Wireless Network Simulator (*WiNS*)

Giao Nguyen
University of California at Berkeley

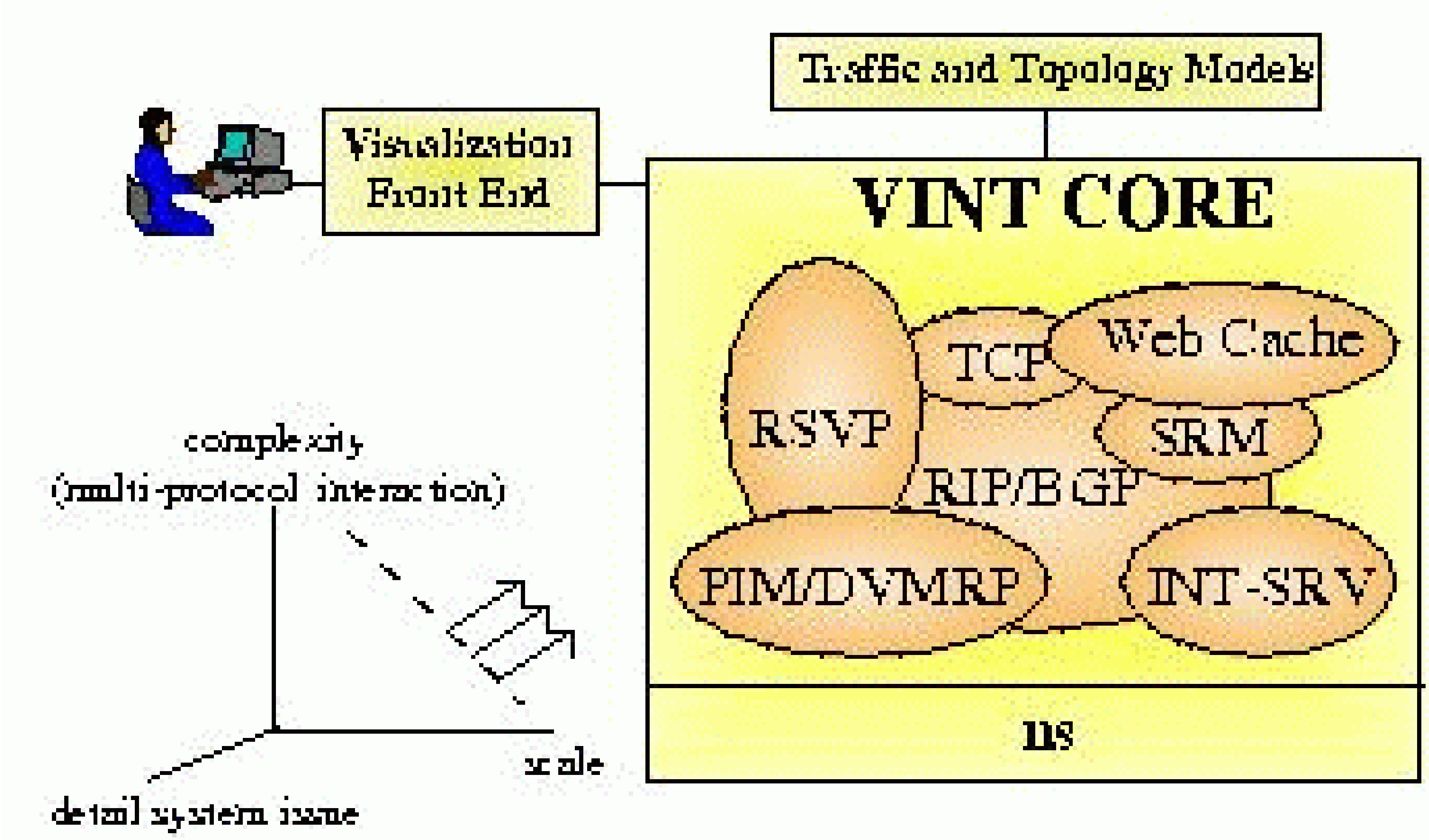
Motivation

- Want to investigate:
  - New protocol designs and algorithms
  - Integrated services, components, and their interaction
  - Scalability of (wireless) network
- Desirable properties of a simulator
  - Easy to extend, robust, and efficient
  - Many integrated components available
  - Large user and developer community

Introduction to *ns*

- Virtual InterNet Testbed simulator
  - Principle developer: Steven McCanne (UCB)
  - Collaboration: USC/ISI, PARC, LBNL, and UCB
  - Active developers: 30 and growing
  - Widespread use in the Internet community
- Key features of *ns*
  - Composable, modular, object oriented framework
  - Graphical tools for animation and configuration
  - Lack support for wireless networks

VINT Project



Existing Components in *ns*

- Application/Traffic: Web, ftp, telnet, CBR
- Transport: TCP, UDP, RTP, SRM
  - TCP: Tahoe, Reno, Vegas, SACK, FACK, and more
- Routing:
  - can do both static and dynamic routing
  - support multicast protocols (DVMRP, PIM)
- Queueing:
  - DropTail, RED, CBQ, SFQ, DRR
- Network Emulation: tap to the real network
- Tracing

Architecture of *ns*

- Two language
  - Split object design
- OTcl
  - Focus on **policy**
  - Compose objects
  - Configure simulations
- C++
  - Focus on **mechanism**
  - Core simulation objects

Split Object

- C++

  bind("variable", &variable)

  NsObject::command(argc,argv)

  o = new NsObject;

  o->variable = value;

  o->method(args);

- Tcl:

  NsObject instvar variable

  NsObject instproc method {args}

  set o [new NsObject]

  $o set $variable $value

  $o method $args

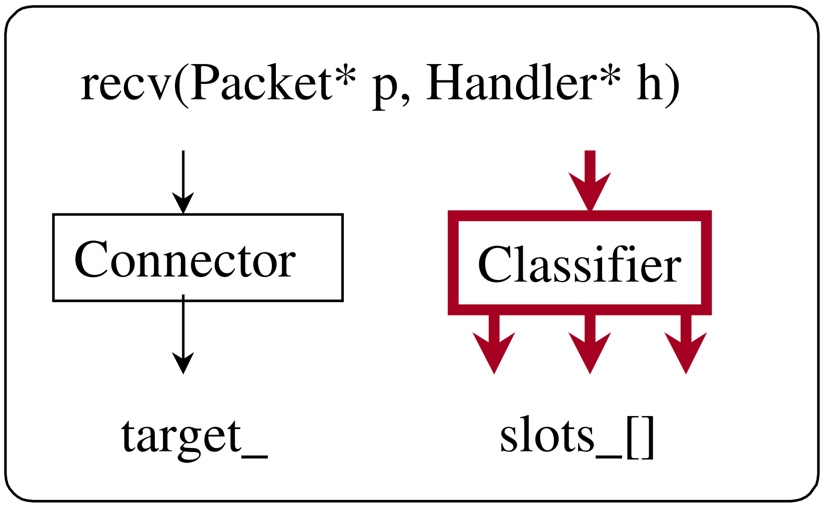
Core Components in *ns*

- Event and Handler
  - Key method: void handle(Event*)
    - handle assynchronous events invoked by the scheduler
  - Every NsObject is a handler
- Packets
  - derived from Event
  - Headers: in hdr_* structure
- Scheduler:

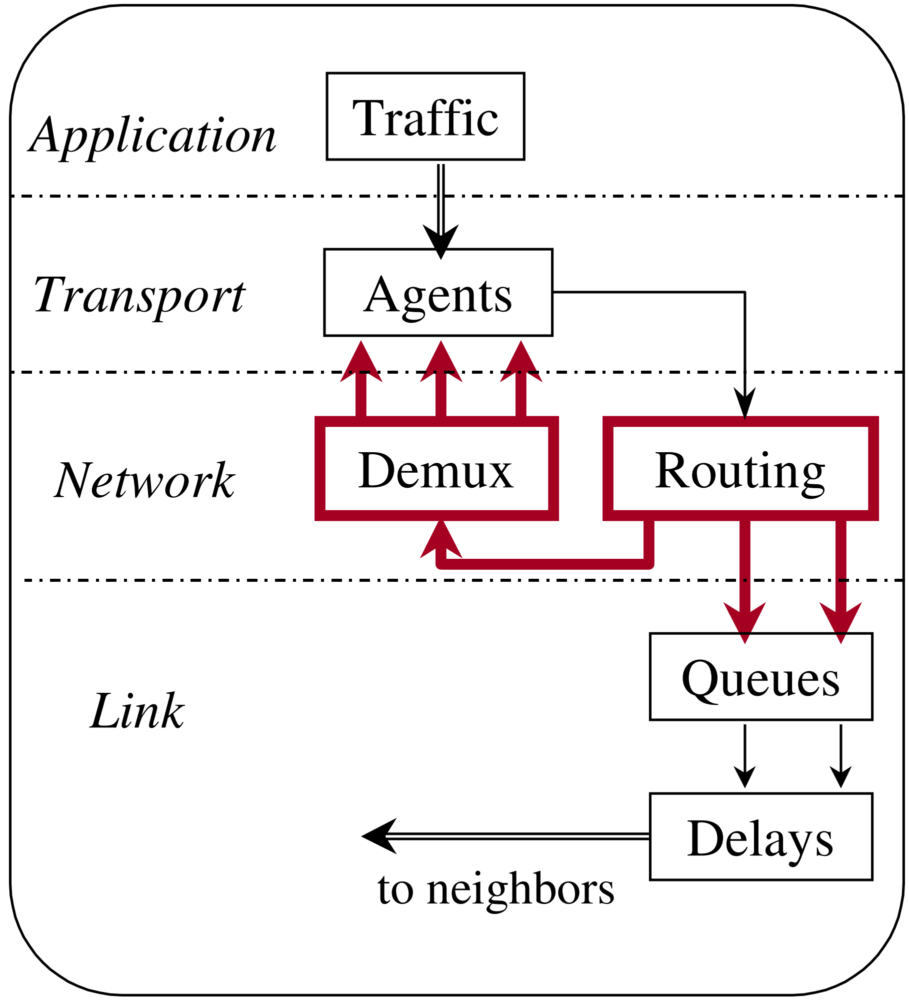
  s = Scheduler::instance();

  s.schedule(h, event, delay);

Packet Handling Components



- Key table for routing
  - link_(src:dst)

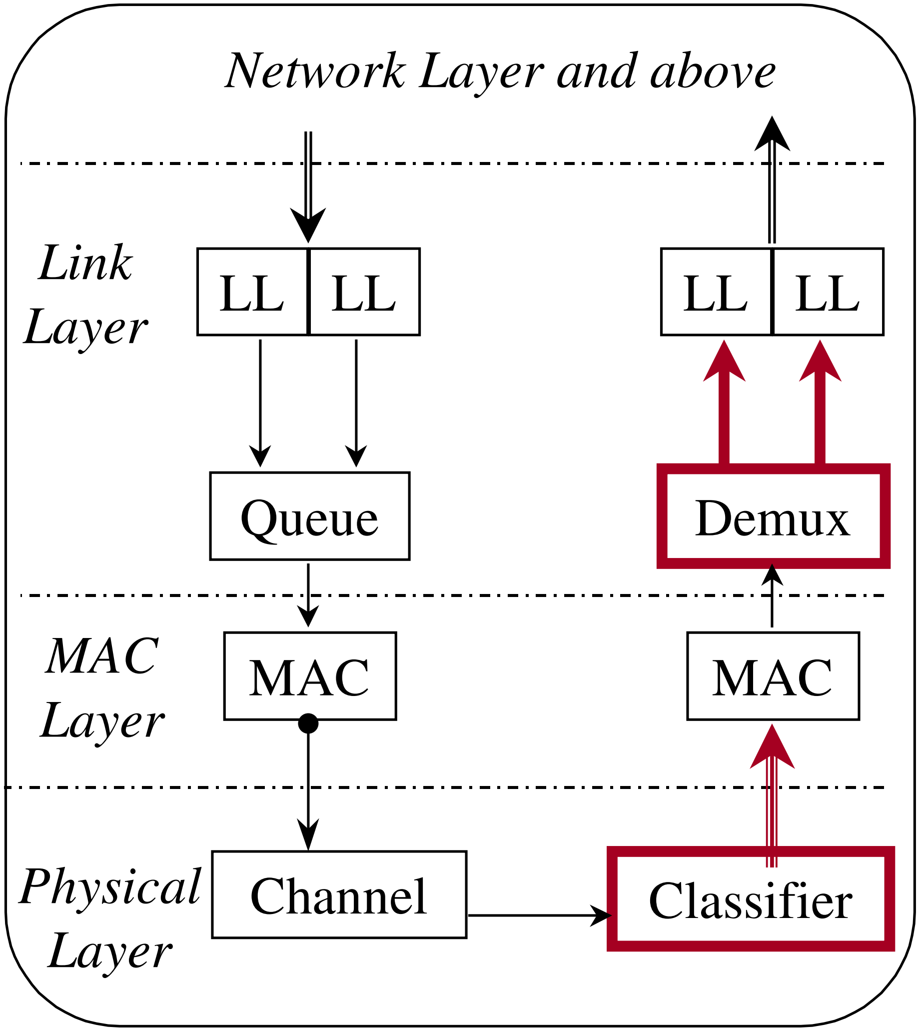


Wireless Network Extensions

NetIface:

  – Extends *Link* component

- *Link Layer* (LL)
- *MAC Layer*
- *Physical Layer*
  – channel: contention
  – MAC classifier: delivers packets to the right MAC



Link Layer

- Functionality
  - Send: queuing
  - Recv: demultiplexing packets
  - Supports peer-to-peer protocols
- Peer-to-peer protocols implemented
  - Basic: no retransmission, no reordering
  - SNOOP protocol: “TCP smart” retransmission

MAC Layer

- Protocols implemented
  - CSMA/CD: Ethernet
  - CSMA/CA: WaveLAN
  - IEEE 802.11: RTS/CTS option
  - Multihop: Metricom Ricochet radio

Channel and Error Models

- Functionality
  - Propagation, carrier sense, collision and detection
  - Supports coexistence of multiple MAC types
    - differences in protocol and bandwidth
- Wireless Error Models
  - Efficient in simulating wireless channel
  - Empirical models based on measurement [WSC96]
  - Replay error characteristics from traces

ErrorModel

- Separate mechanism from policy
  - Mechanism: recv(Packet* pkt, Handler* h)
    - mark the packet if corrupt(pkt) return non-zero;
    - drop the packet if there is a drop target, drop_
  - Policy: corrupt(Packet* pkt)
    - Three types of error unit: packet, byte, or time
- Setup in Tcl

```
set em [new ErrorModel]
$em set rate_ 0.01
$link install-error $em
```

Asymmetric Network Extensions

- Examples of asymmetric networks
  - Cable modems (both wired and wireless)
  - Satellites
- Protocols implemented
  - End-to-end solution
    - ACK congestion control
  - Router-based solutions
    - ACK filtering
    - Priority scheduling of ACKs

Summary

- Key infrastructure to support wireless in *ns*
  - Wireless channel and error models
  - MAC protocols
  - Link Layer protocols
  - Asymmetric network
- Other contributions
  - Link scheduling
  - Tracing of TCP state

Future Work

- Upcoming extensions
  - Satellite and wireless cable modem support
  - Mobile-IP and handoff algorithms
  - Improve visualization of wireless and LAN
- For up-to-date tools and information:

  http://mash.cs.berkeley.edu/ns