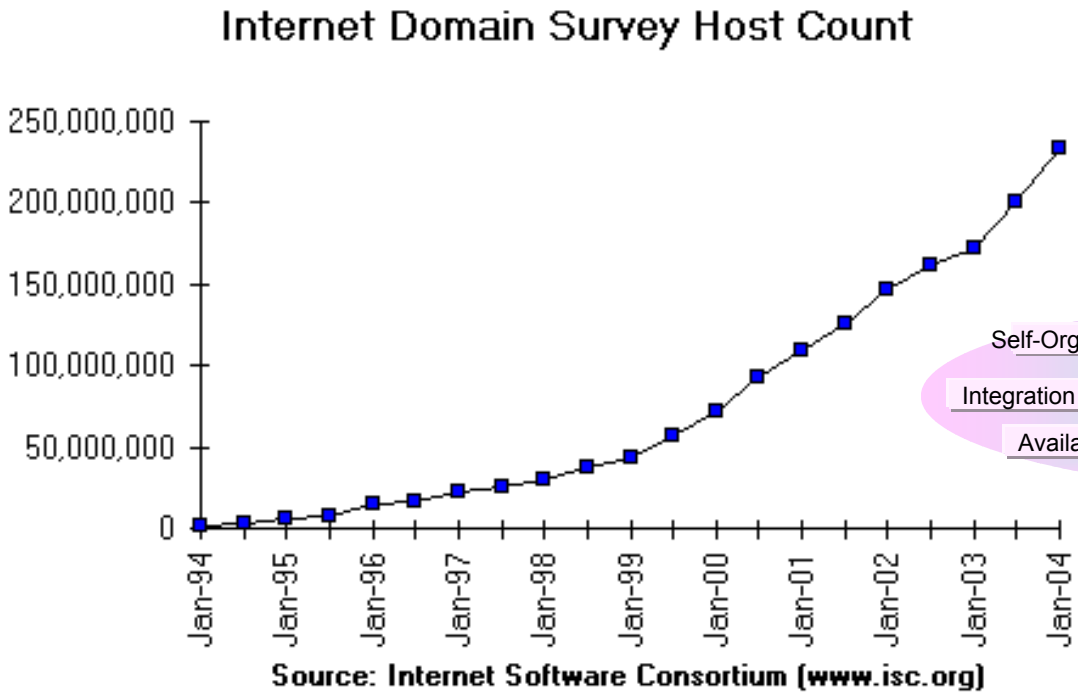
- New concepts outside or besides layering:
  - Explicit handling of networking context
  - Adaptive control:
    - Adaptation of many (independently designed and implemented) controls (resource, access, addressing, etc.)
    - Orchestrated control under the governance of locally computed fitness
- X-layer/Non-layer approaches
  - give away “layer secrets”
  - open [layer] functions for inspection of function’s **workflow**
    - a fundamental foundation for **trust establishment** and anomaly detection
    - Basis for dynamic reconfiguration (for high availability) and **service creation per service element**

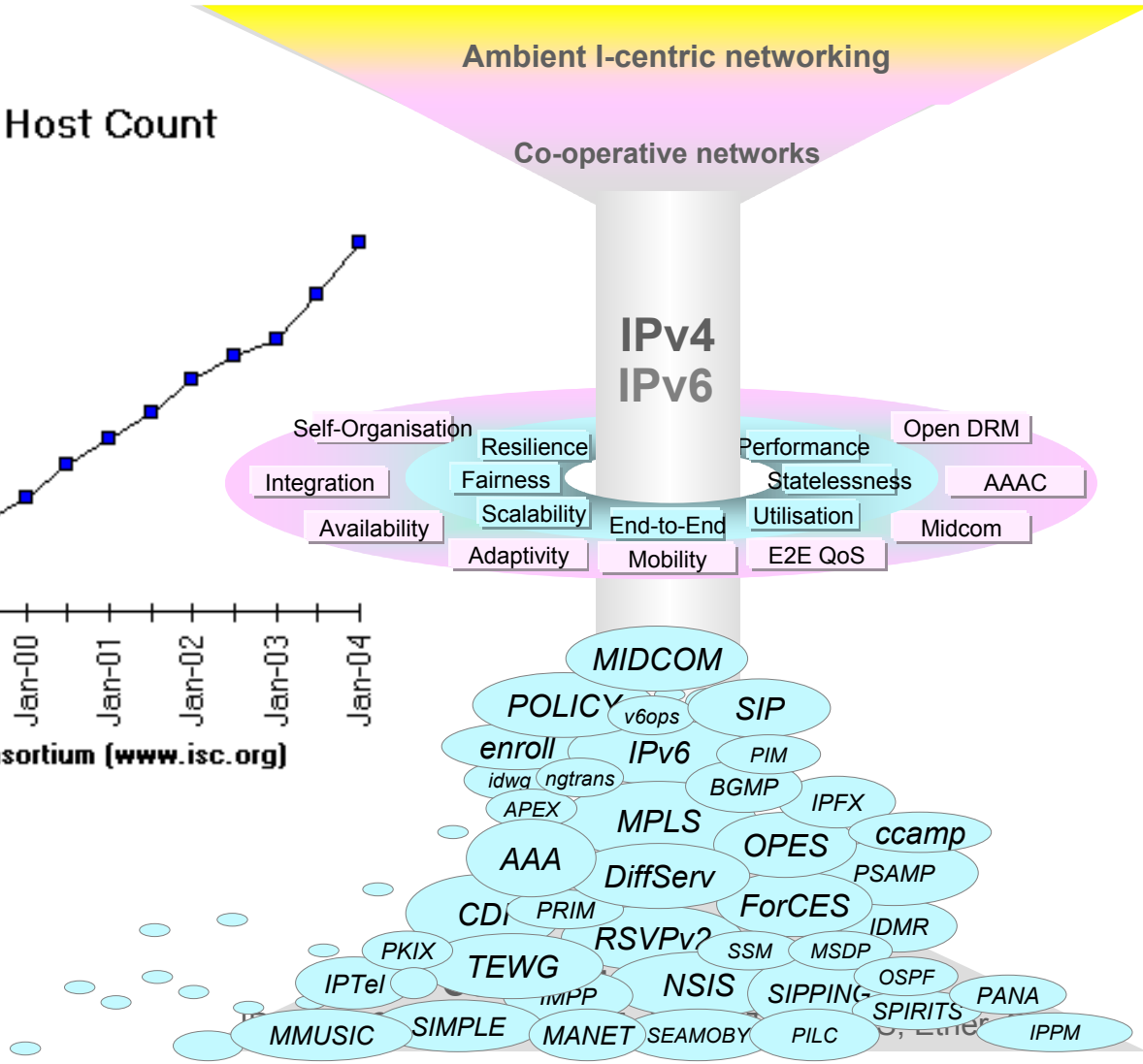
Managing Internet Complexity

Complexity Measures

- Many types
  - Computational, System, Evolutionary, ...
- Many measures:
  - Maintenance cost [P. Baran, 1964]
    - Total cost of managing the diverse set of mechanisms
  - Kolmogorov complexity [1965]
    - ~ Min. size lossless compression
  - Composite metric [J. Casti, 1979]
    - Structural C. (Hierarchy, # of Types, Connectivity, Layering)
    - Dynamic C. (Behaviour, Time scales)
  - Fault diagnostics [J.-F. Huard, 1993]
    - Min # of binary queries to find the cause of a fault

Internet Growth & Complexity



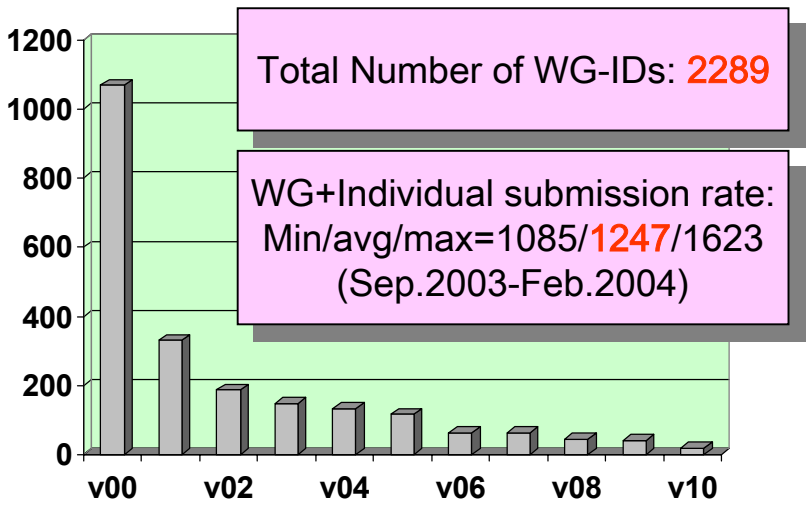


Internet Complexity Measures

- Complexity ~ management and maintenance effort
- How many protocols an ISP needs to support? ~1400, but +
  - Standards – 82
  - Draft Standards - 69
  - Proposed Standards - 797 (SIP, RTP, MPLS, RTSP, ...)
  - Experimental – 175
  - Historic - 86 (html v.2, snmpv2, X.500. BGP-OSP interaction,....)
  - BCP - 82 (XML Registry, SIP Basic call flows, ...)
  - RFC Editor Queue: in total 211; 96 already approved

http://www.rfc-editor.org/rfcxx00.html

+ Internet Drafts + ...

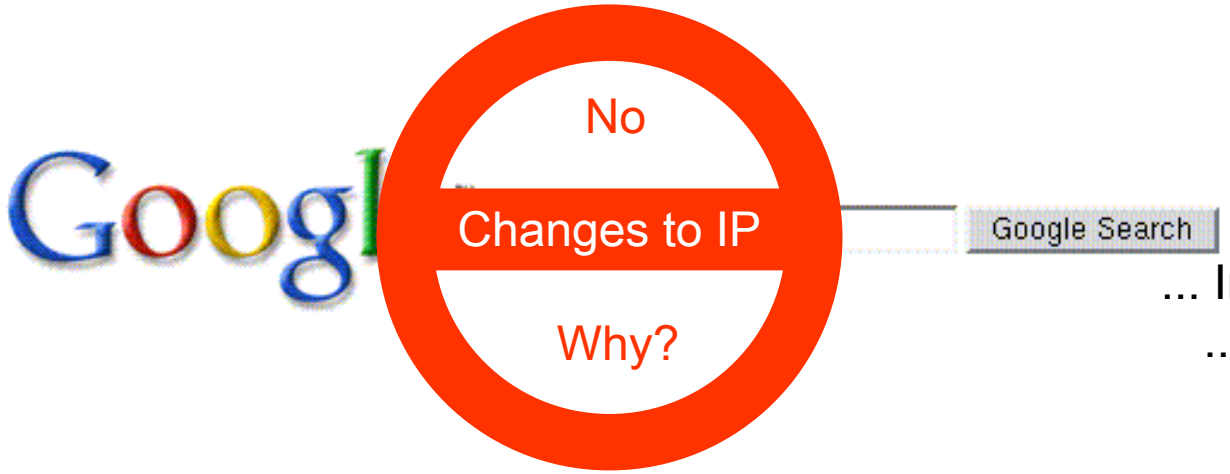


1id-index.txt

+ IANA registered TCP/UDP port numbers: 8394 + ...
+ Number of protocol parameters: 183 normative documents + ...
+ MIBs + ...
+ vendor specific solutions + ...
+ interop + ...
+ CRM (SLA, etc.) +
+ OSS + ...

With all this Complexity ...

Without ...
No ...



... Policy
... Address
... Distribution
... Processing
... Implementation
... Configuration
... Routing
... Security

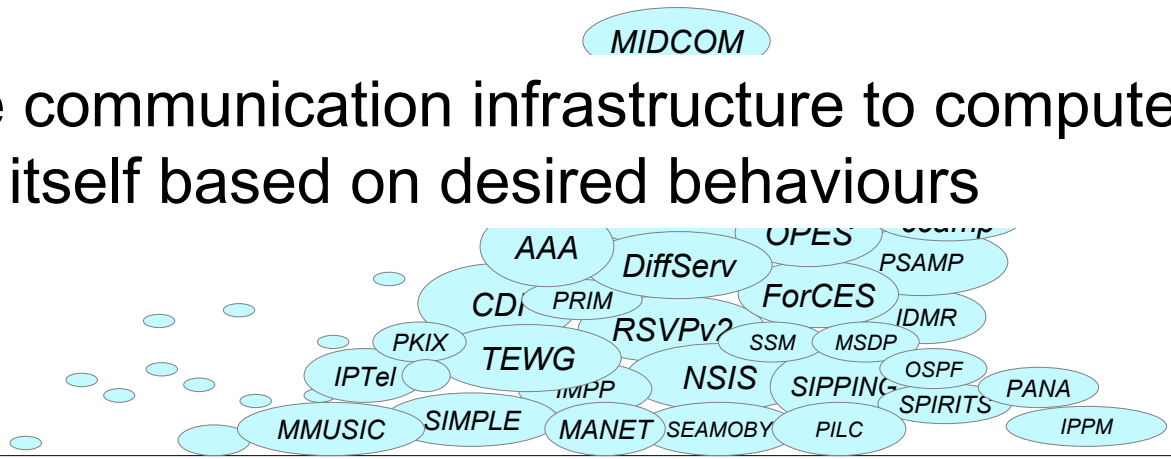
- IP is a least common *functionality* for data transfer between heterogeneous link layer technologies
  - IP datagram is self-contained (header)
  - Supports exception handling (ICMP)
  - Capitalises on ubiquitous infrastructure (IP module)
- Can we do the same for network control & management plane? YES!

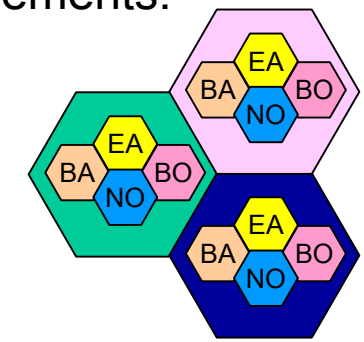
Responding to a Challenge ...

Autonomic Communication fills the gap

With network selfware based on universal and fine-grained multiplexing of numerous policies, rules and events that is done autonomously but facilitates desired behaviour of groups of network elements.

Let the communication infrastructure to compute itself based on desired behaviours





Self-similarity of control plane

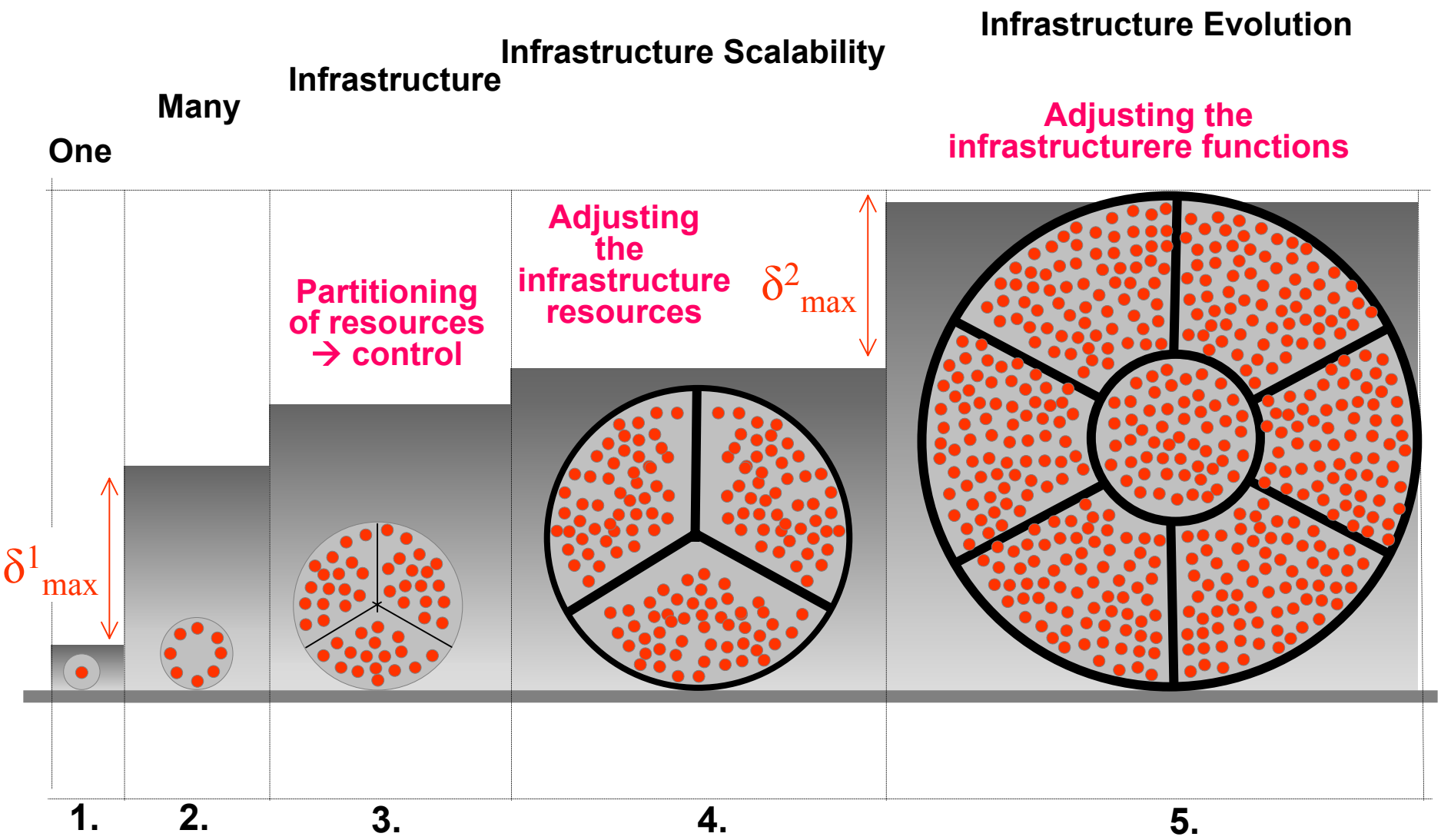
Autonomic Communication in the Internet

Imagine

- Service system comprised of autonomous components
- Self-organisation triggered and driven by service requests
- Several components perform parts of requested service in collaboration
- No assumption that
  - all involved components interpret the service request equally
  - service system is static in terms of the number of its components, their services, capacity, etc.
- Each autonomous component has a Self-Organisation Function (SOF) reflecting the globally shared purpose
- SOF can be invoked potentially by any other component
- The invoked SOF may eventually refuse the service, however only under the condition that service provisioning will disrupt the provisioning of on-going services.

- The described system fits IP routing
- Service model = best-effort
- It works fine
- Power law consequence: other concerns at the same layer?

Evolvability Hypothesis



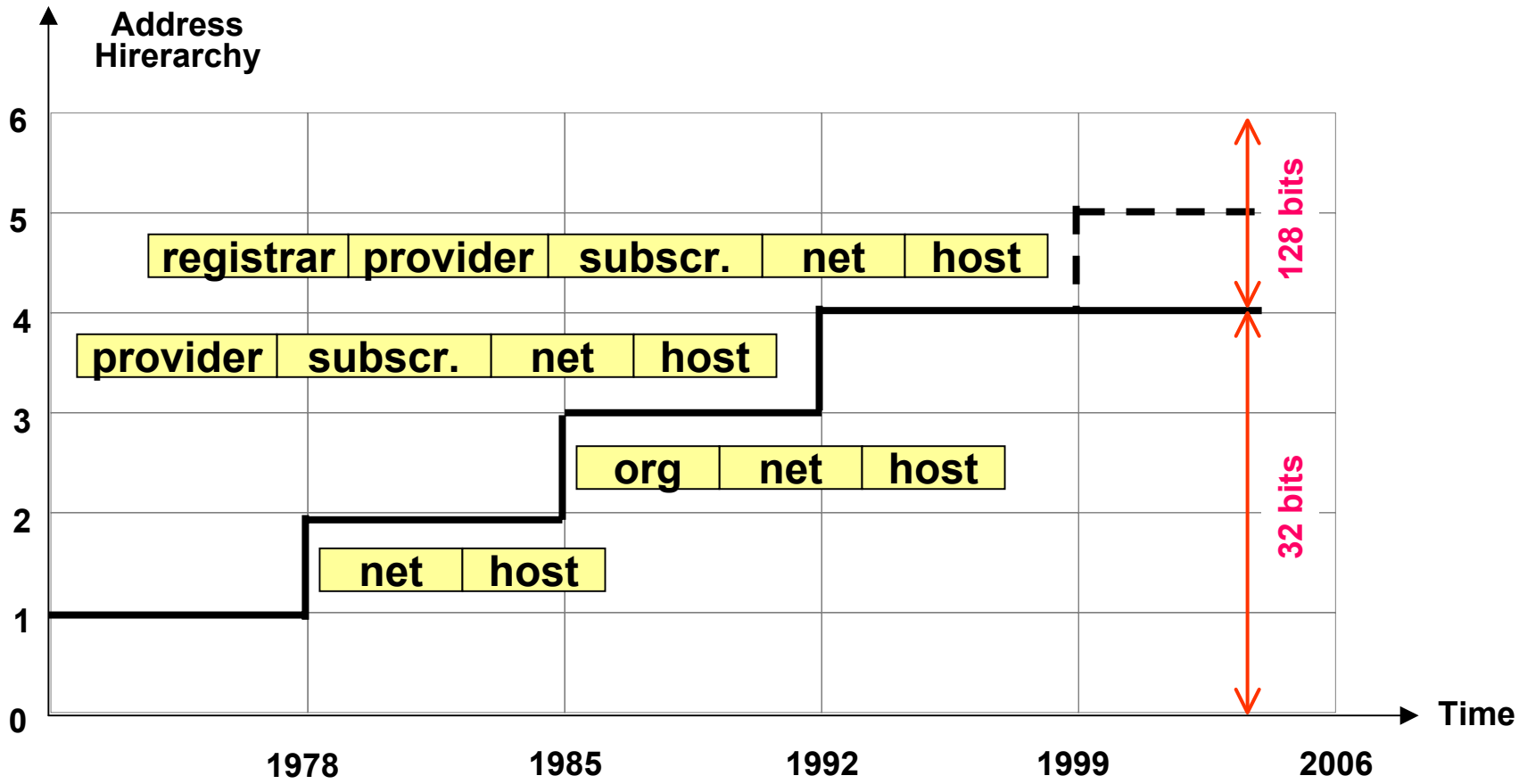
Nolan‘s Stages of Growth

1. Initiation
   - low-level operational systems in functional areas
2. Contagion
   - organization encourages innovation & extensive application of technology
3. Control
   - Manage data resources through control & planning

   Transition Point
4. Integration
   - database & interactive terminals, users see the value & will pay anything (explosive growth)
5. Data Administration
   - data administration introduced
6. Maturity
   - application portfolio completed

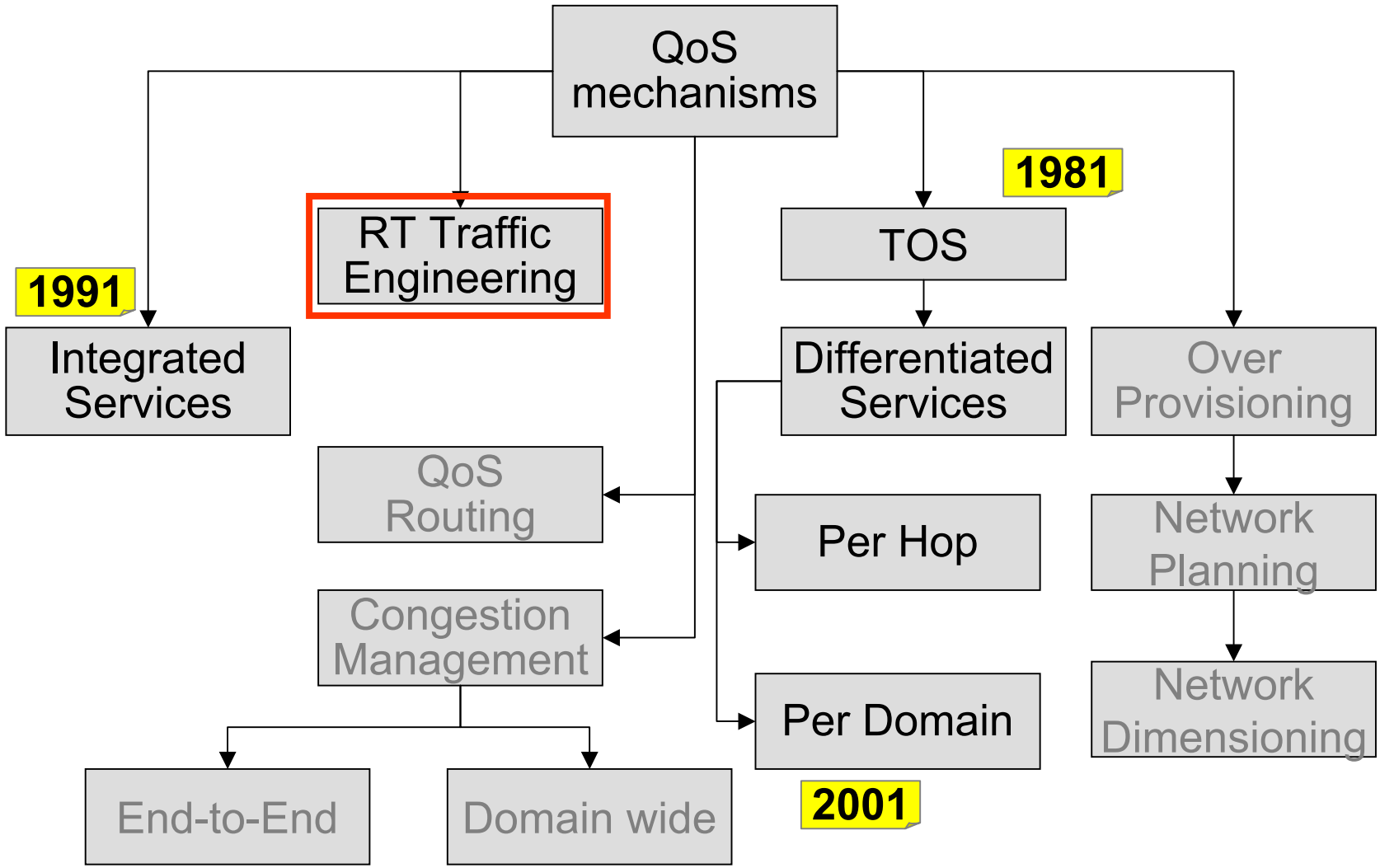
Credit: **Nolan**, R.L. (1979). Managing the crises in data processing. *Harvard Business Review, March-April, 115-126.*

E.g.: Addressing Evolution



- Hierarchy squeezes the address space
- Must keep adding levels of hierarchy to keep routing working
- Each additional layer reduces total addressing capacity

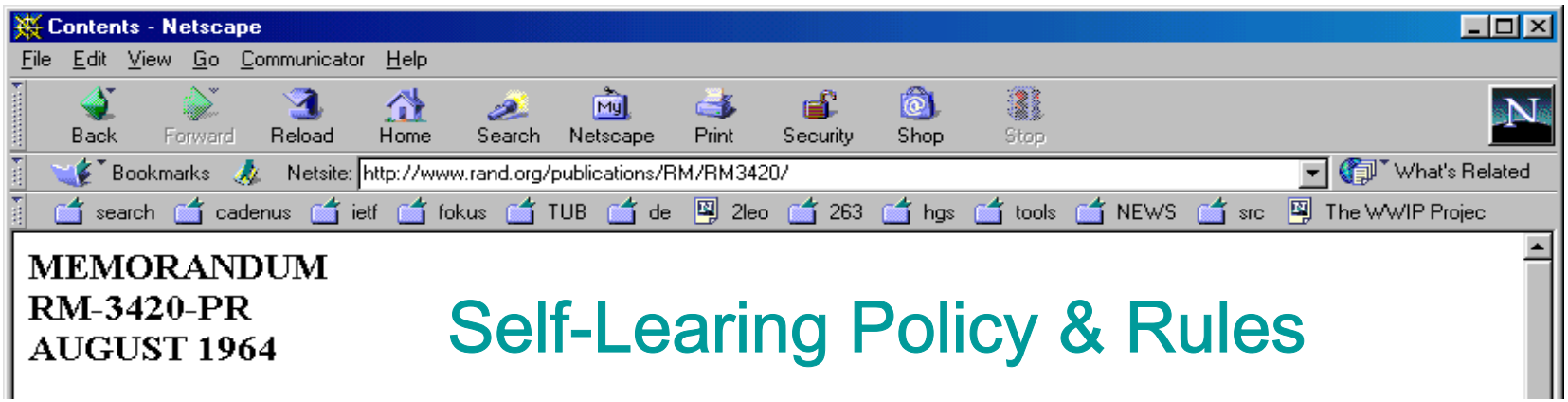
E.g.: QoS Mechanisms Evolution



ECHO

- Emergent Control Hierarchy Organisation – ECHO
- Emergence of layers:
  - evolution at many layers
  - traditional middleware will create a layer on top of a protocol stack
  - network infrastructure evolution will create layers in-between existing stack layers
    - a new abstraction layer
    - needs to create two interfaces
- Host Extension principle:
  - a host shall not depend on any type of infrastructure but can rely on it when eventually available:
    - Virtual servers (DNS, DHCP, CIB, …) at each host are hooks of potential control trees rooted at host

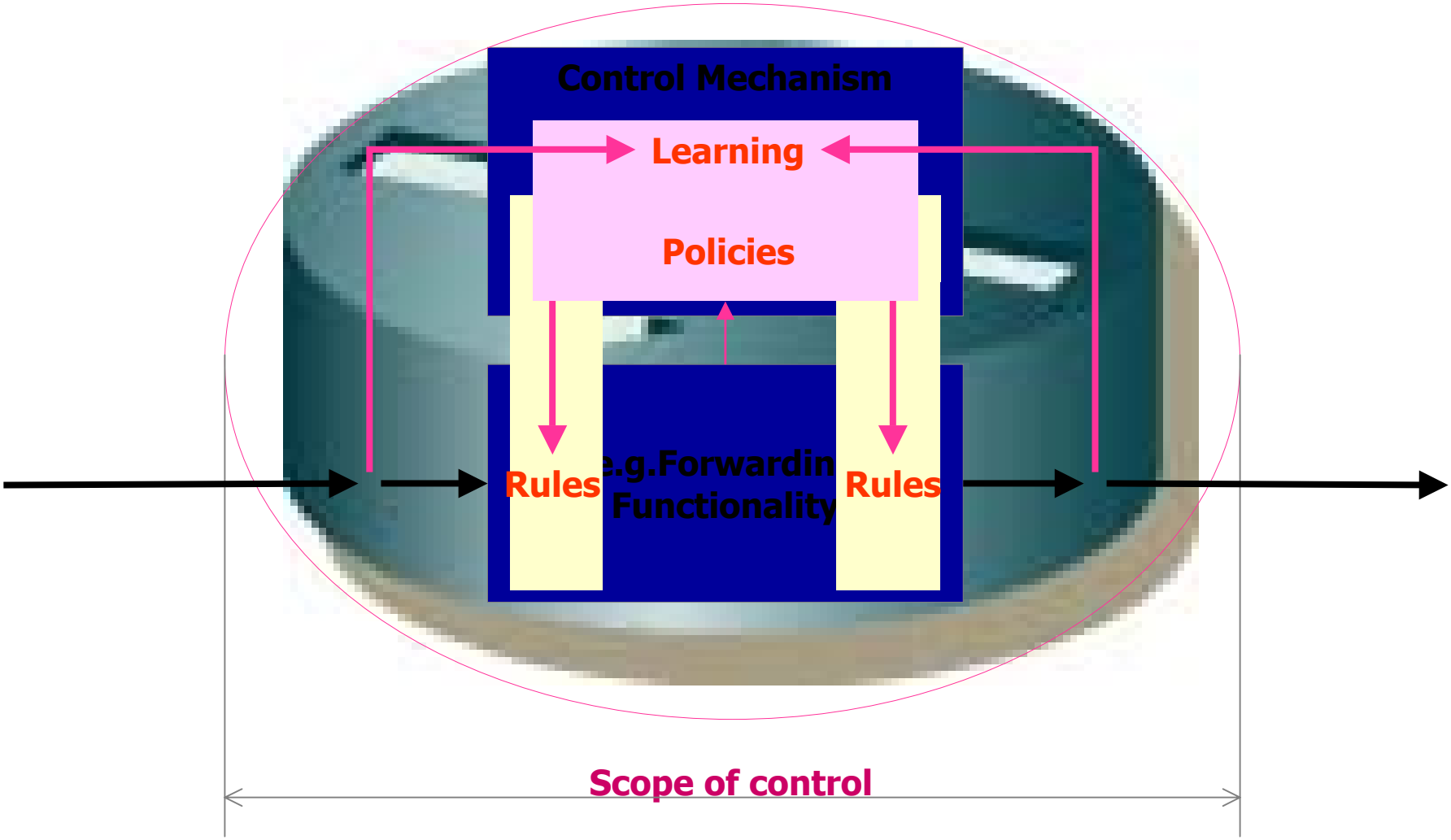
40 years ago ...



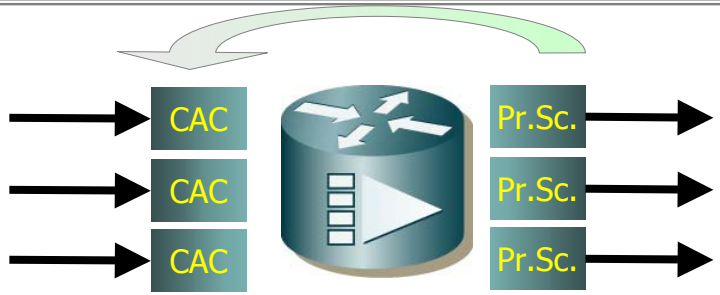
- „What is envisioned is a network of unmanned digital switches implementing a self-learning policy at each node so that overall traffic is effectively routed in a changing environment--without need for a central and possibly vulnerable control point“
- „The network can be made rapidly responsive to the effects of destruction, repair, and transmission fades by a slight modification of the rules for computing the values on the handover number table“

Source: Paul Baran, ODC, 1964, v.1. RM-3420-PR, ch4

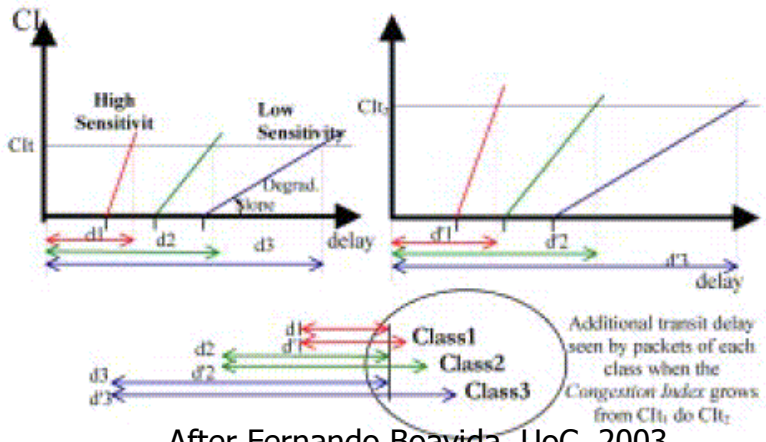
Self-Learing Policy & Rules



Autonomic Control Example



After Jim Roberts, France Telecom R&D, 2003

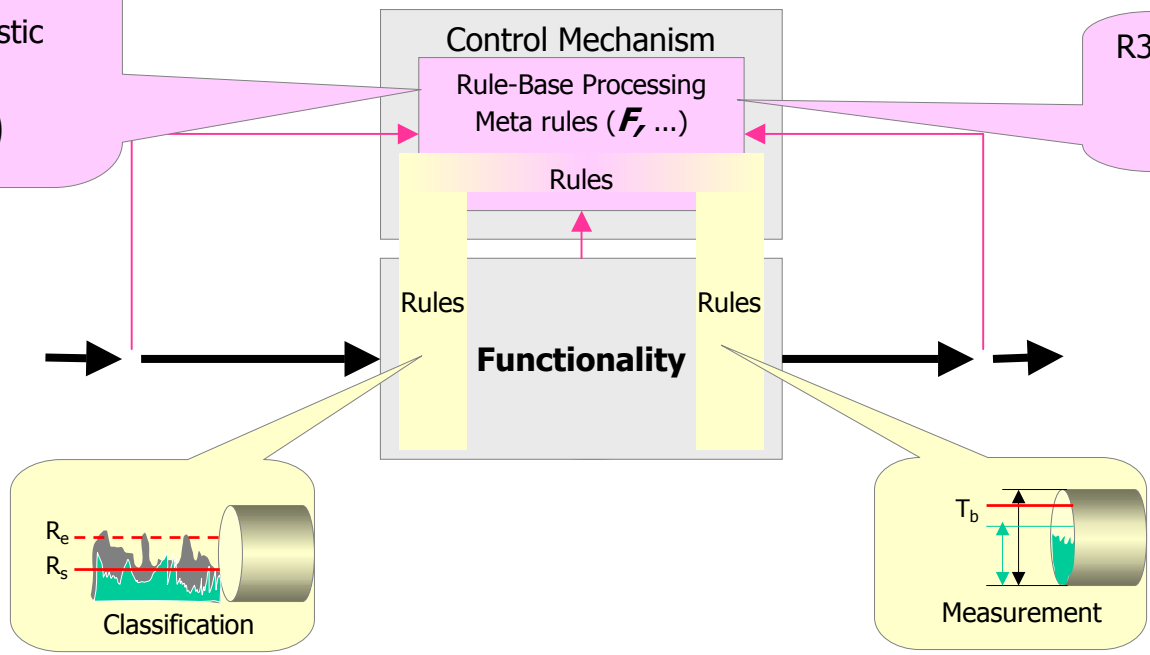


After Fernando Boavida, UoC, 2003

R1: Effective bandidth vs Threshold

R2: Streaming/Elastic ratio

$$T_b = F\left(\sum R_s / \sum R_e\right)$$



R3: Global fairness among service classes

AC as a design umbrella

- Routing
- QoS
- Security
- Mobility
- Media
- Context
- Wireless
- ...

AC Etiquette Design

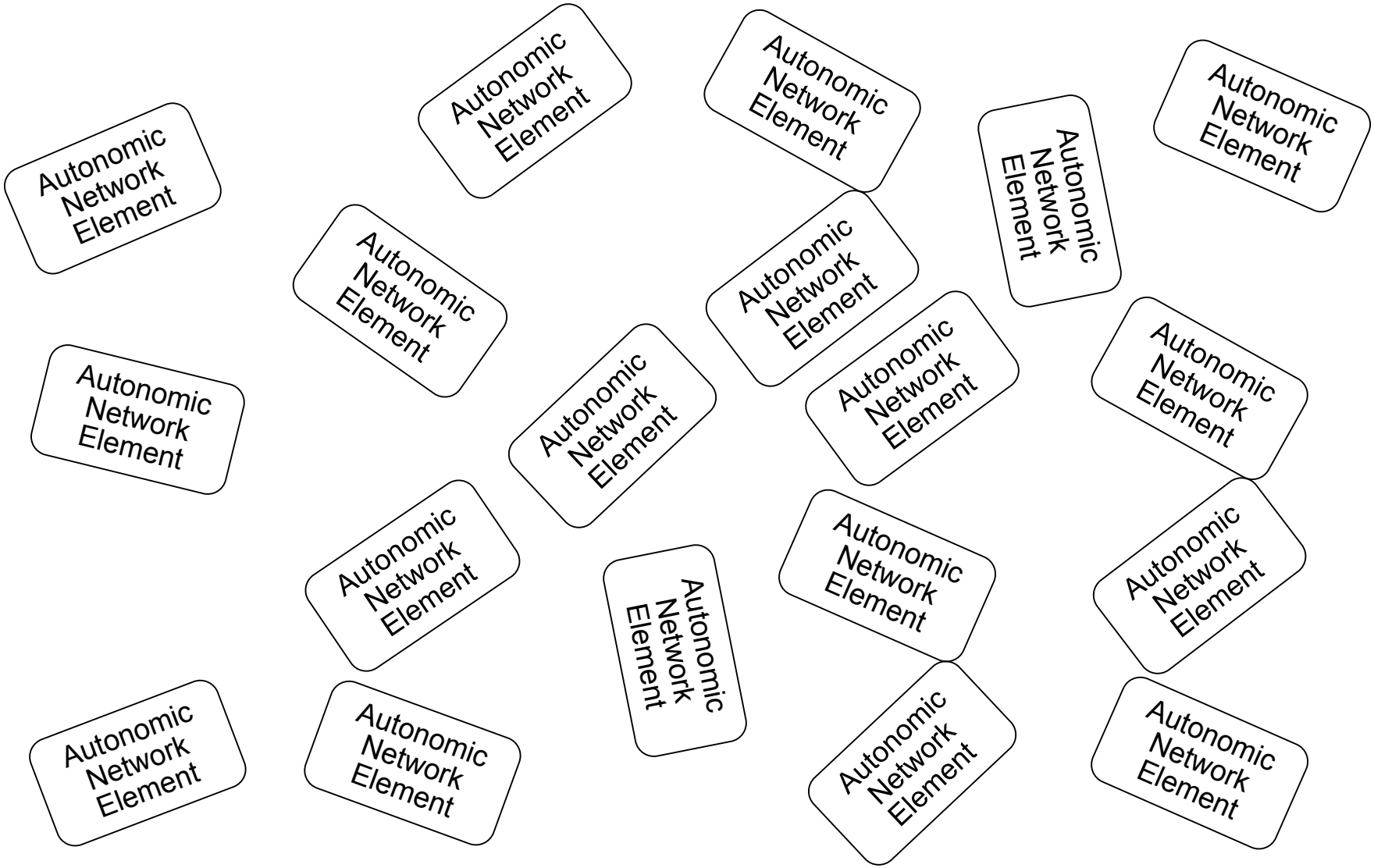
The Trend

- More flexible and fine grained in-node packet processing
  - Wired or wireless will be able to handle any protocol concurrently on the same interface
    - IETF Forces; Intel IX-based router
  - wireless will benefit from
    - Instant opportunistic selection of frequency ranges
    - Selection and composition of networks;
- communication capabilities seamlessly added
  - To things
    - We wear,
    - We use
    - We carry in our bodies

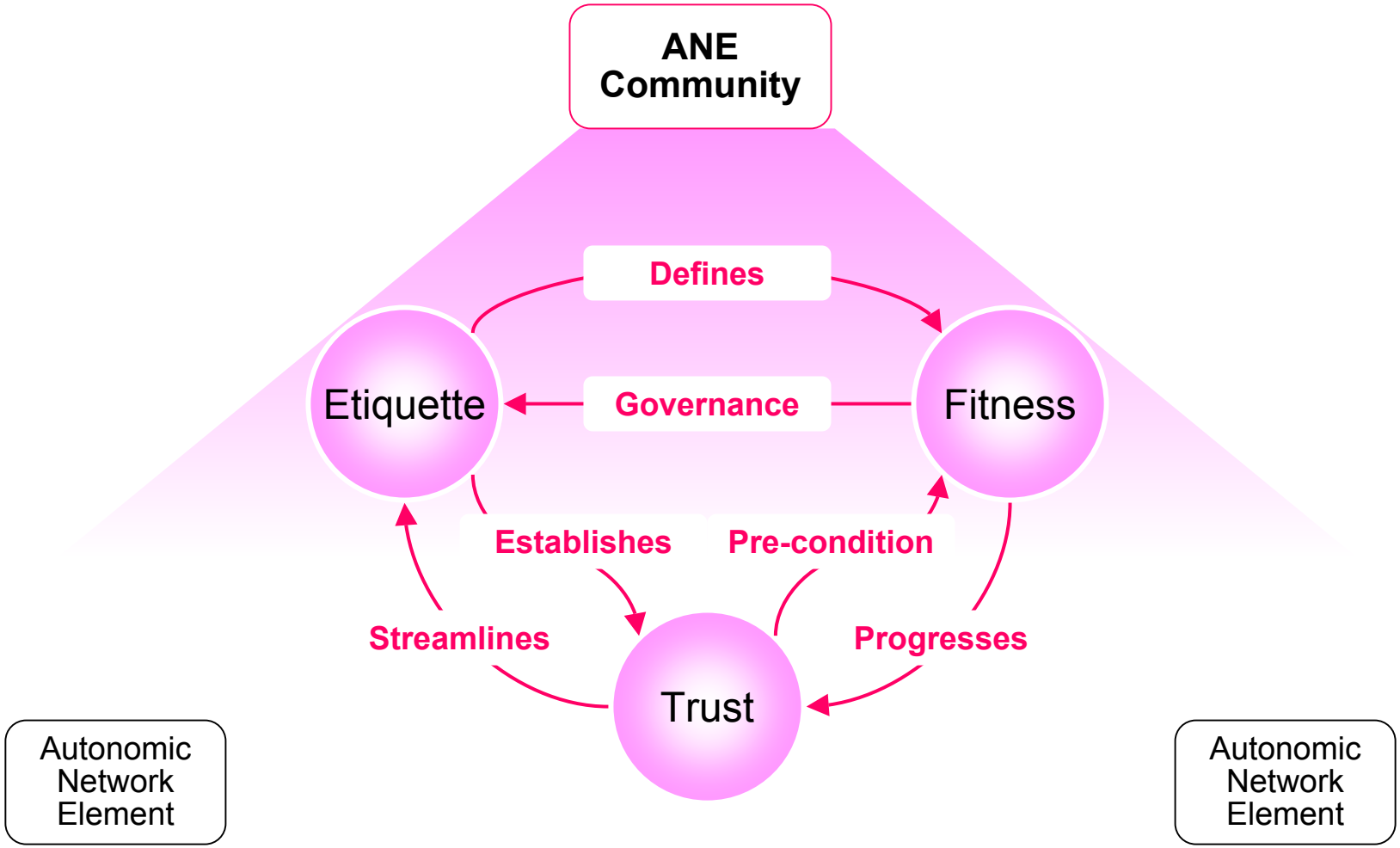
AC Design ...

- ... Is hard!
  - Many features (NL-services, aka protocols)
  - Known but not understood inter-dependencies
    - Between requirements
    - Between concerns
  - Huge gap in models, approaches, between e.g.
    - IETF (practice)
      - Solve problems at design phase (exception handling)
    - Complex Systems (approach)
      - Embed problem solver into a system (exception = normal)
- … is feasible
  - Concentrate on non-functional meta-requirements (e.g. Self-Star):
    - No system partitioning into concerns (e.g. security, management)
    - No replication
    - No need for later integration

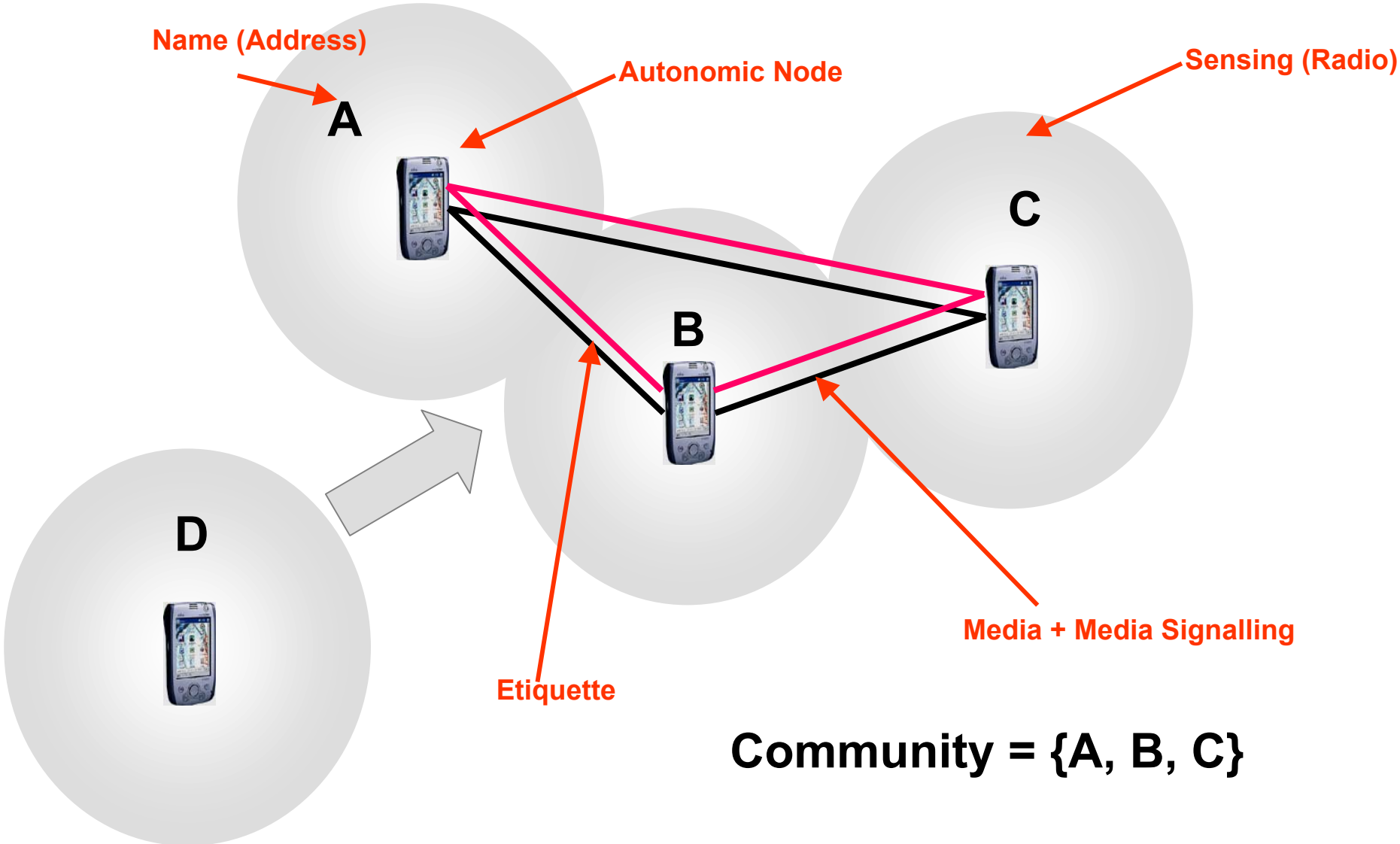
Autonomics? Hmmm...



Community Power



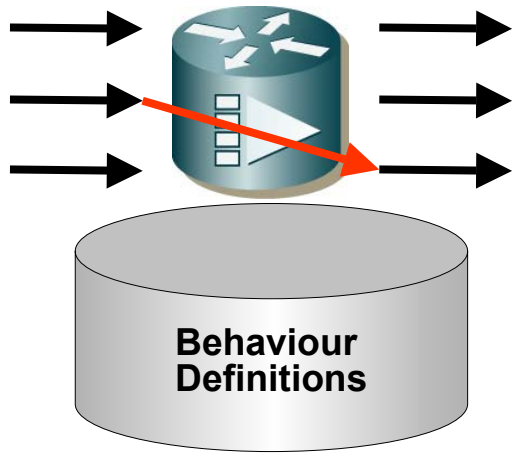
Ad Hoc Scenario



Node‘s Functionality

- Model of a Function
  F = F(component, internal_resources, local_control)
- Model of a Network Node
  - Input-Output (Why?)

	O_1	O_2	...	O_M
I_1	F(1,1)	F(1,2)	...	F(1,M)
I_2	F(2,1)	F(2,2)	...	F(2,M)
...	...	...	...	...
I_N	F(N,1)	F(N,2)	...	F(N,M)



- Examples
  - Intelligent Networks: Call Forwarding
  - Internet: Forwarding, Firewall, Network Address Translator, SIP Router, … - any function that operates with media and/or media signalling

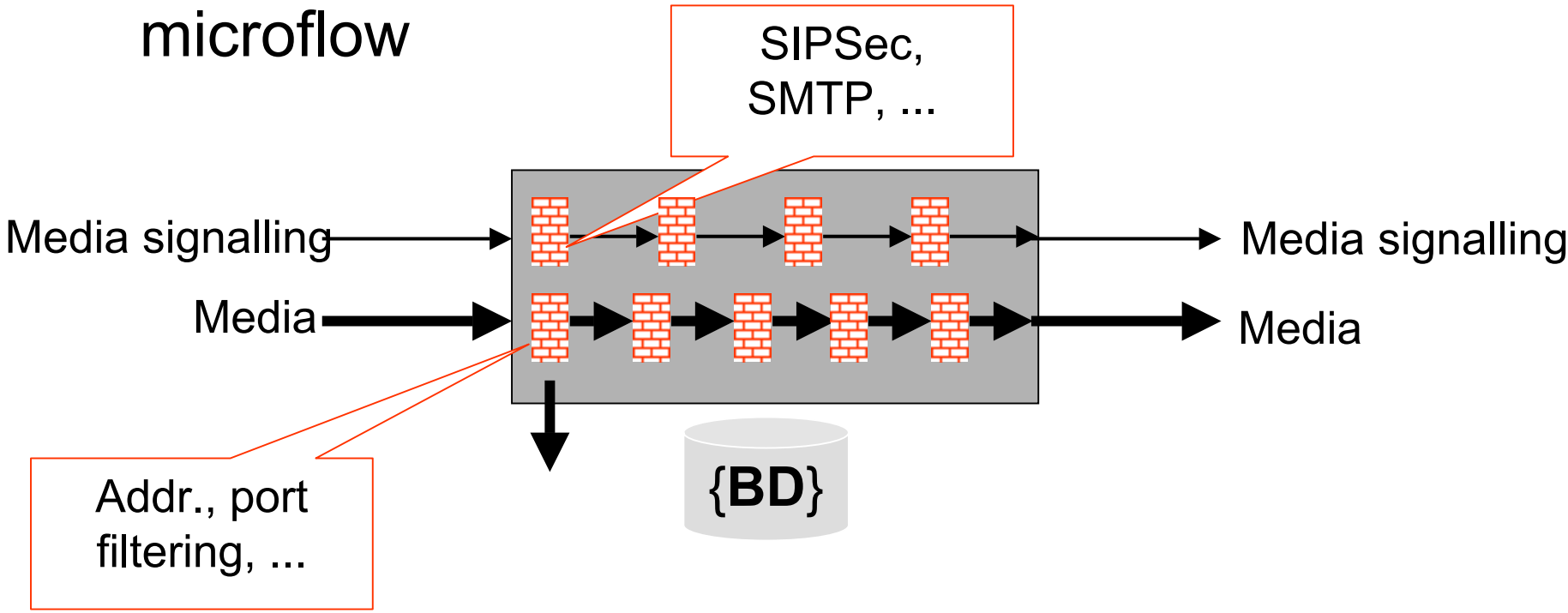
Behaviour

- How simple/complex is each F(i,j)?
- Depends!
  - Simple F(i,j) → simple behaviour → N, M ~ 10^6
  - Complex F(i,j) → complex behaviour → complex control and management
    - Hypothesis:
      IF F(i,j) is simple THEN self-star is also simple
- Atomic & generic behaviours (few*)
  - receive [-classify] [-firewall] [-execute][v4/v6 interop][SSL process] -forward [-meter -schedule] –transmit
  - Per micro-flow <Src, Dst, SrcPort, DstPort, Proto>
    - Intel's IX-based component router

*)H. Hegde, Building an IPv6 Router, Proceedings of 2002 Communications Design Con-ference, Network Processing Forum

Behaviour (2)

- Workflows - chains of atomic behaviours per microflow

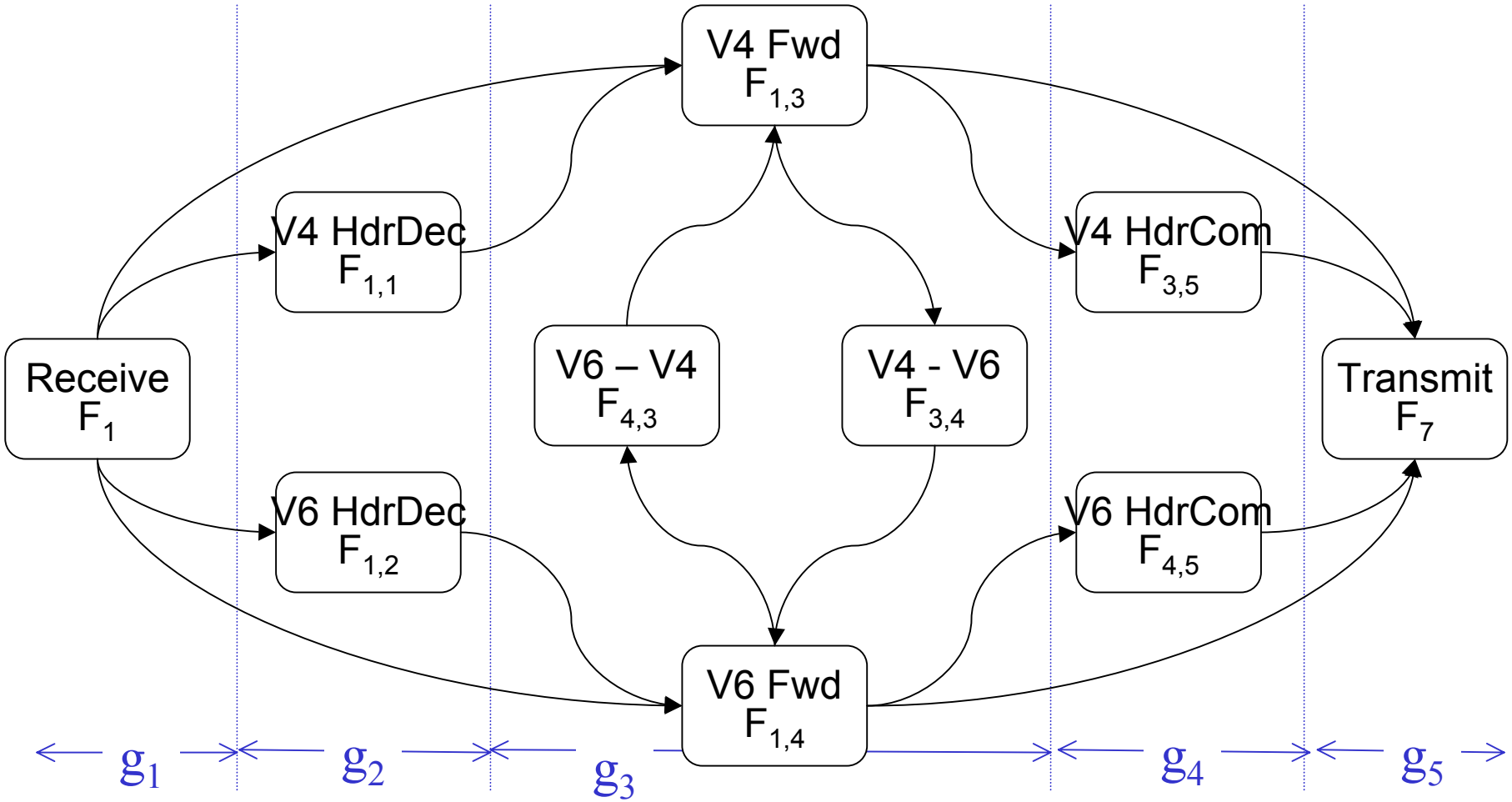


- Architecture: **pipes and filters** [AT&T]
  - Inside a node and between nodes

Description of Behaviours

- What is a behaviour?
  - Assume a grey box
    - We do not see internals but know behaviour choices (= policies)
    - Policy-based control
      - Policy is a rule defining choices in the behaviour of a system
- Community observation of behaviours
  - Message passing
    - Message := <header, payload>
      - Header ~ concern of self-*
      - Payload ~ workflow
    - Order and timing
      - FIFO
      - Soft State

Node‘s Functionality

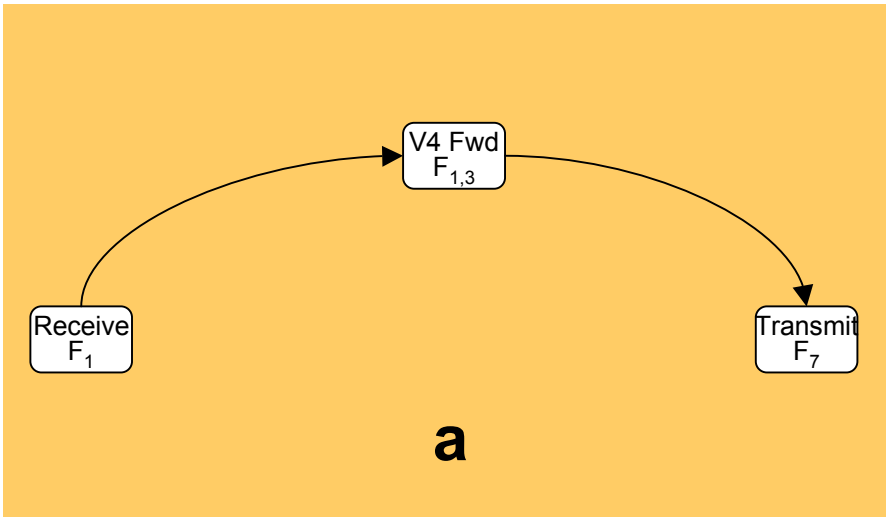


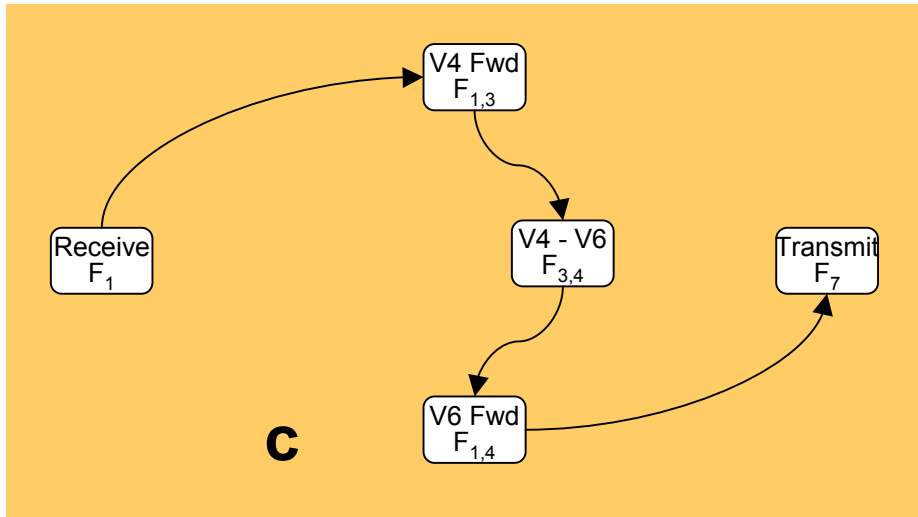
After: R. Kokku, T. Riche, A. Kunze, J. Mudigonda, J. Jason,H. Vin, A Case for Run-time Adaptation in Packet Processing Systems,
ACM SIGCOMM Computer Communication Review, Volume 34, Issue 1

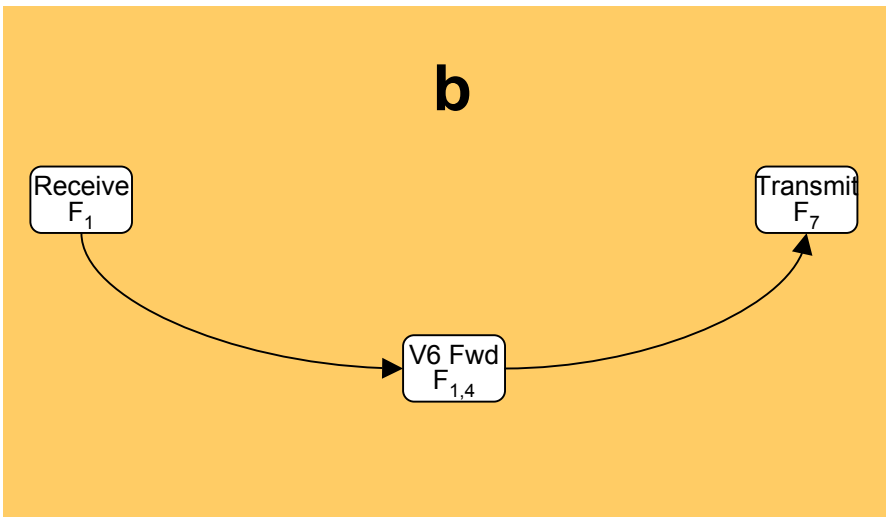
Communication of Behaviours

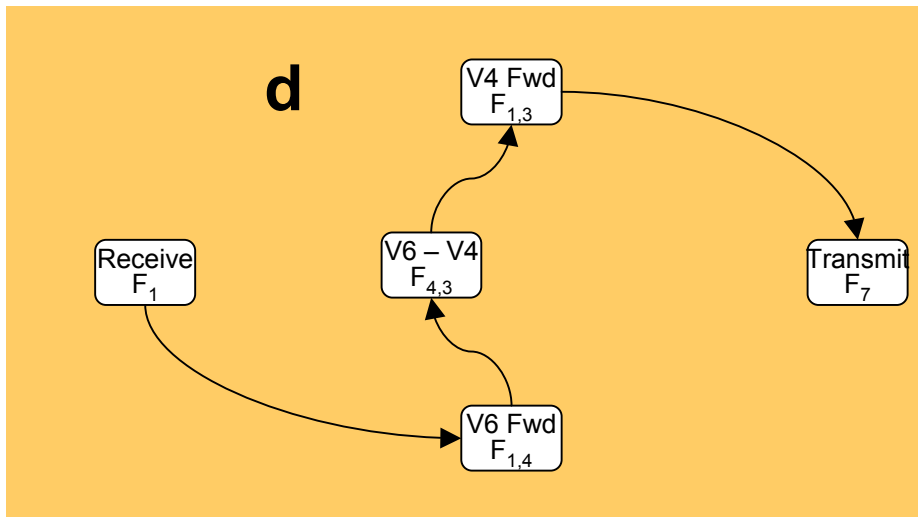
- Behaviour is a resource
- Behaviour ~ workflow
- Workflow encoding → message payload
- Behaviour concern → community address in message header
- Common self-star concern → common **etiquette rules**

Workflows



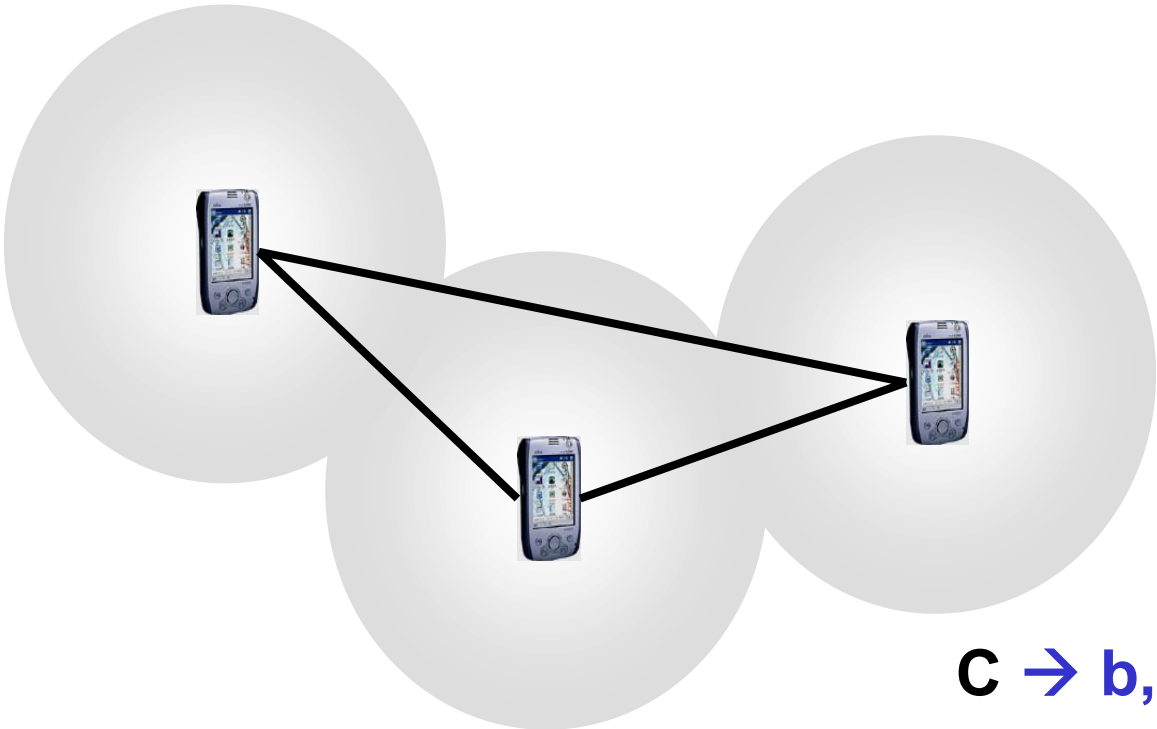






Active Workflows

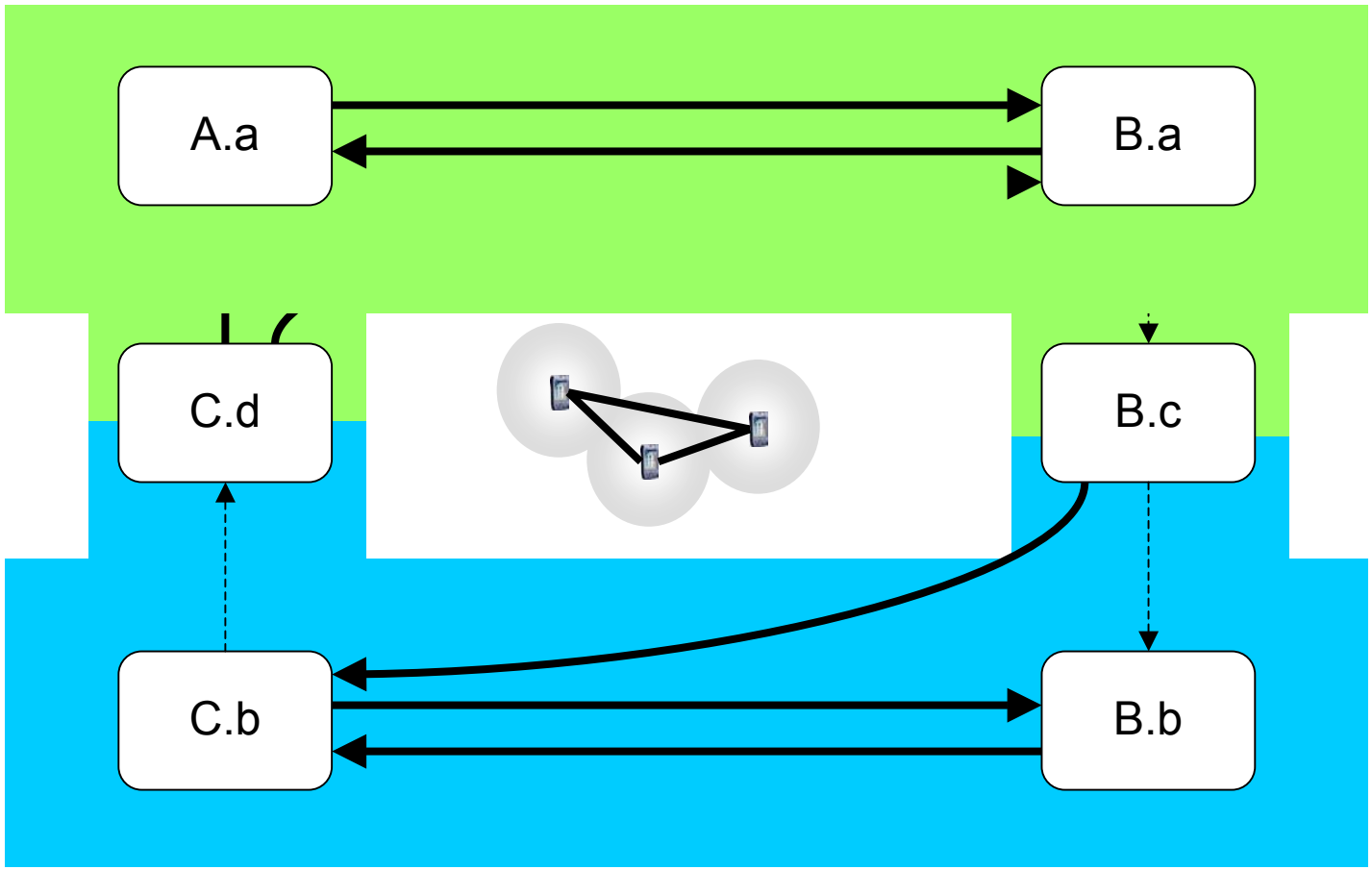
A → a



C → b, d

B → a, b, c

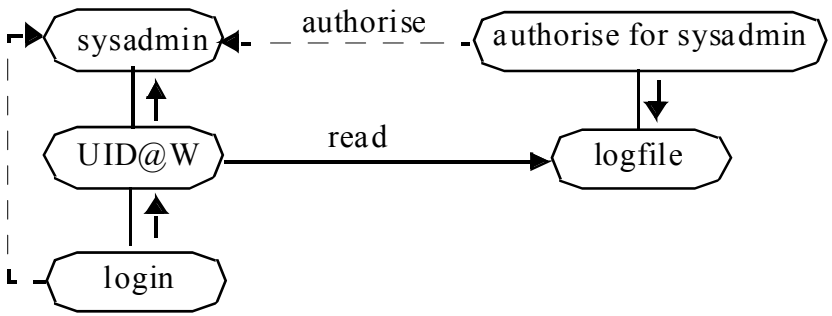
„Connectivity“



Trust By Workflow

- Autonomic nodes that cooperate at media delivery level can eventually elaborate significant trust based on successful history of common work.
- Motivation: Routing with a Clue [Sigcomm 99]

  Y. Afek, A. Bremler-Barr, S. Har-Peled, **Routing with a Clue**, Proceedings of ACM SIGCOMM '99
- Bootstrap - confirmation of the obvious:
  - E.g. 2 routers serving same flow aggregate on a link, path
- Role-based access control



A. Garg, R. Battiti, G. Costanzi, Dynamic Self-management of Autonomic Systems: The Reputation,Quality and Credibility (RQC) Scheme, Proceedings of WAC 2004, LNCS 3457, pp. 165-179

A. A. Pirzada, C. McDonald, Establishing Trust In Pure Ad-hoc Networks, Proc. 27th Australian Computer Science Conference, New Zealand, 2004

Sample Etiquette

- E0: autonomic communication message heard by a node is consumed if message header represents actual node's concern;
- E1: each active workflow is advertised to the concerned community;
- E2: every heard advertisement of a remote workflow that is locally active is consumed and notified; every consumption notification is consumed (by remote peer);
- E3: the trust per workflow is considered to be established between peers after a certain number of notifications (N_n) is exchanged

AC Network

- Only community-defined communication governed by E-Rules (Etiquette Rules) per concern
- Message payload = node‘s behaviour (media workflow)
- AC network - nodes of a media network;
- Each node is a first-class citizen;
- No distinction between user, access, edge, backbone, etc. node types
- Autonomic communication - exchange of messages with no visible relation between messages (no flow)
- Message source and destination are not necessarily always applications
  - messages sent/received on the discretion of protocol stack entities

Notation & Axioms

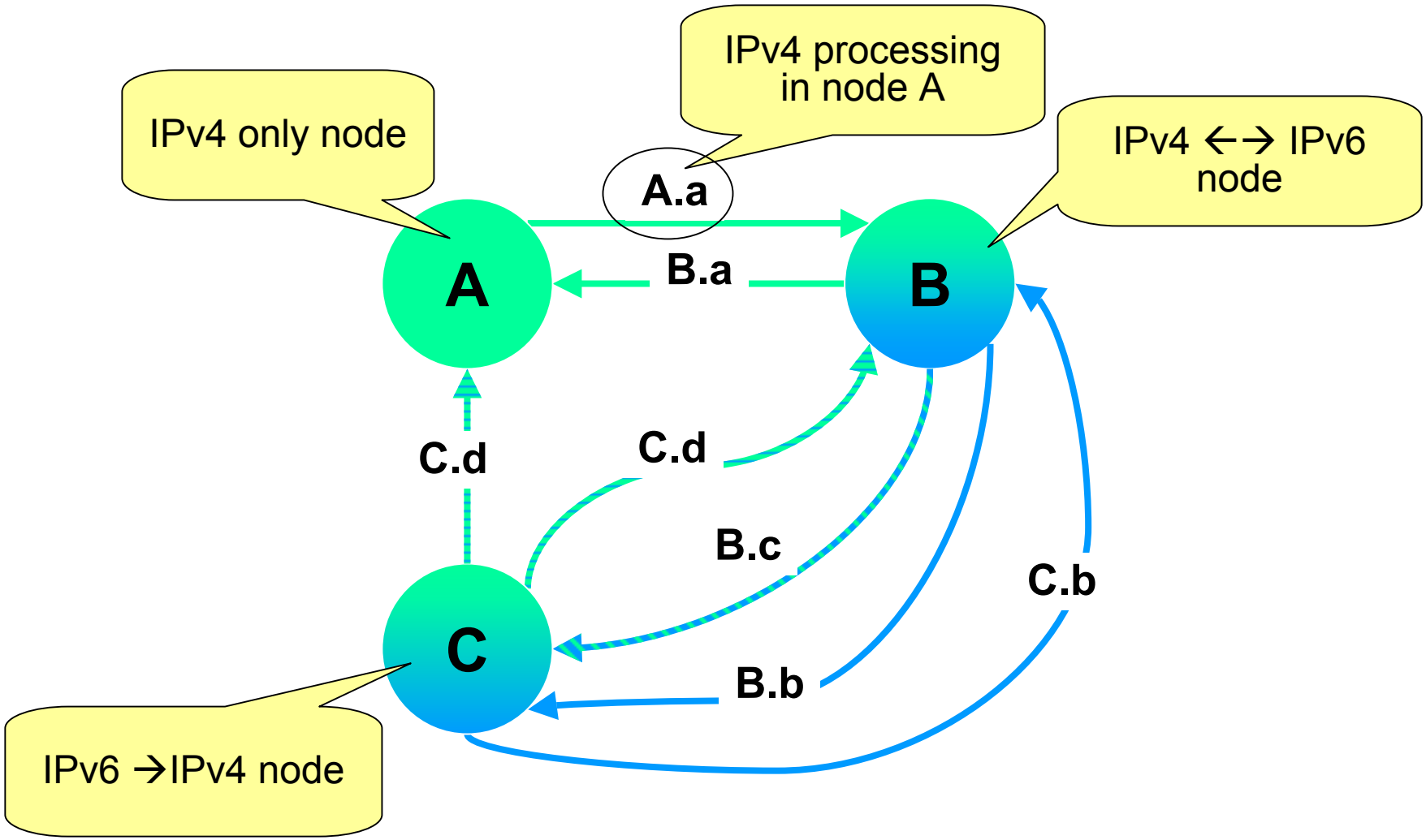
- A, B, C – nodes
- a, b, c – messages (containing a,b,c,… workflows)
- Message FIFO buffer at a node
- Extended Protocol Expressions
  - a – received message
  - 1/a – sent message
  - a – consumed message
  - 1/a – notification on consumption
  - • – [time] order, + - alternative behaviour
  - ⊗ – cross operation [Holzmann, 1982] for soundness
  - Reserved messages: z – no trust; x – trust in progress; t - trust

$$a = \frac{a}{1};\ a \bullet \frac{1}{b} = \frac{a}{b};\ \frac{1}{a \bullet b} = \frac{1}{a} \bullet \frac{1}{b};\ \frac{a+b}{c} = \frac{a}{c} + \frac{b}{c};\ \frac{a}{b+c} = \frac{a}{b} + \frac{a}{c}$$

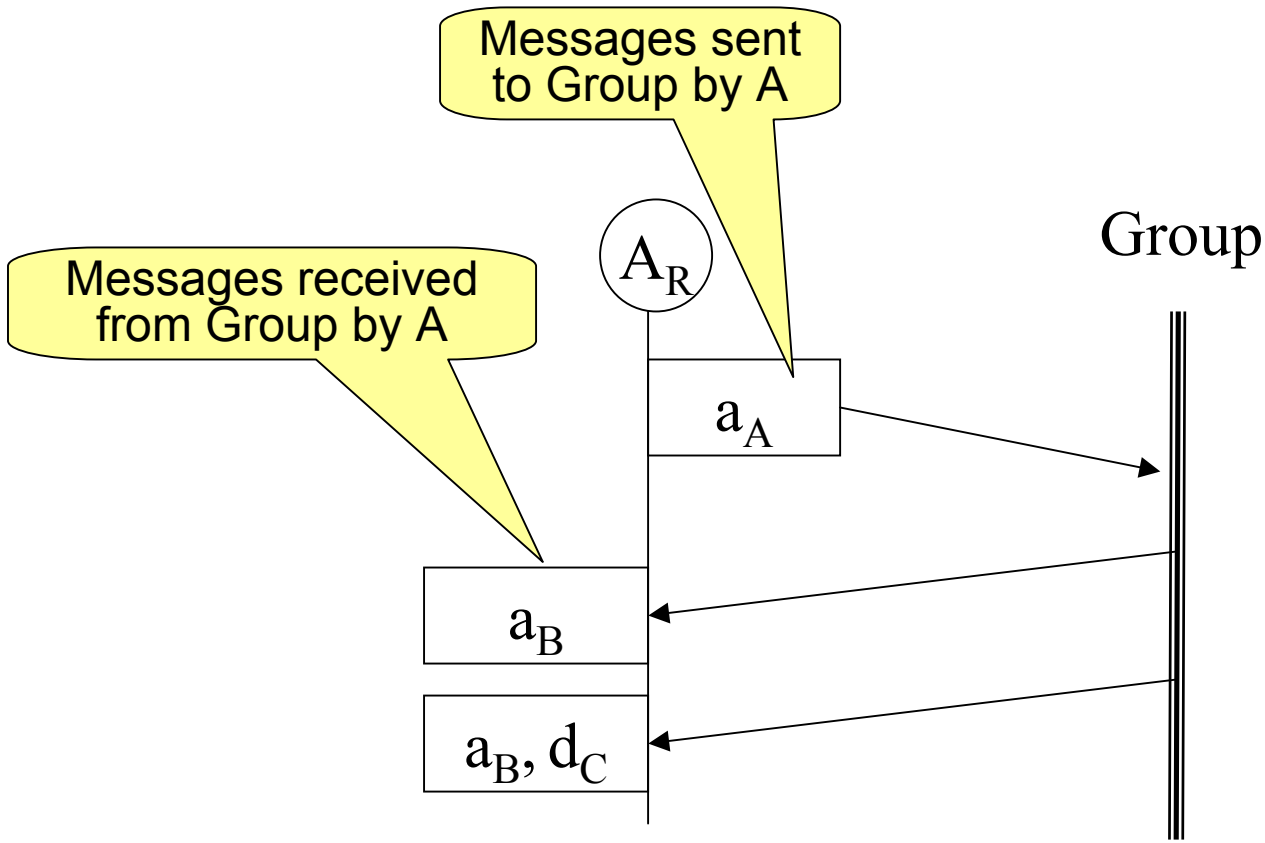
$$[B_1] \otimes B_2[\ldots]$$

G. J. Holzmann, A Theory for Protocol Validation, IEEE Trans. on Computers, Aug. 1982, Vol C31, No 8, pp. 730 - 738

Example


IPv4 only node
IPv4 processing in node A
IPv4 ←→ IPv6 node
A.a
A
B.a
B
C.d
C.d
B.c
C.b
C
B.b
IPv6 →IPv4 node

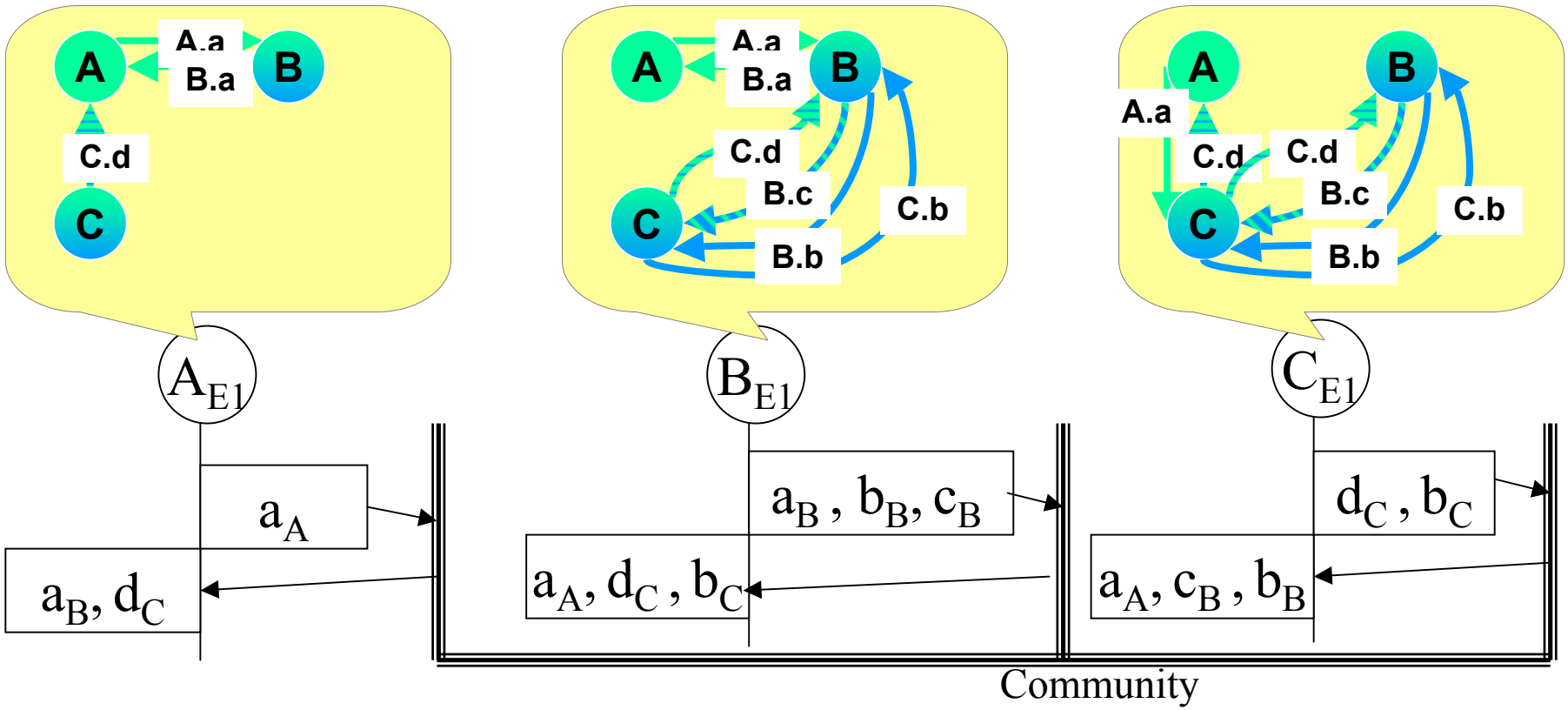
MSC


Messages sent
to Group by A
Messages received
from Group by A
A_R
Group
a_A
a_B
a_B, d_C

E1 induced behaviours

$$A_{E1} \rightarrow 1/a_A \bullet (a_B + d_C)$$
$$B_{E1} \rightarrow 1/a_B \bullet 1/c_B \bullet 1/b_B \bullet (a_A + d_C + b_C)$$
$$C_{E1} \rightarrow 1/d_C \bullet 1/b_C \bullet (b_B + c_B + a_A)$$

E1
(must advertise)



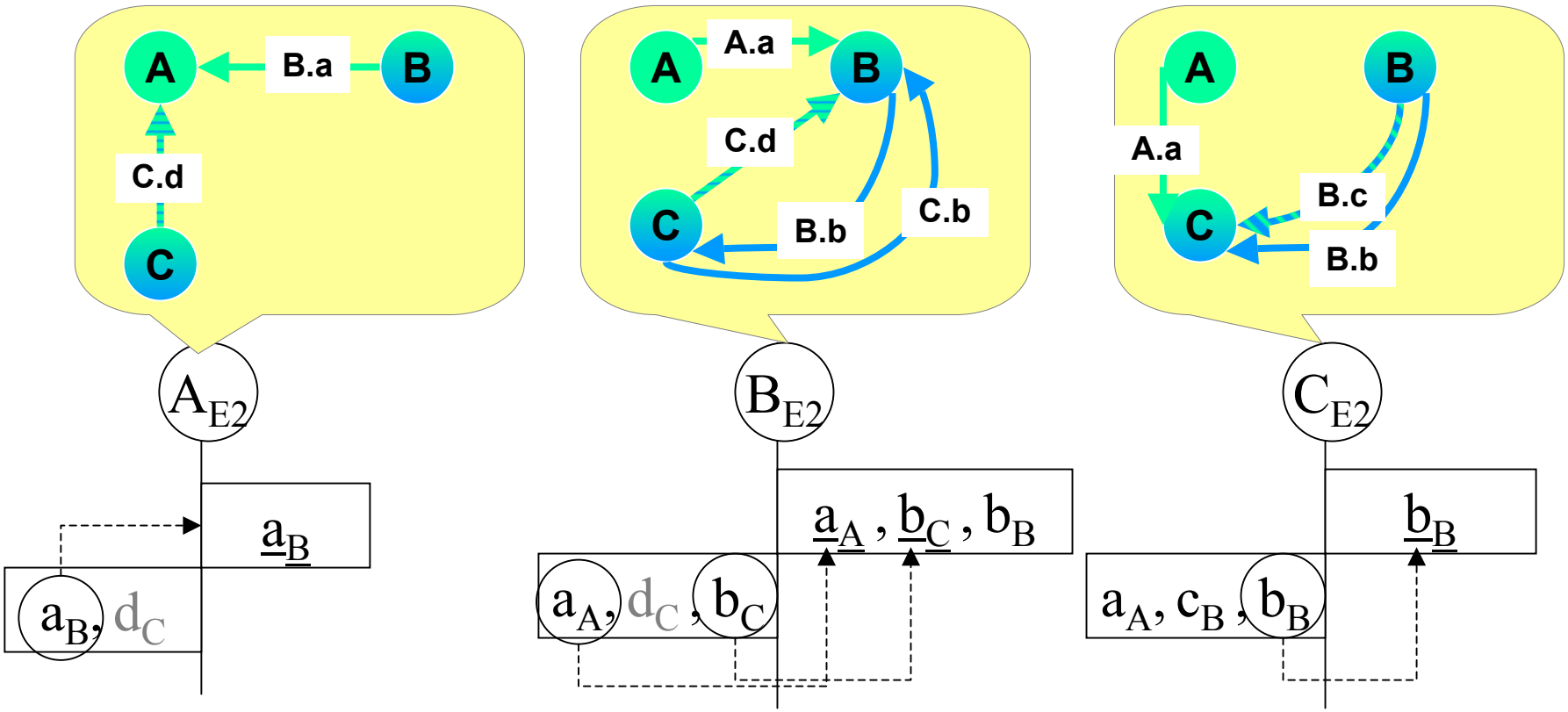
E2 induced behaviours

$$A_{E2} \rightarrow a_B \bullet 1/\overline{a_B} + \overline{a_A}$$

$$B_{E2} \rightarrow a_A \bullet 1/\overline{a_A} + b_C \bullet 1/\overline{b_C} + d_C + \overline{a_B} + \overline{b_B}$$

$$C_{E2} \rightarrow b_B \bullet 1/\overline{b_B} + c_B + \overline{b_C}$$

E2
(must consume)



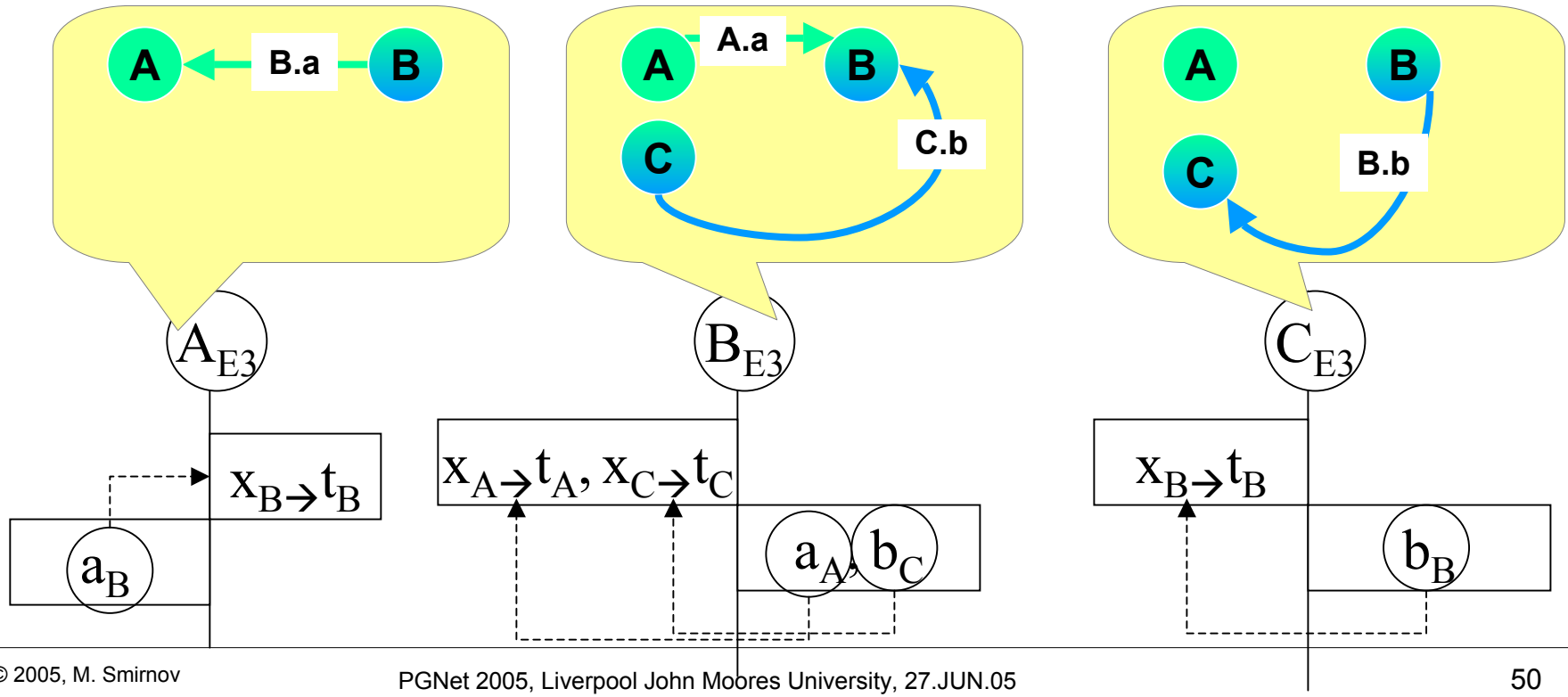
E3 induced behaviours

$$A_{E3} \rightarrow (a_B \bullet 1/\overline{a_B} \bullet 1/x_B)^{N_n} \bullet 1/t_B + \overline{x_A} + \overline{t_A}$$

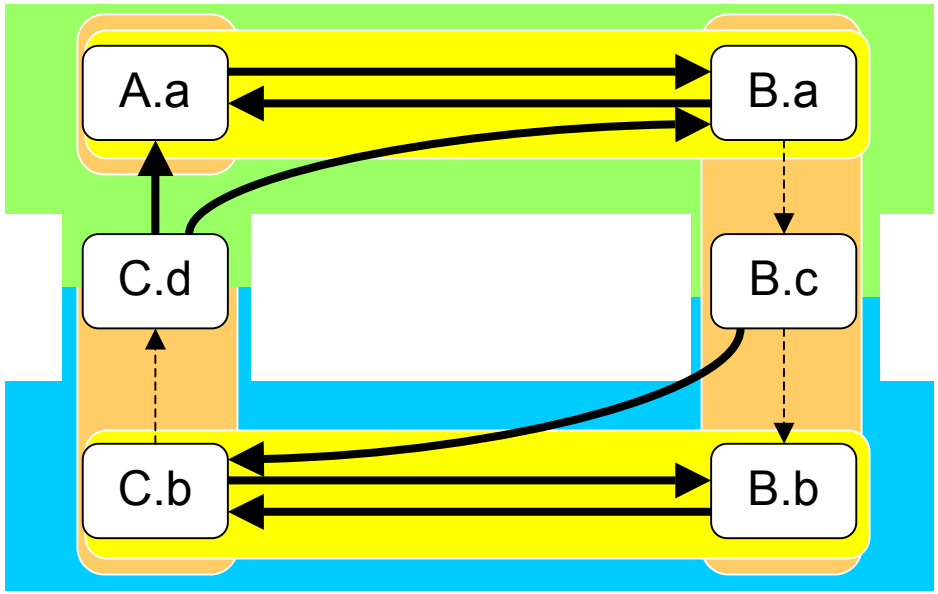
$$B_{E3} \rightarrow (a_A \bullet 1/\overline{a_A} \bullet 1/x_A)^{N_n} \bullet 1/t_A + \ldots$$
$$\ldots + (b_C \bullet 1/\overline{b_C} \bullet 1/x_C)^{N_n} \bullet 1/t_C + \overline{x_B} + \overline{t_B}$$

$$C_{E3} \rightarrow (b_B \bullet 1/\overline{b_B} \bullet 1/x_B)^{N_n} \bullet 1/t_B + \overline{x_C} + \overline{t_C}$$

E3
(must trust)



Residuals



A message box	$A_{E1} \otimes A_{E2} \otimes A_{E3} = d_C + \overline{t}_B$
B message box	$B_{E1} \otimes B_{E2} \otimes B_{E3} = 1/c_B + d_C + \overline{t}_A + \overline{t}_C$
C message box	$C_{E1} \otimes C_{E2} \otimes C_{E3} = 1/d_C + c_B + \overline{t}_B$

Fitness

Community nodes with trust in progress

Advertised workflows

Trusted (wf) community nodes

Node Fitness	$F_{\varphi}(i) = \frac{N_t(i) + \omega \cdot N_x(i)}{N} \cdot \frac{n_w(i)}{\overline{n_w}}$
Community Fitness	$F_{\varphi}(C^o) = \frac{\sum [N_t(i) + \omega \cdot N_x(i)] - N}{\sum n_w(i)}$

Known community nodes

Known workflows

Weighted Fitness

Node	Node Fitness	Community Fitness All nodes	Community Fitness Without	Node Fitness Weight	Weighted Node Fitness
A	0,22	0,67	0,40	k_A=0,60	0,13
B	0,50		0,00	k_B=1,00	0,50
C	0,50		0,50	k_C=0,50	0,25

$$F_{\varphi,\,w}(i) = \left(1 - F_{\varphi}\left(\tilde{C}^{o}\right)\right) \cdot F_{\varphi}(i) \qquad i \not\subset \tilde{C}^{o}$$

Fitness-based governance: etiquette progression

Inferring new etiquette

Exchange between trusted peers:

A box	$A \rightarrow (d_C + \overline{t_B}) \bullet (1/c_B + d_C + \overline{t_A} + \overline{t_C})\big\|_B$
B box	$B \rightarrow (1/c_B + d_C + \overline{t_A} + \overline{t_C}) \bullet \left[(d_C + \overline{t_B})\big\|_A + (1/d_C + c_B + \overline{t_B})\big\|_C\right]$
C box	$C \rightarrow (1/d_C + c_B + \overline{t_B}) \bullet (1/c_B + d_C + \overline{t_A} + \overline{t_C})\big\|_B$

Behaviour suggested by node A (modified E1 induced behaviour)

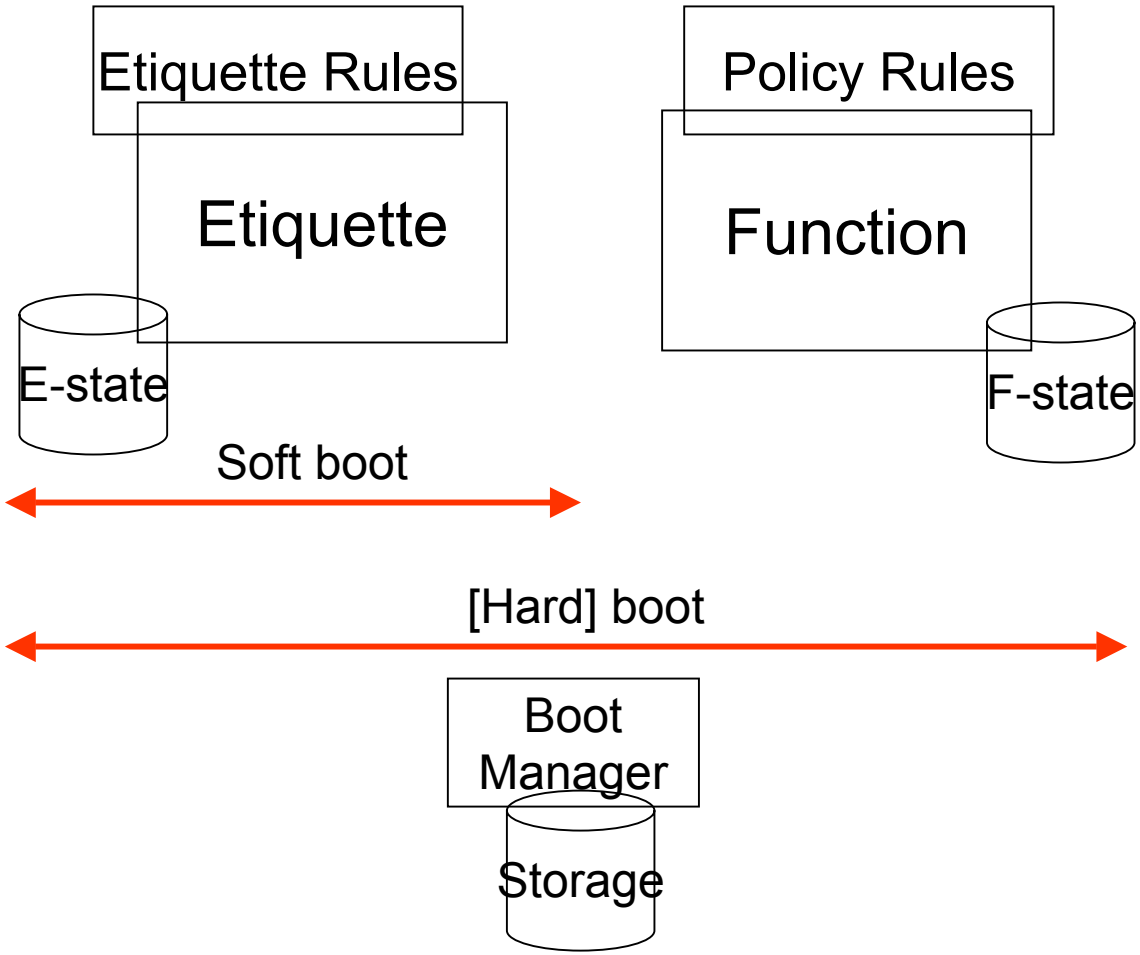
$$A_{E1'} \rightarrow (1/a_A \bullet d_C) \,\|\, (1/c_B \bullet 1/b_B)\big|_B \,\|\, (b_C \bullet d_C)$$

V4/v6 interoperability

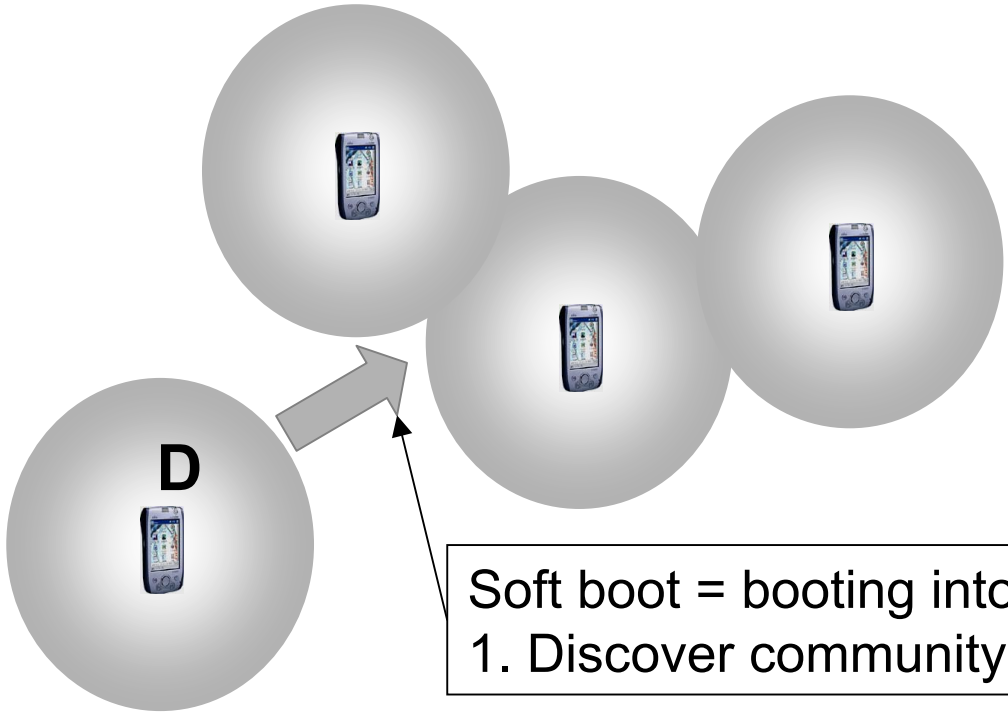
v6 forwarding

V6/v4 interop.

Bootstrapping



Boostrapping



Soft boot = booting into a community
1. Discover community Etiquette(concern)

$$D_{E1} \rightarrow 1/y_{D,1} \bullet [1/y_{D,2}]$$

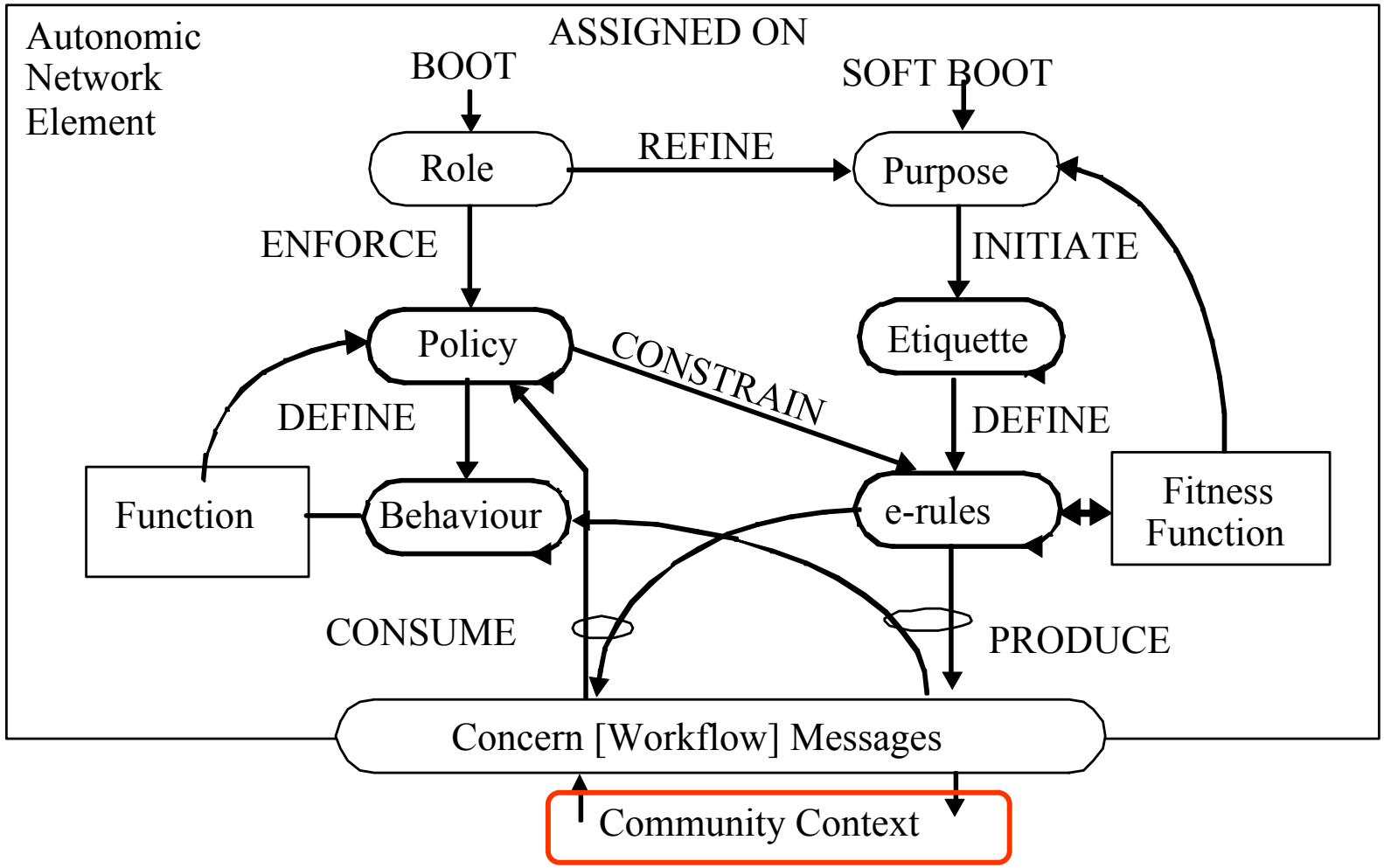
$$D_{E2} \rightarrow y_{D,1} \bullet 1/\overline{y_{D,1}} + [y_{D,2} \bullet 1/\overline{y_{D,2}}] + [\ldots] + \overline{y_{Y,1}} + \overline{[y_{Y,2}]}$$

$$D_{E3} \rightarrow (y_{D,1} \bullet 1/\overline{y_{D,1}} \bullet 1/x_Y)^{N_n} \bullet 1/t_Y + [\ldots]$$

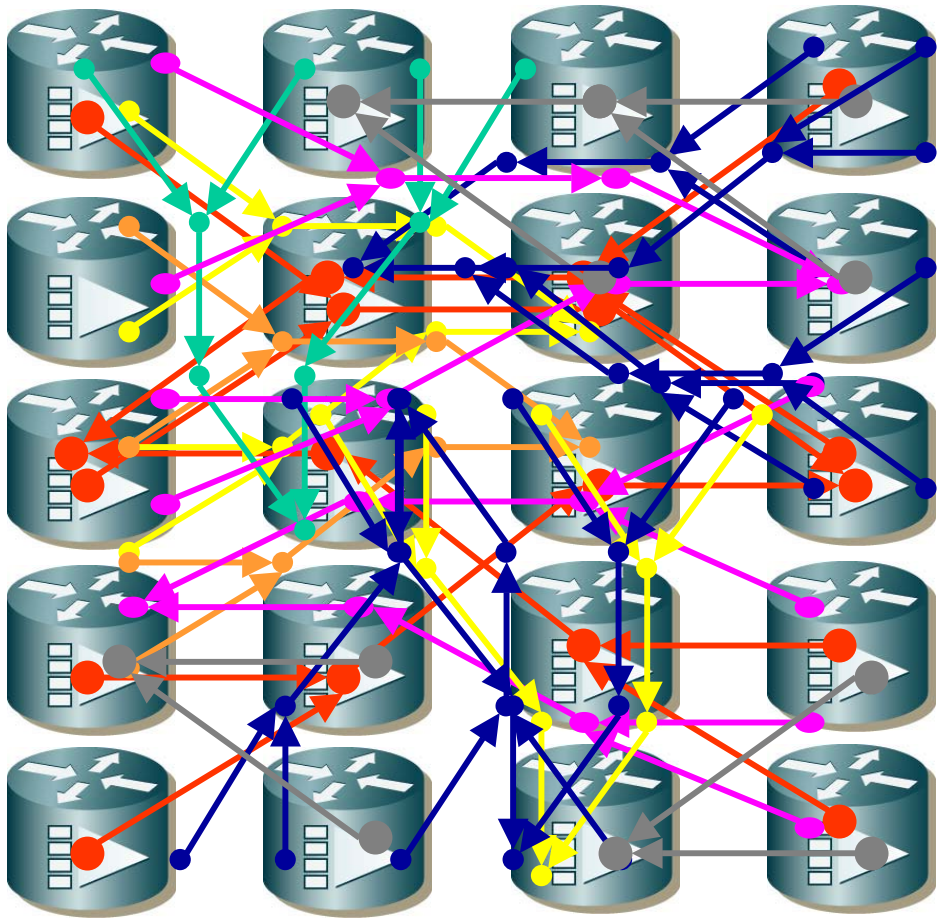
Four Takes

Instead of conclusion

Take on Context



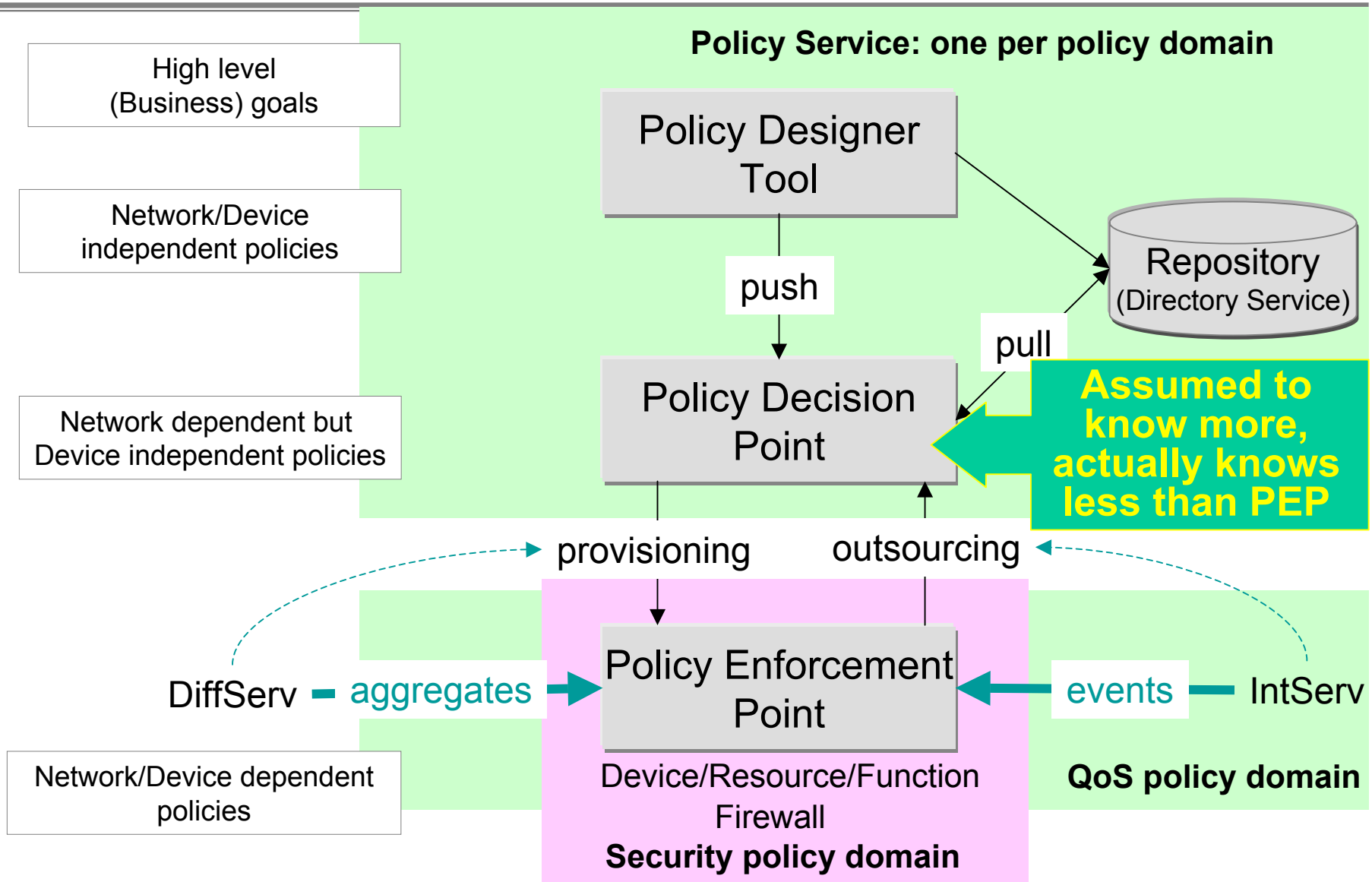
Take on sensor networks



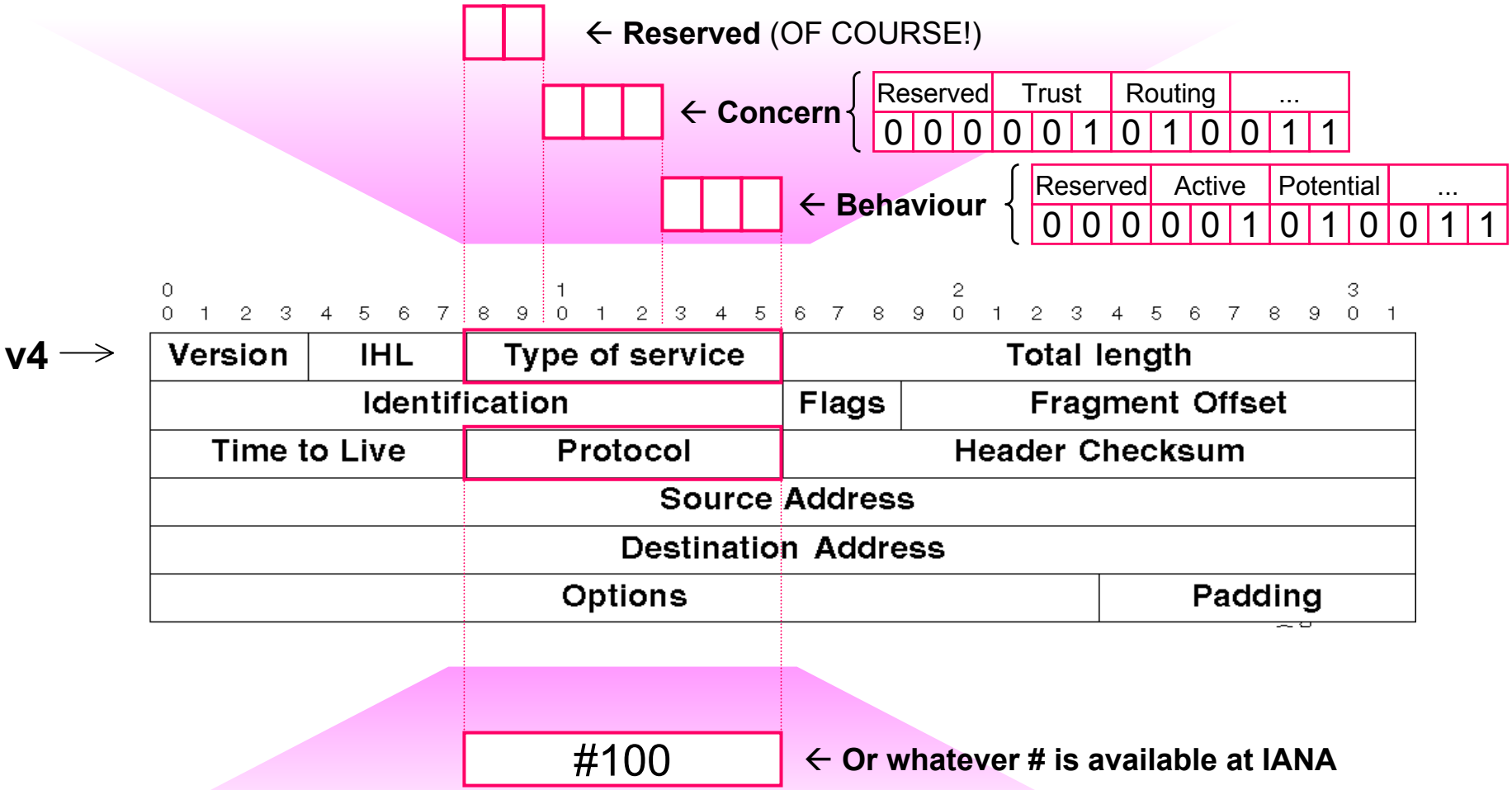
“Sensor-actor combined node will find much more applications”

[I. Akyildiz, Invited Talk at WWIC 2005, Xanthi, Greece]

Take on Policy



Take on IP


← Reserved (OF COURSE!)
← Concern
Reserved Trust Routing ...
0 0 0 0 0 1 0 1 0 0 1 1
← Behaviour
Reserved Active Potential ...
0 0 0 0 0 1 0 1 0 0 1 1
0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
v4 →
Version
IHL
Type of service
Total length
Identification
Flags
Fragment Offset
Time to Live
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding
#100
← Or whatever # is available at IANA

Thank you for your attention

- Learn more at www.autonomic-communication.org
- Discuss at www.autonomic-communication-forum.org
- Participate in
  www.autonomic-communication.org/wac
  www.autonomic-communication.org/acc
  www.co-next.net
- Submit to www.autonomic-communication.org/annales