Diversified Internet Substrate Architecture

- Objectives and design principles
- Data plane considerations
- Control plane
  - » metalink configuration
  - » multi-domain substrate
  - » substrate control communication
  - » security and mobility

Washington University in St. Louis

Role of Substrate

- Substrate provides resources to metanets
  - » provide processing resources for *metarouters*
  - » implement *metalinks* – both point-to-point and multipoint
  - » not intended to provide end-to-end packet delivery
- Support metanet backbone configuration
  - » long-term reservations, generally coarse granularity
  - » support for advance planning
- Access metalink provisioning
  - » on-demand (typically when host boots, or re-connects)
  - » mechanisms that enable metanets to provide mobility
- Multiple substrate domains
  - » multi-domain metanets, inter-domain metalink routing
  - » different trust levels for different substrate domains
- Substrate Control Metanet (SCM)
  - » control messages for substrate configuration
  - » support service advertisements by substrate domains, metanets

Design Principles

- Support varied roles of participants
  - » end users, substrate network providers, metanet providers
- Interoperability among substrate domains
  - » distributed management, but coherent operation
- Architectural neutrality
  - » avoid putting limits on metanets, don't play favorites
- Design for security
  - » enable secure metanets, but don't impose requirements on all
- Design for mobility
  - » enable metanets that support mobility, but don't dictate how
- Adaptability to lower level network technologies
  - » exploit the full range of current and future link technologies
  - » enable wide range of substrate routers implementations
- Expose lower level technology to metanets
  - » so that metanets can take full advantage of underlying links, metarouter implementation technologies

Metanet Abstractions

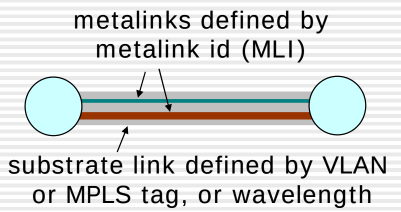
- *Metalinks* – abstraction of a physical link
  - » point-to-point metalinks
    - – may be provisioned, but need not be
    - – may be implemented over point-to-point substrate links, or multipoint substrate links
  - » multipoint metalinks
    - – allow metarouters to use broadcast features of LAN technologies
- *Metarouters* – abstraction of a multiport network device
  - » implemented using generic processing resource within substrate router
  - » may include multiple processing engines
- *Metaterminal* – abstraction of an end system
  - » metanet-specific protocol stack
  - » each metaterm in same physical end system has own metalink

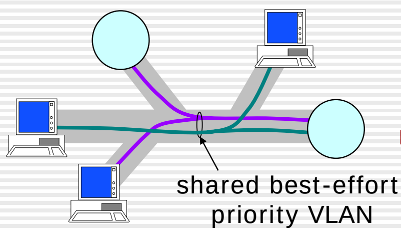
Secondary Abstractions

- *Peering metalink*
  - » point-to-point metalink joining two metanets operated by different organizations
  - » peering subnets may have same or different protocols/services
  - » up to metanets to address interoperability issues
- *Gateway*
  - » connection to a network outside the diversified internet (e.g., the IPv4 Internet)
  - » up to metanet to address interoperability issues
  - » substrate may monitor traffic for policy reasons
- *Meta-transport layer*
  - » allow metanets to easily reallocate transport level capacity
  - » may be implemented by TDM or optical cross-connects

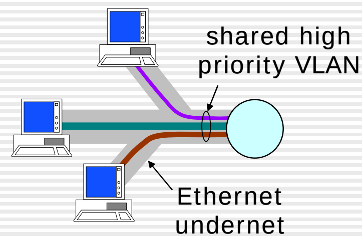
Metalink Implementation

- Pnt2pnt metalinks over pnt2pnt substrate link





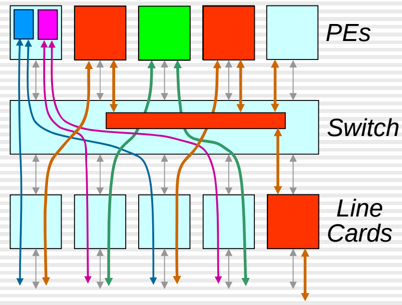
- Pnt2pnt metalinks on multi-access substrate link
  - » high priority VLAN for provisioned substrate link
  - » substrate router limits usage



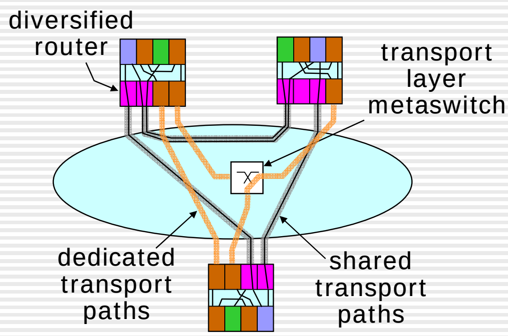
- Best-effort multipoint metalinks
  - » so metanets can use broadcast LAN features

Implementing Metarouters

- Processing Engines (PEs) implement metarouters
- Line Cards terminate external links, mux/demux metalinks
- Shared PEs include substrate component
- Dedicated PEs need not include substrate component
  - » use switch and Line Cards for protection and isolation
- PEs in larger metarouters linked by metaswitch
- Larger metarouters may use dedicated Line Cards
  - » allows metanet to define transmission format/framing
  - » configured by lower-level transport network

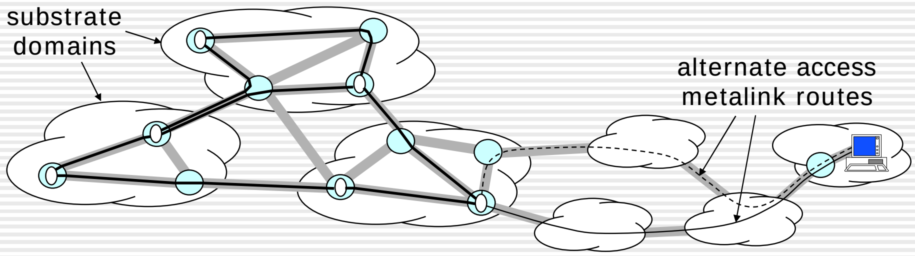


Flexible Transport Layer



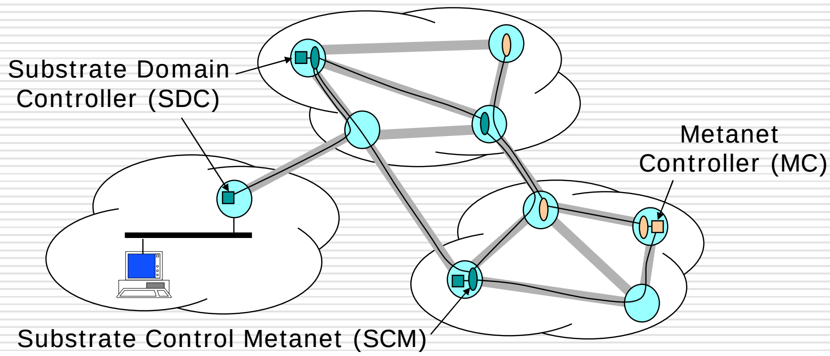
- Shared transport paths carry multiple metalinks
  - » semi-static transport level configuration
  - » dynamic configuration of metalinks within path
- Dedicated transport paths serve larger metanets
- Transport layer metaswitches give metanets more direct control over transport layer resources
  - » allows metanet to shift capacity as traffic changes
  - » may be implemented using a shared cross-connect
  - » potentially highly dynamic

Metanet Configuration



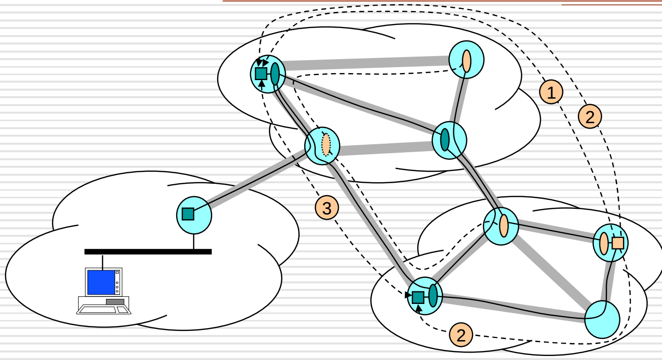
- Metanet backbone provisioning
  - » substrates advertise resource availability, cost information
  - » metanet planner requests bids for metanet segments
  - » iterate, as needed
- Access metalink configuration
  - » users may request connection from anywhere, at anytime
  - » metanet determines termination point, domain-level route
  - » substrate domains determine route segments

Substrate Control Communication



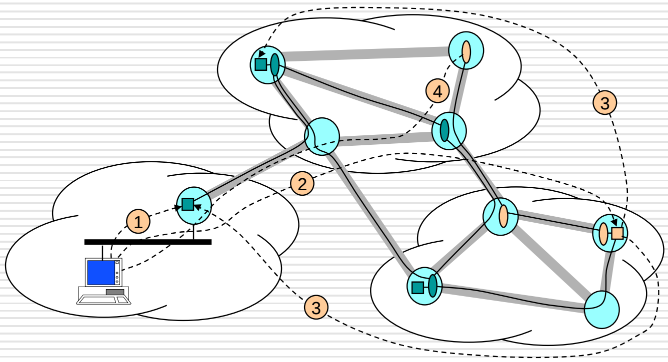
- SCM for control communication outside metanets
  - » may have more than one for reliability, upgradability
- SDCs provide control interface to substrate domains
- MCs provide control interface to metanets
- Substrate domains may also have private SCM

Configuring Metanet



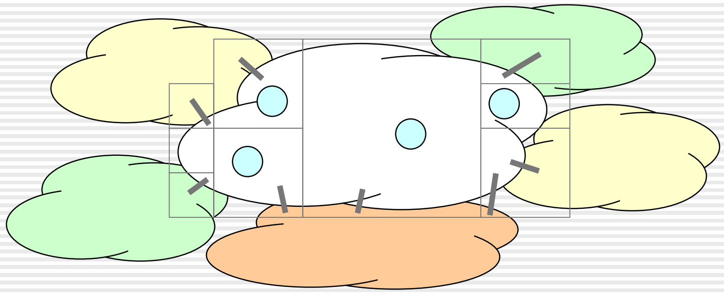
- Adding metarouter and metalinks
  1. MC requests new metarouter & intra-domain metalink
     » configures metarouter within metanet
  2. MC requests inter-domain metalink
  3. peering domains coordinate metalink configuration

Configuring Access Metalink



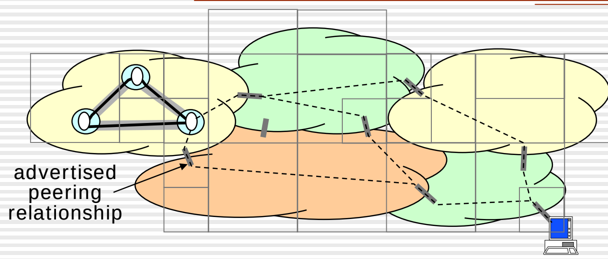
- When host connects to network
  1. discover local substrate router (returns MAC address)
  2. send metanet connect request to local SDC
     » request forwarded through SCM to MC for desired metanet
  3. MC requests metalink configuration from SDCs
  4. SDCs configure access metalink

Substrate Advertisements



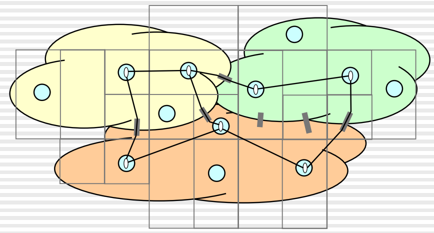
- Substrates advertise so that metanets can use them
  - » hosting capabilities advertisements
    - – in region *R*, can host metarouters on type *T* substrate platforms
    - – multi-scale region specifications
  - » distance advertisements
    - – distance from R_1 to R_2 within substrate is *D*
  - » peering advertisements
    - – D_1 peers with D_2 in region *R*, with capacity *C*
  - » may also specify *scope* to indicate “relevance region”

Metalink Routing



- Metanet uses peering advertisements to identify paths
  - » geographic information used to estimate distances
    - – vertices of path are region center points
  - » for substrates that supply internal region graph, use distances implied by region graph
- Metanet requests route segments from substrate nets
  - » request to domain *D*: metalink *L*, from D_1 in R_1 to D_2 in R_2
  - » request may include a provisioned capacity
  - » adjacent substrate domains use metalink identifier (*L*) to coordinate across domain boundary

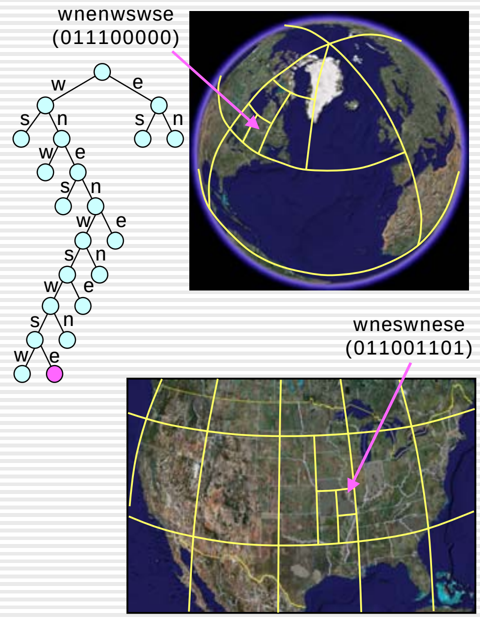
Metanet Backbone Configuration



- Inputs to metanet planner
  - » substrate domain adverts
  - » expected users within regions
  - » expected traffic among regions
- Planner
  - » selects regions for metarouters
    - – typically driven by users in region
    - – may also include transit metrouters
  - » selects metanet topology
    - – determination of metalink capacities
    - – peering points for inter-domain metalinks
  - » determines metarouter configurations
    - – number and capacity of interfaces
    - – number and type of PEs
- Metanet negotiates subnets with substrate domains
  - » potentially iterating over different options

Region Encoding

- Quad-tree on globe
  - » alternate east/west, with north/south
- Trie for regions of interest
  - » quickly find regions containing given point
- Smallest region (equator)
  - » with 32 bits, .6×.3 km
  - » with 40 bits, 2×1 m
- Easy conversion from latitude/longitude



Security Issues

- Enable secure metanets, but minimize substrate role
  - » enable continuing evolution of security mechanisms
- Oversight by *Diversified Internet Governing Authority*
  - » DIGA controls *Top-level Certificate Authority*
  - » TCA provides identifiers for SCMs, substrate domains and metanets; accredits *trusted substrate domains* and SCMs
  - » trusted entities must meet standards or lose accreditation
- Securing metanets
  - » configure metarouters only on trusted substrate domains
  - » protect metarouter-to-metarouter metalinks using encryption
    - – prevents eavesdropping, traffic insertion
    - – can detect lost packets and hold substrate provider accountable
  - » protect access metalinks from misuse
- Substrate Control Metanets
  - » must provide high level of security
  - » protect SCM clients from DoS attacks
    - – disallow source address spoofing and use fair queueing

Mobility in Diversified Internet

- Minimize role of substrate
  - » give maximum control to metanet
  - » substrate supports dynamic access metalink establishment
    - – both at startup and as needed to handle mobile hosts
    - – metalink creation controlled by metanet
- Typical handoff scenarios
  - » 802.11, moving between access points on same subnet
    - – no change visible to substrate router or metanet
  - » moving between access points on different subnets
    - – mobile host informs metanet of impending handoff
    - – metanet requests new access metalink from substrate (may terminate on same or different metarouter)
- Implications and options
  - » requires metanets hosting mobile hosts to have metarouters near host location for seamless handoffs
  - » could support access metalink “grafting” to reduce need for nearby metarouters

SCM Design Elements

- Provided services
  - » unicast, best-effort datagram delivery
  - » advert distribution service
    - –senders may restrict delivery to region, recipient type
    - –receivers may subscribe by advert type, source domain, metanet
- Address, routing forwarding
  - » 64 bit address for all meta-interfaces of all metarouters
    - –hierarchical address assignment to reduce routing state
    - –initial prefix specifies region containing metarouter
  - » point-to-point access metalinks
    - –endpoints use address of metarouter interface – no spoofing
  - » advertise address prefixes with region specifications
    - –restrict scope to control distribution
  - » support forwarding to domain+service or metanet+service
- Security
  - » metarouters only on trusted substrate domains
  - » link level encryption (optional on access)
  - » fair queueing and on-demand blocking filter insertion

Substrate Addressing

- Each substrate domain assigns a 64 bit address to all physical interfaces of substrate routers
  - » initial prefix specifies region containing substrate router
  - » assignment of remaining bits is at discretion domain admin
  - » scope of addresses limited to substrate domain
- Addresses are mostly private to substrate domain
  - » exception for peering links
  - » peering domains exchange addresses assigned to peering links
  - » for control purposes, links are identified by unordered pair $\{domain_1{:}address_1, domain_2{:}address_2\}$
  - » peering links must be point-to-point
- Metanets need not be aware of substrate addresses
  - » know metarouter region, but not hosting substrate router
  - » metalinks specified by metarouter+meta-interface of endpoints
    - – sufficient for metarouter and hosting substrate domain to associate meta-interface numbers with same PE in metarouter

Substrate Configuration

- Substrates advertise using SCM
  - » hosting capabilities and peering connections
  - » configuration service – using region spec to limit scope
- Metanets register to receive adverts
- Metanets request resources by contacting config service
  - » XML-based interface over TCP-like transport
  - » request transit metalink by specifying
    - – ingress peering domain+region, egress peering domain+region
    - – guaranteed bandwidth, max bandwidth, max latency
    - – similar specification for other metalinks
  - » request metarouter by specifying MR#, desired region plus
    - – meta-interfaces
      - MI# and direction
      - PE that terminates MI
      - max bandwidth
    - – Processing Engines
      - PE# and type
      - meta-switch interface bandwidth
  - » support advance reservation

Top-Level Certificate Authority

- Controlled by DIGA
- Hosted on servers reachable through SCM
  - » advertises availability using SCM advert distribution
  - » has well-known public key for secure interaction
  - » stores (identifier, public key) pairs for accredited entities
    - – trusted substrate domains
    - – registered metanetworks
    - – second-level certificate authorities
  - » revoke accreditation by deletion (timestamp query responses)
- Second-level certificate authorities
  - » must mirror top level data
  - » may provide additional services
    - – identifiers and public keys for endpoints, unaccredited substrates
    - – certification of lower level authorities