Some challenges for architectures and protocols in resource limited environments and systems

Saleem Bhatti

http://www.dcs.st-andrews.ac.uk/~saleem/

Networked and Distributed Systems (NDS) Research Group

http://nds.dcs.st-andrews.ac.uk/

School of Computer Science, University of St. Andrews

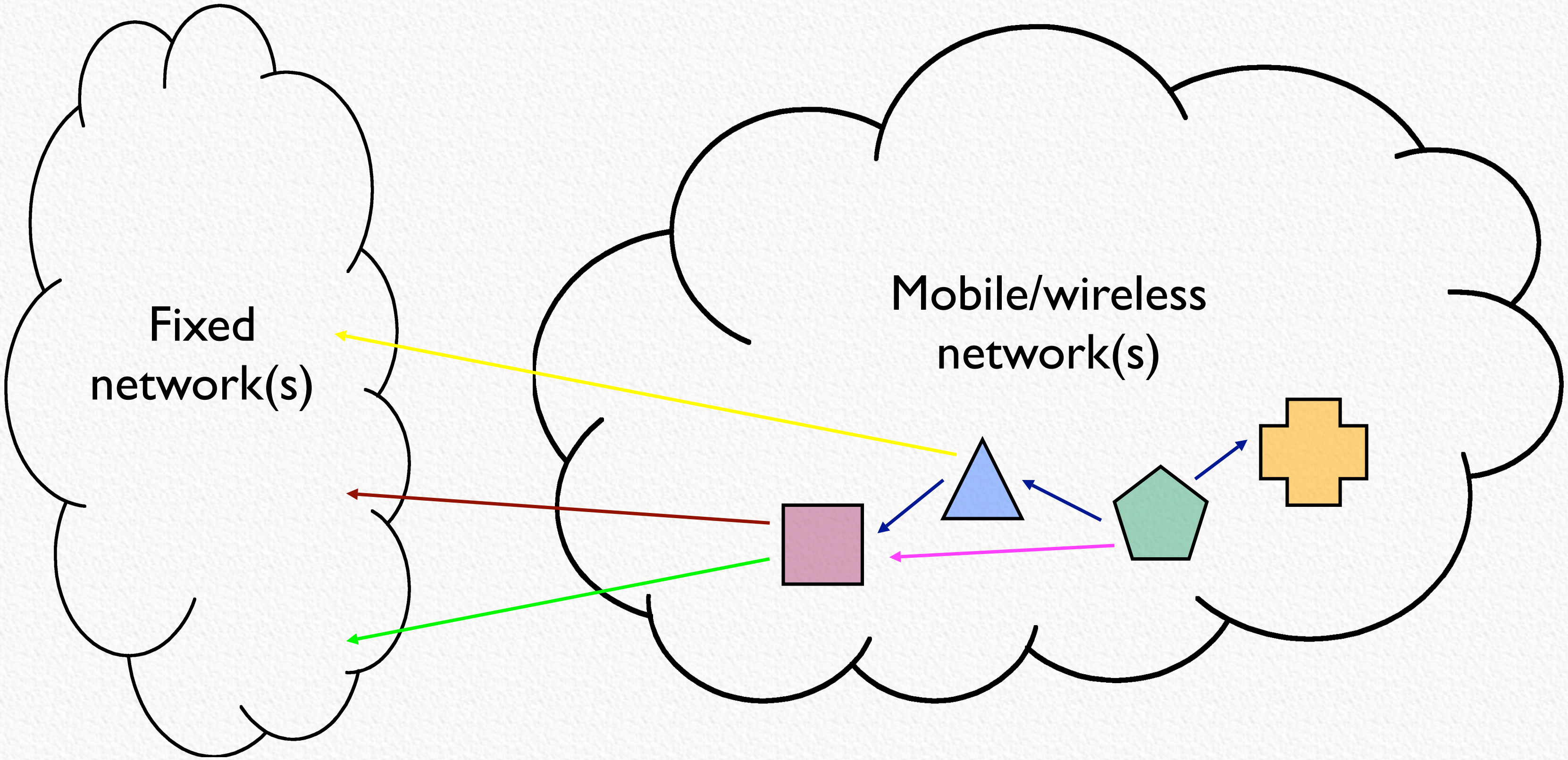
Devices - range of capability

- PDAs & Smart Phones:
  - Some PDA functions
- Media devices:
  - Some PDA functions
- Hand-held games:
  - Some PDA functions
  - Phone functions
- (Ultra) Thin clients:
  - **low** cost devices
- Embedded systems:
  - Commercial
  - Domestic
  - Personal
- Sensors:
  - Wide ranging
- RFIDs and other tags
- Smart Dust
- ???

Resource limited environments and systems

- **Limited** networking capability:
  - Connectivity (variable and diverse)
  - Capacity (different technologies)
  - QoS (e.g. error rates, packet loss - fluctuations)
- **Limited** end-system capability:
  - Operating system and/or execution environment
  - CPU (computational power and functionality)
  - Power
  - Storage (including memory)

General Scenario



Communication and access to other resources

Edge networks

*... for the Internet, much of the creative energy is at or near the **edge** of the network.*
*It is at the edge that most **applications** are created.*
*It is at the edge that most **devices** are connected.*
*It is at the edge where we usually see the development of **new networking technologies**.*
*It is at the edge of the network where the **economic conditions most favor innovation**, as the barrier to entry (for applications, devices, and networking technologies) is typically lower at the edge.*

*So if we want to think about where networking might be in 10 or 15 years, it behooves us to look at **(r)evolution at the edge**.*

Clark et al, *'Making the world (of communications) a different place'*
ACM Computer Communications Review, vol. 35, no. 3, July 2005, pp 91-96
http://portal.acm.org/citation.cfm?id=1070873.1070887

Edge networks of resource limited devices

- By ~2010 the number of processors produced each year will double.
- ~75% of all processors are 8-bit or 16-bit.
- ~90% of all processors are used in embedded systems.
- There are more mobile phones in the world than there are hosts connected to the Internet.
- Resource limited edge devices are more likely to be mobile.

(My) List of Challenges [1]

Technological challenges

- Ubiquitous connectivity - 'always connected':
  - use any connectivity available
- End-system platform - environment and APIs:
  - the 'correct' end-system paradigm
- Storage - reliable, highly-available, distributed:
  - files, data and media flows
- Power management:
  - conservation of limited (battery) power

(My) *List of Challenges [2]*

Architectural challenges

- Robustness - dependability of systems:
  - systems should not be *brittle*
- Holistic design - co-design, embedded systems:
  - consideration of the whole system
- Naming, addressing and routing:
  - a cleaner separation between identity and location
- Security - end-system and network defence:
  - in the system design and *not* retro-fitted fixes

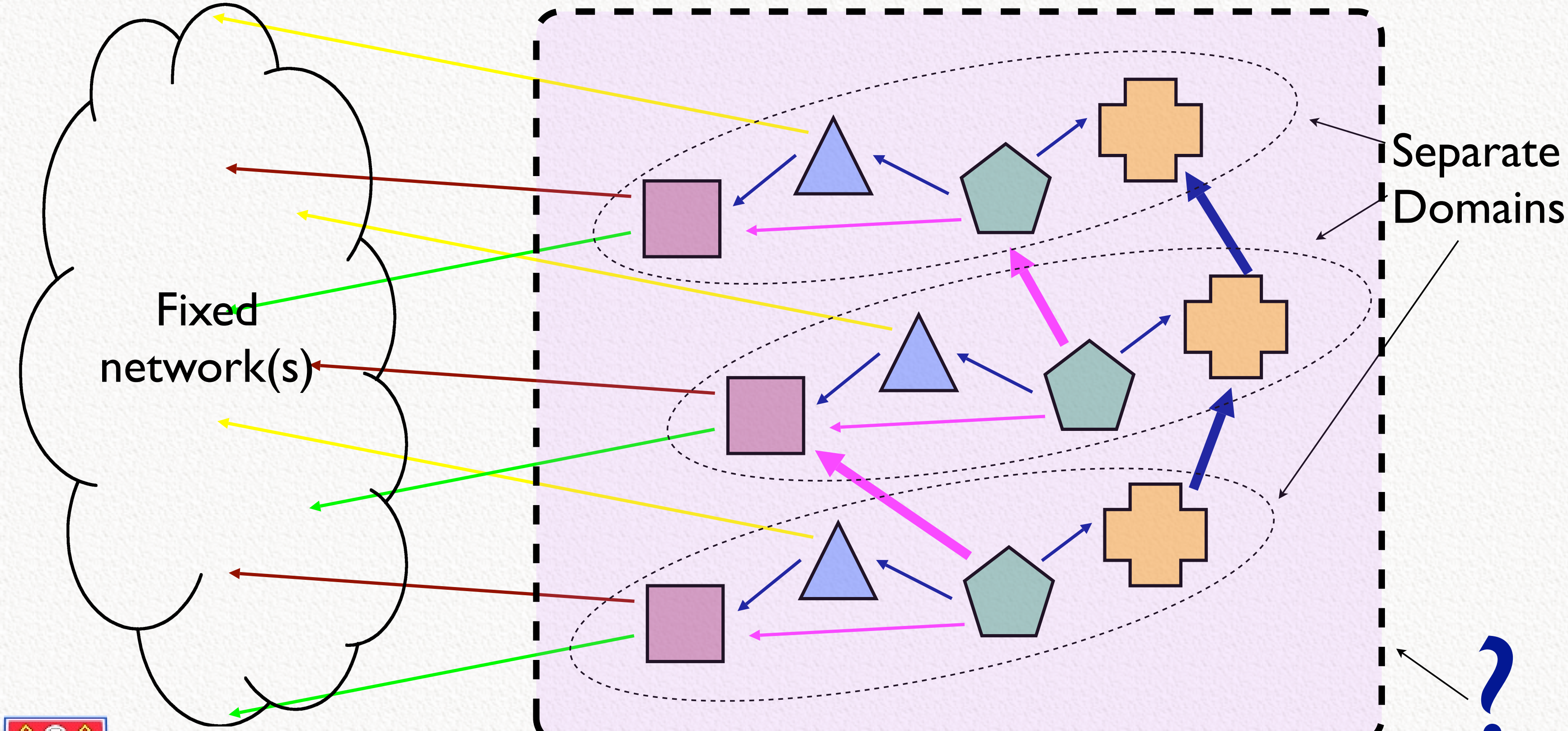
Meeting the challenges

- Diversity of skills required
- Co-operation between academic institutes
- Co-operation between industry and academic
- Co-operation between disciplines:
  - computer science and engineering
- Leverage skills and knowledge
- **Collaboration is key**

Ubiquitous connectivity [1]

- Many devices have multiple connectivity:
  - e.g. GSM, GPRS, 3G
  - e.g. WLAN (802.11*)
  - e.g. Bluetooth
- Some have only local connectivity
- Some have both local and 'global' connectivity
- Ubiquitous connectivity through:
  - sharing local(ised) connectivity
  - multi-homed 'global' connectivity

Ubiquitous connectivity [2]



Coalition Peering Domain (CPD)

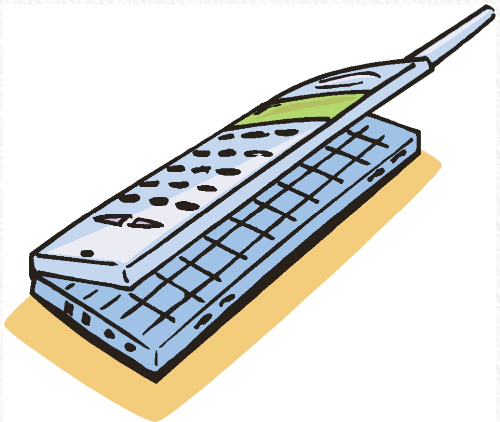
- Ongoing work at St. Andrews and UCL
- Coalition Peering for communication:
  - explicitly collaborative
  - involves resource sharing
  - requires shared policy
- Multi-homed, multi-path connectivity
  - robust communication
- Many applications:
  - emergency communications, sensor arrays

CPD - simple application [1]

3G

128-384 Kb/s

This ...



... is much less ...

802.11

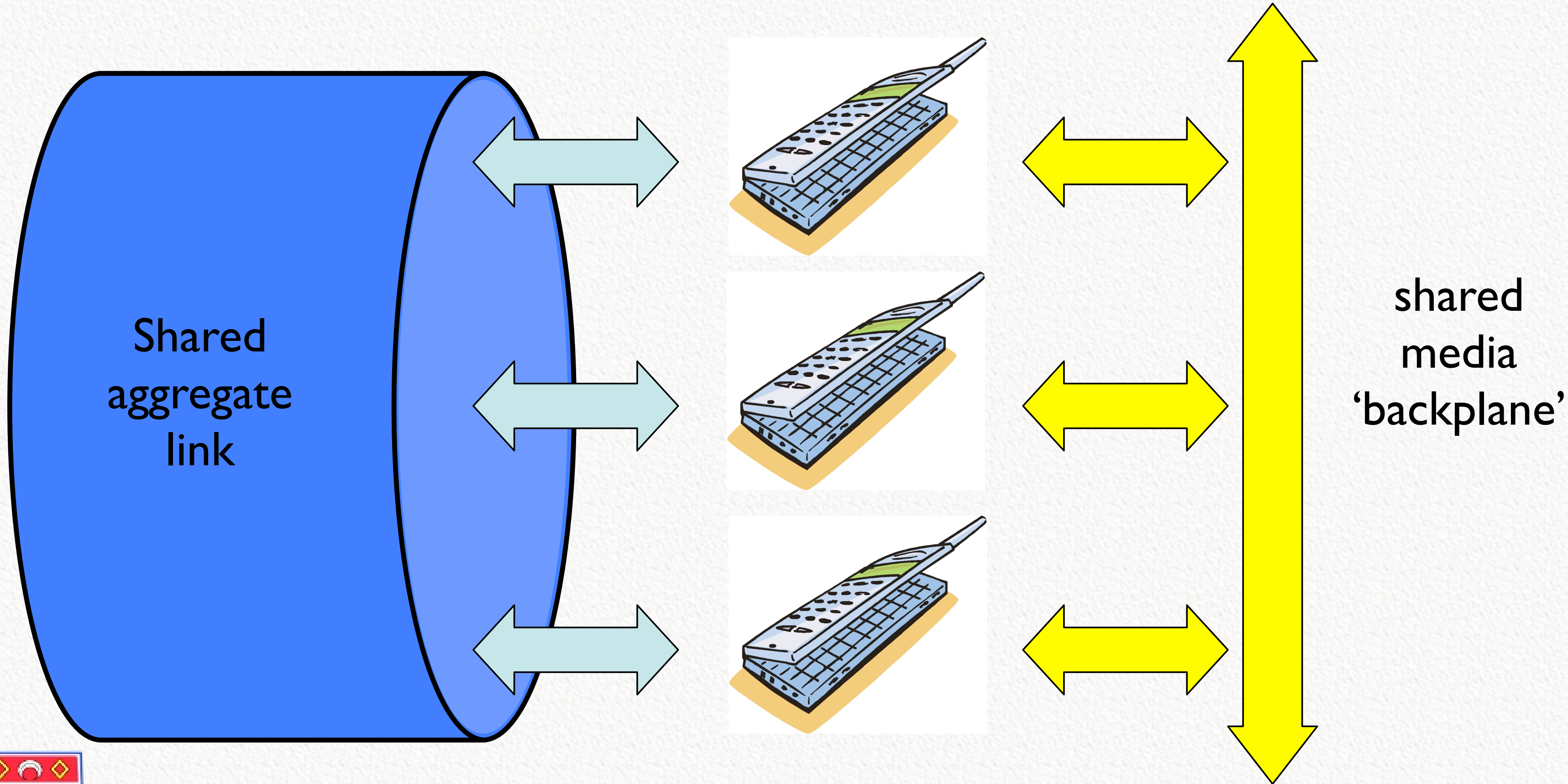
11-54 Mb/s

... than this.

CPD - simple application [2]

- Shared media interface using 802.11:
  - Local only (up to 100m)
  - For data applications
  - Greater capacity than individual 3G link
  - Integrated with handset
- Use shared media interface as 'backplane':
  - **Allow 3G link aggregation**
  - **Share aggregated links between users**
  - **Gain through natural stat-mux of data**

CPD - simple application [3]



CPD - *simple application [4]*

Example use scenarios

- Domestic use examples:
  - Sharing downlink capacity for media streaming
  - Sharing uplink capacity for peer-to-peer applications with remote users
  - Faster 'web-surfing'
- Commercial use examples:
  - Sharing by commuters on a train
  - End users establish *dynamic* hotspots
  - 'In the field' emergency services, e.g. Police, Rescue, etc.

CPD - technical challenges

- Naming, addressing and routing:
  - IP does not handle well multi-homing + multi-path
  - address management in the CPD
  - applications protocols: flow identification
  - dealing with mobility
- Transport protocols:
  - end-point identification, multi-point connectivity
  - congestion control algorithms
- CPD management

End-system platform

- Port a "full" operating system?
  - **general purpose?**
  - multiple file systems?
  - fine-grained process scheduling control?
  - large I/O library?
  - extent of memory management?
  - support for high-level language constructs?
- Or ... a lightweight execution environment?
  - highly modular and configurable

Determinism and Reliability

- Limited resources?
- Execution time?
- Memory usage?
- Computational cost?
- Power costs?
- Can these things be evaluated easily?

Dynamic components with certified constraints on execution time, memory usage, computational cost and power consumption

Higher-order Unified Meta-Environment - HUME

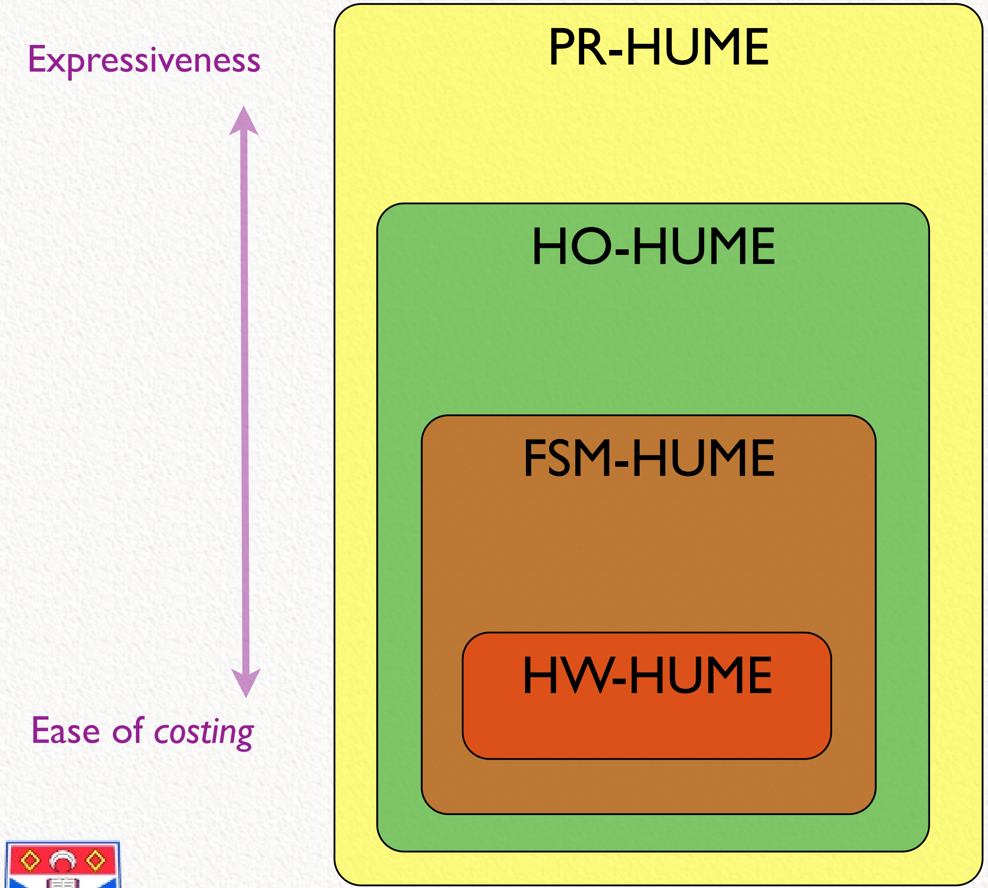
***Hume** (Higher-order Unified Meta-Environment) is a strongly typed, mostly-functional language with an integrated tool set for developing, proving and assessing concurrent, safety-critical systems. Hume aims to extend the frontiers of language design for resource-limited systems, including real-time embedded and safety-critical systems, by introducing new levels of abstraction and provability.*

http://www.hume-lang.org/
Downloads available for Linux and MacOSX
(RT-Linux, Symbian and ARM coming soon)
St. Andrews and Heriot-Watt Universities

HUME properties

- Reliability:
  - correctness by construction
- Determinism:
  - in scheduling, communication, function, behaviour
- Expressiveness:
  - high-level language features, memory management
- Costability:
  - memory, time and power

HUME architecture



primitive recursive functions
primitive recursive data structures

higher-order non-recursive functions
non-recursive data structures

1st-order non-recursive functions
non-recursive data structures

no functions
non-recursive data structures

Other projects at St. Andrews

- Storage:
  - Secure Location-independent Autonomic Storage Architecture
  - http://asa.dcs.st-andrews.ac.uk/
- Holistic design:
  - Design, Implementation and Adaptation of Sensor Networks through Multi-dimensional Co-design
- Naming, addressing and routing
- Software dependability

Summary

- Resource limited environments and systems
- Edge networks
- Many challenges
- Engineering and architectural issues
- Multi-disciplinary work:
  - engineering science and computer science
- **Collaboration is key** to meet challenges:
  - multi-disciplinary, industry and academia
  - complementary knowledge and skills

St. Andrews, Computer Science

- 5A in 2000 Research Assessment Exercise
- MSc - http://www.dcs.st-andrews.ac.uk/postgrad/msc/
  - Advanced Computer Science
  - Artificial Intelligence
  - Software Engineering
  - Networks and Distributed Systems
- PhD programme
- Networked and Distributed Systems:
  - http://nds.dcs.st-andrews.ac.uk/

Questions?