Software Engineering for Telecommunications Protocols: Applications to Ad Hoc Networking

Michel Barbeau
Carleton University

CITO/OCRI TechTalk

Challenges in Wireless Mobile Ad Hoc Networks
Tuesday, November 20, 2001

Motivation

- Inter-working of ad hoc networks and infrastructure networks
  - Design of new protocol elements
  - Implementation/evaluation

Outline

- Network architecture issues
  - Placement of ad hoc routing
  - Heterogeneous ad hoc nets
  - Internet interconnection using Mobile IP
- Implementation architecture issues
  - Generative Programming
  - Conceptual modeling
  - Feature modeling
  - Metaprogramming

Placement of ad hoc routing

- Link layer
  - Firmware implementation in wireless cards
- Network layer
  - Support of multiple net interfaces/heterogeneous nets
- Both ?

Heterogeneous ad hoc nets

- Why? Flexibility!
- Overlay net
  - A subset of the nodes have a second long-range (lower-speed) wireless net interface
- Long range communication with low number of hops
- Bridge gaps between partitions

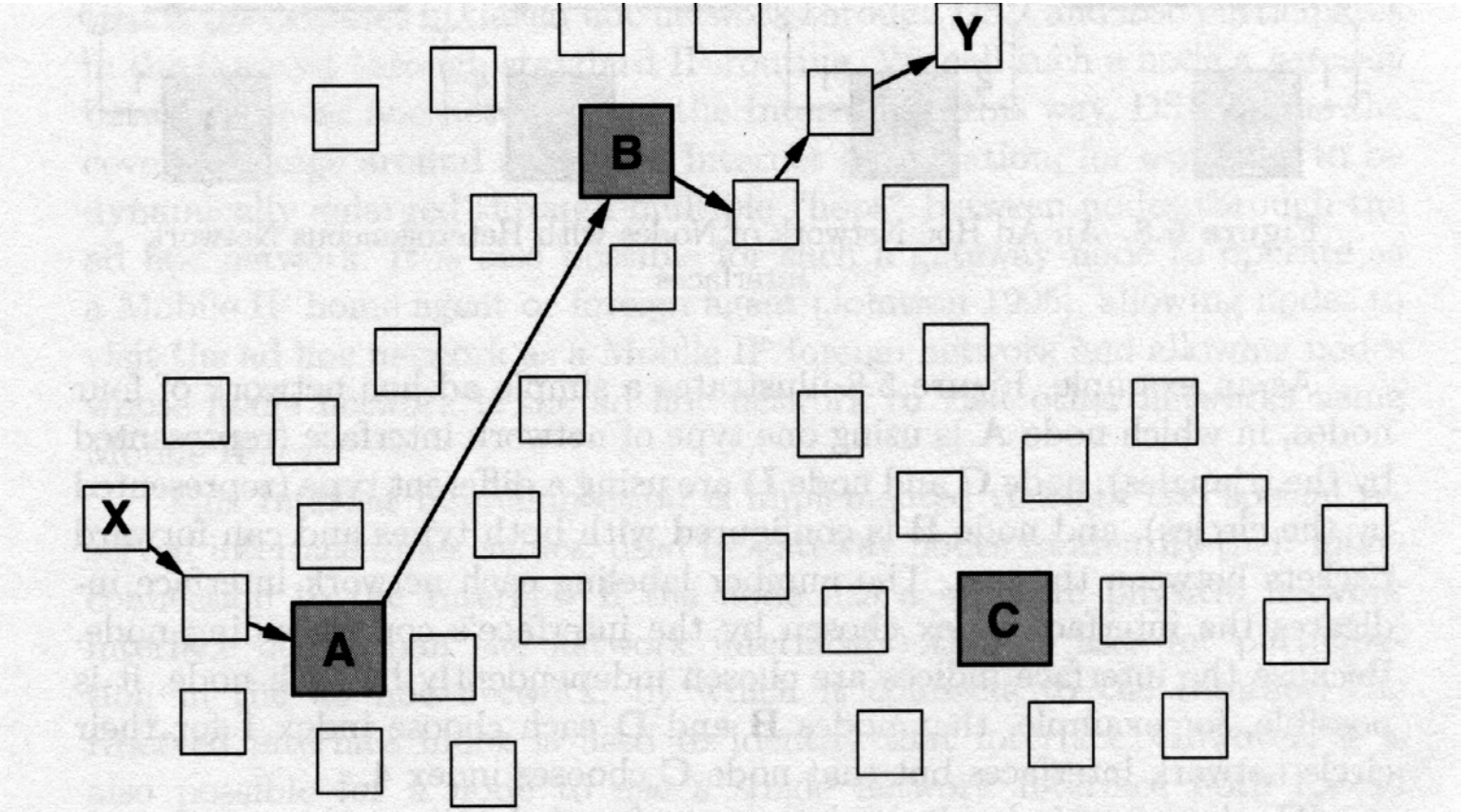


Figure 5.7. A Heterogeneous Ad Hoc Network

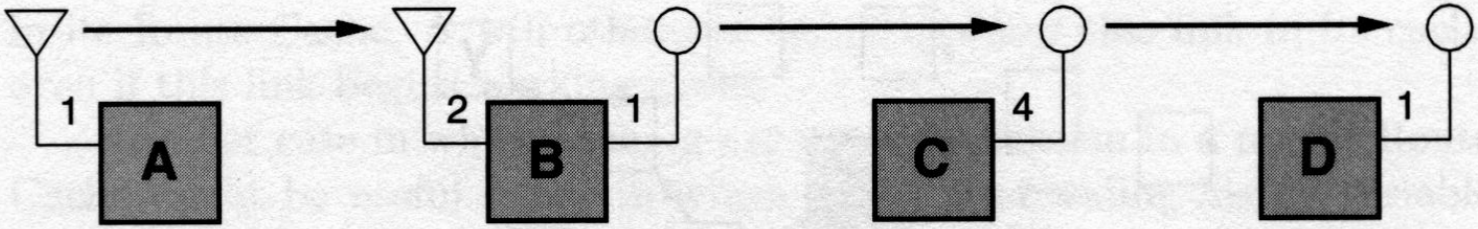


Figure 5.8. An Ad Hoc Network of Nodes with Heterogeneous Network Interfaces

Home address:

id of a node on the ad hoc net

Syntax:

home address/interface index

Source route:

A/1->B/1->C/4->D

Reversed route:

D/1->C/4->B/2->A

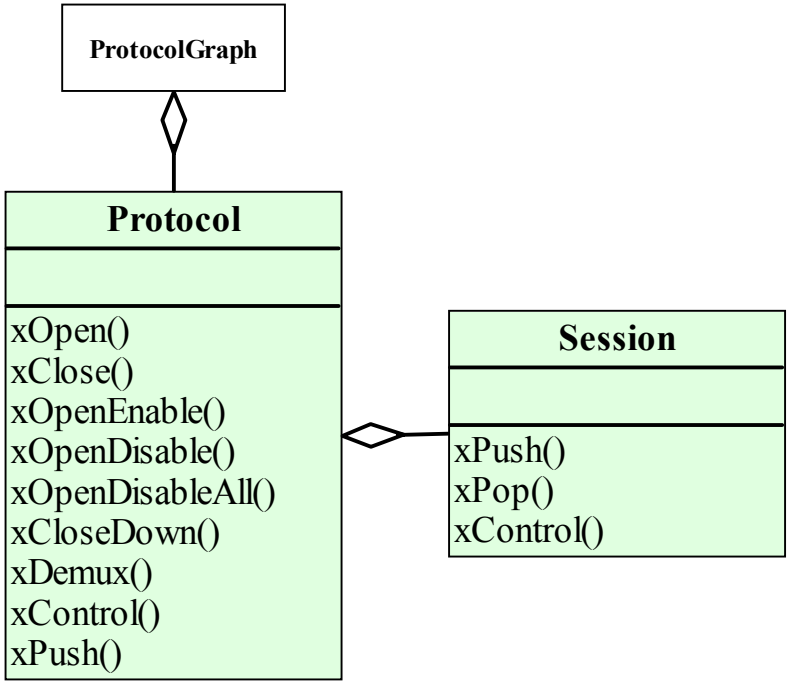
Internet interconnection using Mobile IP

- Gateway
  - Node of the ad hoc net is connected to the Internet
- Models
  1. Participate in ad hoc routing and normal IP routing [Johnson, Maltz, and Broch '99]
  2. Gateway is a foreign agent [Johnson et al. '99]
     - ad hoc net is a foreign net for nodes joining it
     - gateway advertises its existence

Generative Programming

- Domain Engineering
  - Development of base for family of systems
- Steps
  1. Conceptual modelling
  2. Design of a common architecture: collection of components
  3. C++ metaprogramming

Conceptual modeling



- Event
- Map
- Message
- Participant

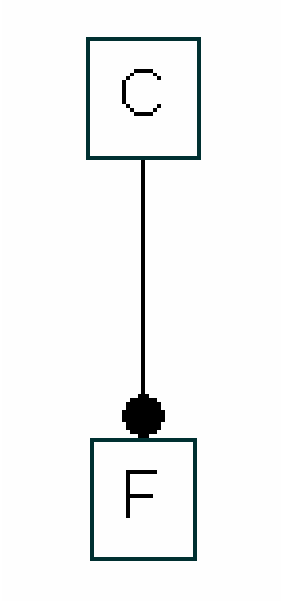
Uniform Interface

Uniform Design

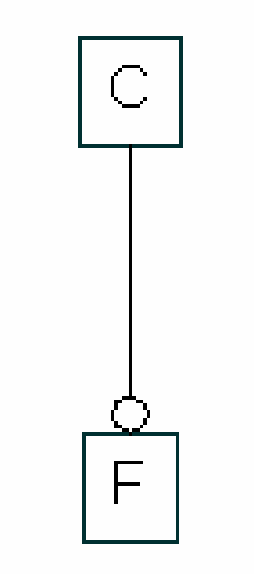
Feature modeling

- Concept
  - Element/structure in applications of the family
- Feature
  - Important property of a concept
  - Point where choices can be made to define a particular instance of a concept
- Feature diagram
  - Representation that captures the features of a concept

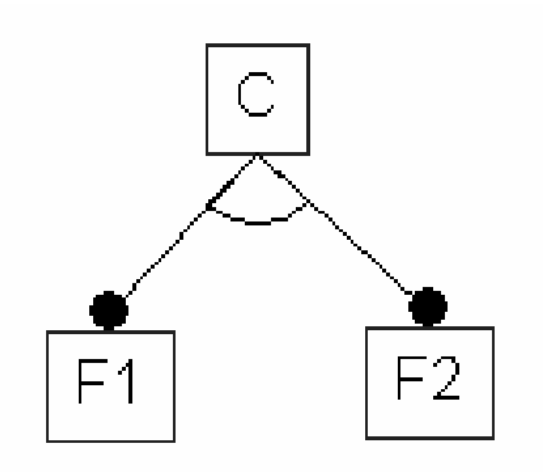
Feature diagram



Mandatory feature

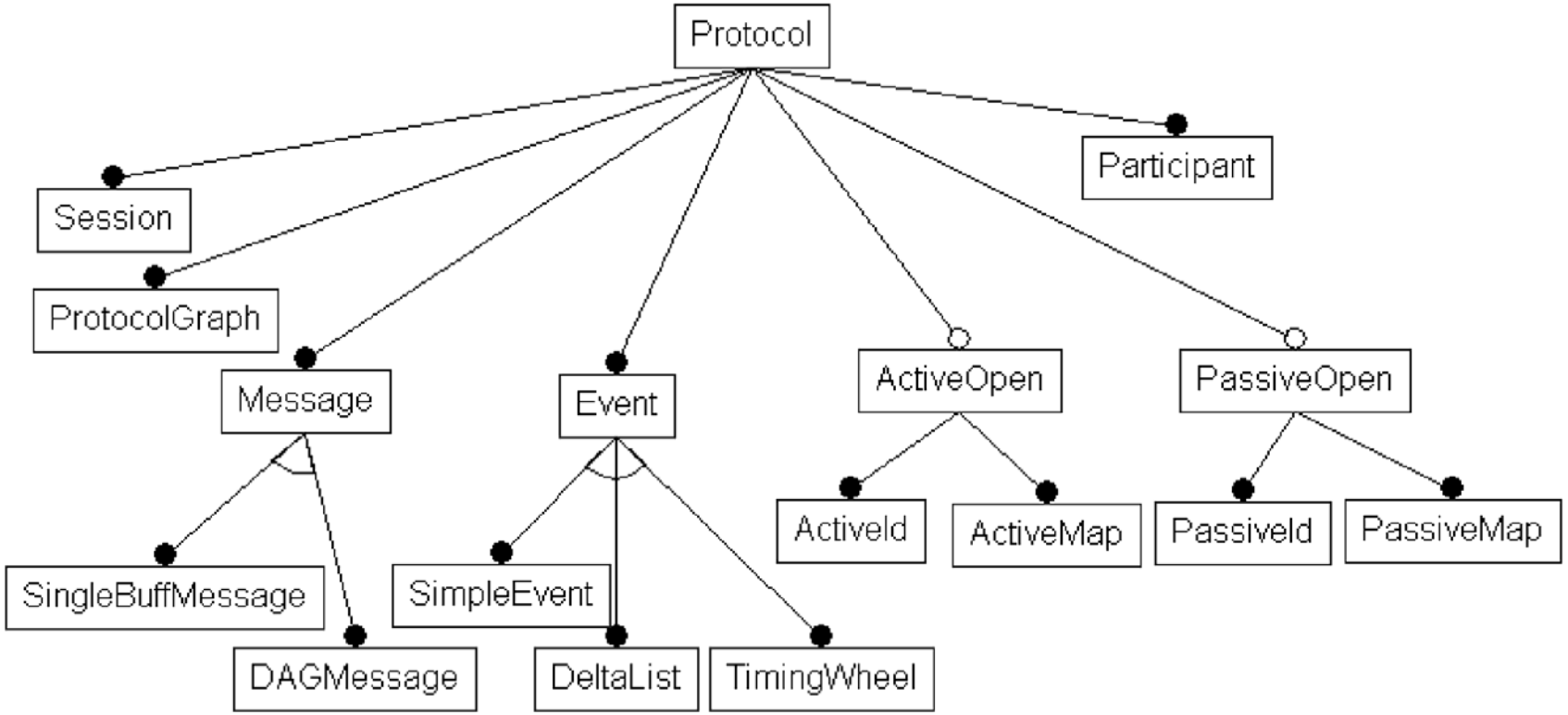


Optional feature



Alternative features

Protocol Concept

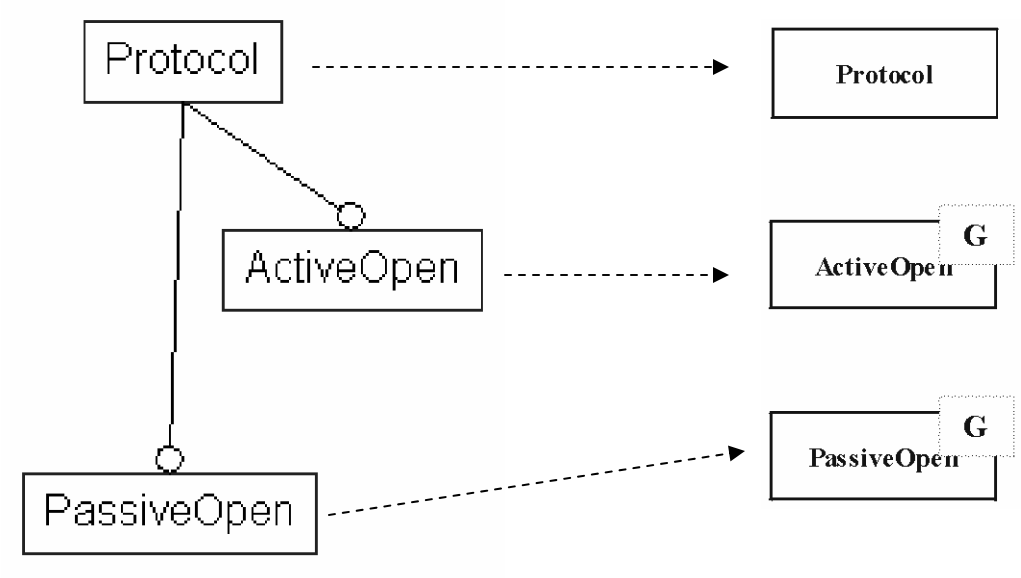


$2 \cdot 3 \cdot 3 = 18+$ variants

What is metaprogramming?

- C++ Programming language
  - if (expression) statement1 else statement2
  - Executed at run time
  - Turing complete!
- C++ Pre-processor operators
  - #if constant-expression … #else … #endif
  - Executed at compile time
  - Not Turing complete!
- Metaprogramming
  - C++ templates: IF<>,…
  - Generic functions/class
  - Executed at compile time
  - Turing complete!

Translation to C++



Generator

```
template
<
   class SessionClass,
   class ParticipantClass = Part,
   class MessageClass = SingleBuffMessage
>
class PROTOCOL_UPI_GENERATOR
{ … }

typedef PROTOCOL_UPI_GENERATOR
< SessionUPI >::RET ProtocolUPI;
```

Work in progress

- Integration/implementation:
  - Mobility support in IPv6
  - Dynamic Source Routing
- Platform
  - SBC
  - Linux
  - Protocol Implementation Framework for Linux (PIX)

References

- Czarnecki amd Eisenecker, *Generative Programming - Methods, Tools, and Applications,* Addison-Wesley, 2000.
- Hutchinson, N.C. and Peterson, L.L., *The x-Kernel: An Architecture for Implementing Network Protocols*, IEEE Transactions on Software Engineering, January, 1991, Vol. 17, No. 1, pp. 64-76.
- Johnson, Maltz, and Broch, *The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, in: Perkins (Ed.), Ad Hoc Networking, Addison-Wesley, 2001, pp. 139-172.
- www.scs.carleton.ca/~barbeau