**Motivation**

- Emerging networked applications
  - video conferencing, distributed simulations, telepresence...
  - applications want high quality service, where quality is an application and user-specific concept

- Existing networking environments
  - made up of diverse generic resources,
  - need to be shared among cooperating and competing users

- Several problems
  - resource discovery
  - resource management
  - service composition
  - accounting, security, ...

---

**Resource Management Requirements**

- Resource allocation in space
  - allocation of an integrated set of diverse resources that meets a set of user requirements

- Resource management in time
  - runtime management on multiple time scales
  - adaptation to changing conditions and requirements

- Resource management across services
  - control how resources are shared by multiple users
  - support both isolation and cooperative sharing

- Optimization of application-specific quality criteria requires customization along all three dimensions.
  - build on lower-level services and quality guarantees

---

**Darwin Architecture**
**Darwin Application Model: Virtual Networks**

- **Virtual Networks**: A set of resources that are allocated and managed in an integrated fashion.
- **Nodes**:
  - desired services (computation, data processing)
  - input/output type specifications (e.g., MPEG)
- **Flows**:
  - communication between nodes
  - typed
  - quality specifications

**Darwin Components**

- **Resource brokers** design the virtual network (Xena)
  - identify resources that meet application needs
  - perform global domain-specific optimization of quality and cost
- **Delegates** manage the virtual network resources
  - customized runtime resource management (adaptation)
- **A hierarchical scheduler** supports the virtual network properties (H-FSC)
  - isolation and controlled dynamic sharing of resources
  - operates on both individual flows and flow aggregates
- **A signaling protocol** coordinates these three resource management mechanisms (Beagle)
  - distributes and maintains the virtual network state

**Motivation for Components**

**Multiple Time Scales**

- **Coarse**: Resource Broker
- **Medium**: Delegates
- **Fine**: Hierarchical Scheduler

**Scope of Information and Actions**

- **Global**: High-level
- **Restricted**: Detailed
- **Local**: Network-specific

**Customization**

- **Domain-specific**: Global Optimizations
- **Customized**: Policies and Actions
- **Parameters**

**Darwin Architecture with Components**

- **Appl**: Application
- **Library**: Library
- **Comm**: Communication
- **Service**
  - Xena
  - Resource Broker
- **Hierarchical Local RM**
  - Classifier + Scheduler
  - Beagle Delegate

- **Beagle**

- **Classifier + Scheduler**

- **Appl**: Application
- **Library**: Library
- **Comm**: Communication
Resource Brokers

- Brokers identify resources and perform optimizations using global network information and domain knowledge.
  - can be part of application, service provider, or stand alone
  - high-level coordinated requests allow optimizations to happen “under the cover”.
- Brokers implement domain-specific QoS
  - optimize quality, cost using domain specific metrics
  - tailor optimization to application preferences
- Xena: generic and video resources
  - node placement to minimize cost or optimize quality
  - insertion of transcoders to deal with flow type mismatches
  - selection of level of video quality
  - tradeoffs between computation and communication
Runtime Resource Management (Delegates)

- Maintaining high service quality is a continuous activity
  - resource availability changes due to dynamic resource sharing
  - application needs might change
- "Presence" inside the network allows responsive, customized adaptation of resource use
  - application manages "its" resources
  - avoid roundtrip delay to endpoint
- Delegates run at each level of resource management hierarchy
  - manage subtree

Delegate Example: MPEG Frame Dropping

- 2 MPEG video streams compete with UDP traffic.
  - random packet drops result in poor performance.
  - reservations help but traffic bursts still result in high random losses.
- Delegates
  - selective dropping of B frames under congestion doubles frame rate.
  - (custom policy) reduce number of dropped frames for critical streams

Delegates: A Closer Look

- How are delegates realized?
  - Java threads running inside a VM "sandbox"
- What can delegates do?
  - monitor network status
    - queue occupancy, error flags, etc.
  - change the resources allocated to flows
    - adjust resource allocation/sharing rules
    - split and merge flows by adjusting packet classification
    - split can be used for dropping packets
  - affect routing
    - take an alternative path (via a transcoder or another compute node)
  - send and receive messages
    - coordination
- Flow-centric perspective of adaptation
  - adaptation of application end-points is outside the framework

Hierarchical Scheduling (H-FSC)

- Maintains virtual network abstraction
  - isolation between competing users
  - dynamic control over excess BW
  - hierarchy of flows and flow aggregates
- Packet classes identify flows
  - source/dest IPs, masks
  - protocol, port numbers
  - application-specific ID (e.g., MPEG)
- Hierarchical Fair Service Curve
  - independently controls BW, delay
  - determines packet scheduling policy
    - when is a packet sent on the link

Beagle Delegate Hierarchical Local RM Classifier + Scheduler
Delegate-Scheduler Handshake

Class C1schd {

    // interface with the packet scheduler
    retrieve -- get current hierarchy
    add -- add node in scheduler hierarchy
    del -- change node parameters

    // interface with packet classifier
    dsc_on -- discard class packets
    dsc_off

    // notification
    probe
    reqMonitor -- asynchronous
}

Low-level Resource Management (Beagle)

- Responsible for
  - setting up local resource managers
    - resource trees (similar to packet-class trees)
    - handles heterogeneity
    - convert flowspecs into low-level scheduler parameters
      - e.g., service curve for H-FSC scheduler
  - downloading delegates
    - resource requirements (CPU, memory, storage)
    - runtime environment: code type (Java) + runtime type (JDK)
    - list of flows

- Optimizations
  - temporal sharing of physical resources between flows

- Details in the paper

Adaptation in Darwin

- Application-oblivious
  - Hierarchical resource managers
    - adaptive packet scheduling

- Application-aware
  - Resource brokers
    - application specifies nodes, flows, flowspec + objective
    - Xena
      - allocates resources
      - inserts transcenders
      - chooses flow quality (for video flows: type known to Xena)
  - Delegates
    - flow-centric (end-point adaptation is outside framework)
    - change flow parameters, reroute flows, etc.

Summary

- Customizable resource management based on concept of a “virtual network”.
  - global domain-specific optimizations using brokers
  - delegates support responsive customize runtime resource management
  - hierarchical scheduling supports isolation and controlled resource sharing for flows and flow aggregates
  - signaling allocates resources and installs delegates based on high level resource specifications
  - customization supports application-specific QoS

- Complements Active Harmony and CC Adaptation, which focus on end-point adaptation