

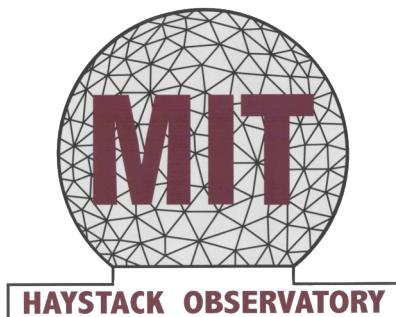
Protocols for Real-Time e-VLBI Data Transport

Chester Rusczyk

chester@haystack.mit.edu

Alan Whitney

awhitney@haystack.mit.edu



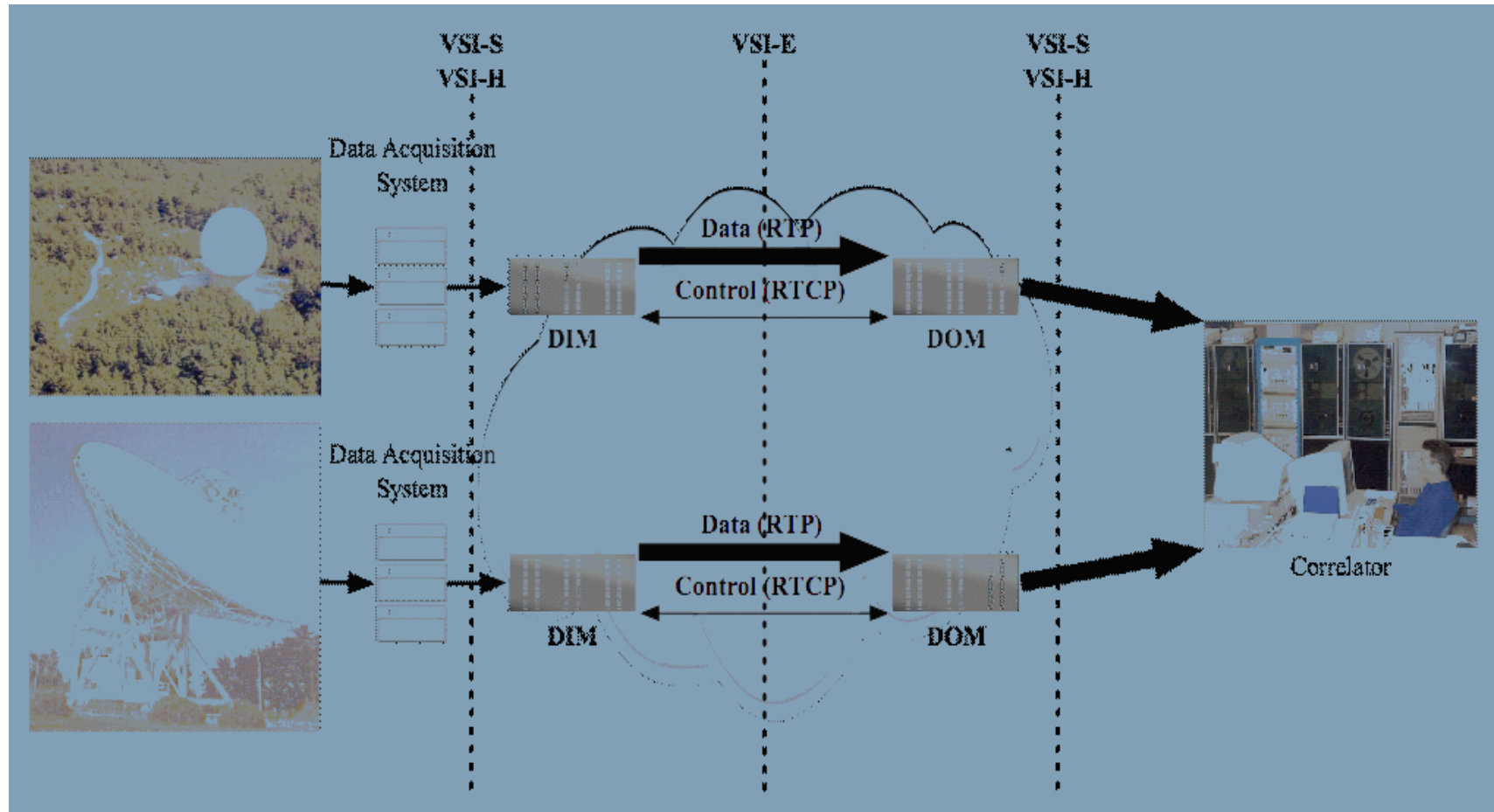
4th e-VLBI Workshop, Sydney, Australia
Tuesday July 12, 2005



Outline

- Summary of VSI-E and EGAE Status
- DRAGON Project Introduction
- Impact on e-VLBI
- E-VLBI / DRAGON Interface
- Conclusion

VLBI Standard Interface electronic (VSI-E)



VSI-E Status

- VSI-E testing phase
- VTP-1.0.H released in 2004
 - Libraries and application
- Transport protocol to use.
 - RTP agnostic to underlying transport protocol
 - UDP, TCP, UDT, Fast TCP, other experimental.
 - Protocol Evaluation
- Protocol Tuning
 - (e.g. TCP –Transmit buffer size, MTU, etc.)
- Minimize network setup / configuration

Experiment Guided Adaptive Endpoint (EGAE)

- Use e-VLBI data characteristics
- Adaptively adjust the loading on infrastructure
 - Network protocols to network conditions
- Transmits data over IP networks
 - Uses a variety of adaptation strategies
 - Best use of the available bandwidth.
- Primary mode of adaptation
 - Use scavenged bandwidth when possible

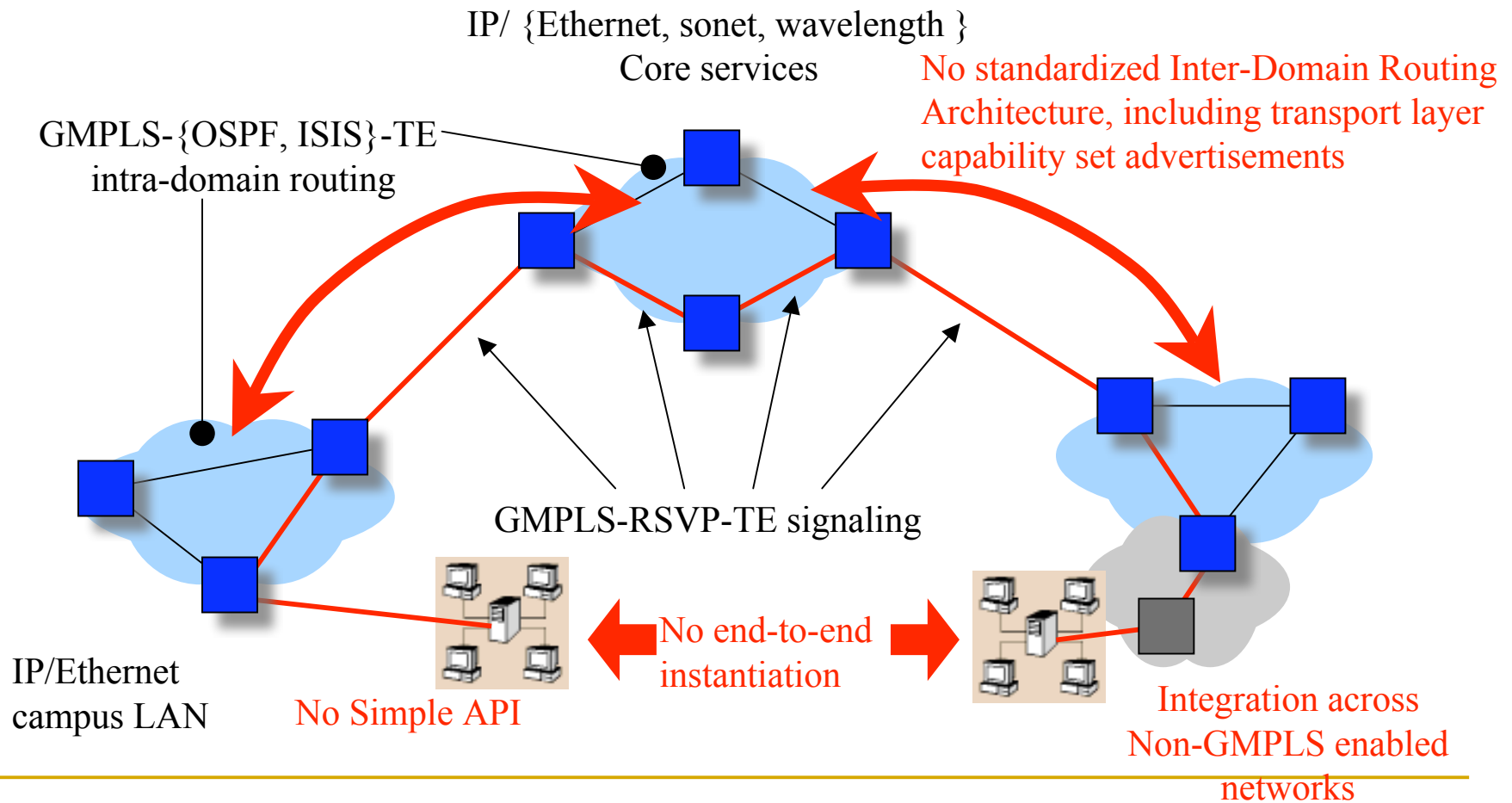
EGAE Status

- Used in K5 to Mark 5 Disk to Disk transfers
 - Tsukuba to Haystack
 - Avg. 150 Mbps transfer rates
 - Application Limitations
 - Utilize bbftp – (TCP)
 - Large number of files
 - File sizes
 - Profiles transferred using ftp to stations
- Adding centralized control
 - Configuration / Management

e-VLBI Real-Time Data Transport

- Stations are endpoints
 - Separated by large distances
 - Network Issues
 - Multiple Backbones
 - Routes
 - E2E Performance varies
 - Last mile
 - VSI-E, EGAE
 - VSI-E – Uses large spare capacity
 - EGAE – Less than best effort
-

End-to-End GMPLS Transport



DRAGON Project Objectives / Key Features

- Uses all optical transport in the metro core
 - Edge to edge Wavelength switching
 - OEO only for signal integrity
 - Push OEO demarcation to the edge
 - Towards end user
- Standardized GMPLS protocols to dynamically provision intra-domain connections
 - GMPLS-OSPF-TE and GMPLS-RSVP-TE

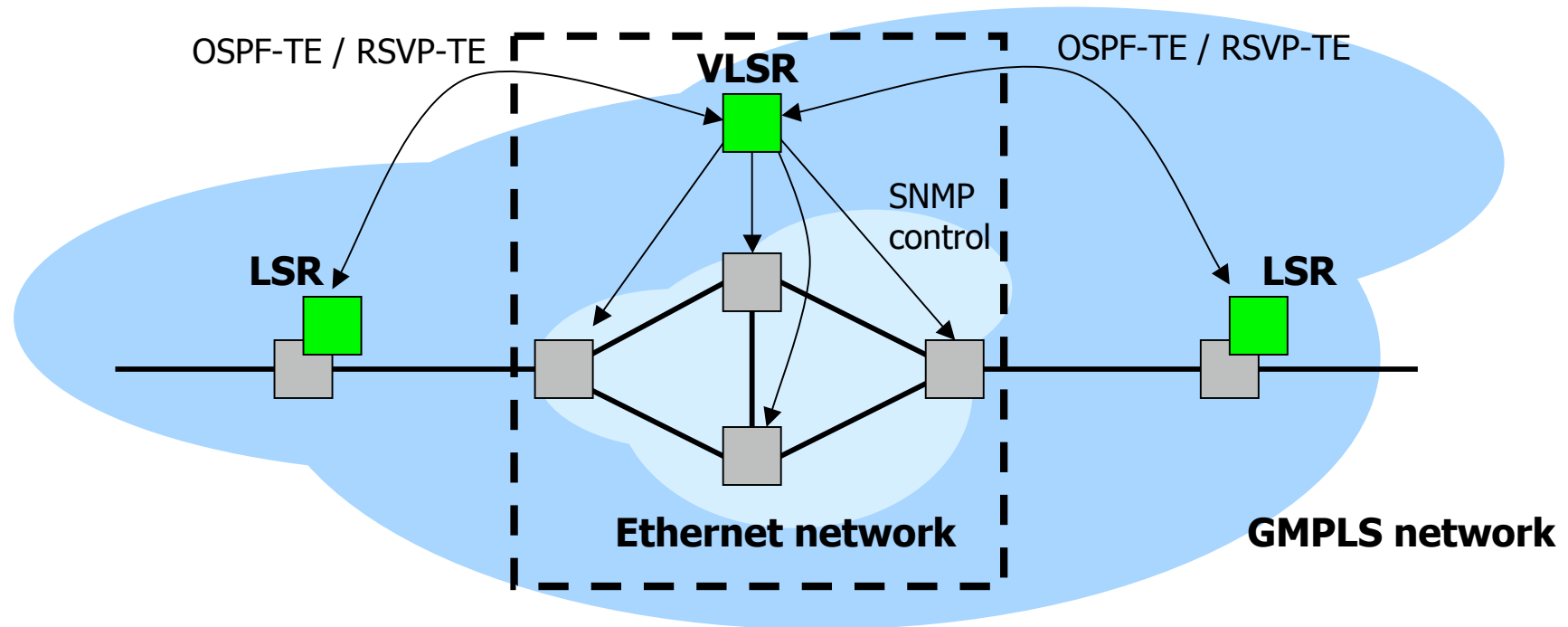
DRAGON Project Objectives / Key Features

- Develop the inter-domain protocol platform
 - Distribute Transport Layer Capability Sets (TLCS) across multiple domains
 - Perform E2E path computation
 - Resource authorization, scheduling, and accounting
 - Develop the “Virtual LSR”
 - Abstracts non-GMPLS network resources into a GMPLS “virtual LSR”.
 - Simplified API
 - Application Specific Topology definition and instantiation
 - Resource resolution, proxy registration and signaling
-

Virtual Label Switched Router (VLSR)

- Many networks consist of switching components that do not speak GMPLS, e.g. current ethernet switches, fiber switches, etc
- Contiguous sets of such components can be abstracted into a Virtual Label Switched Router
 - A management agent (the VLSR) can be created that interacts with the DRAGON network via GMPLS protocols
 - The VLSR translates GMPLS protocol events into a generic pseudo-commands for the covered switches. The pseudo- commands are tailored to each specific vendor/architecture using SNMP, TL1, CLI, or a similar protocol.
 - The VLSR can abstract and present a non-trivial internal topology as a “black box” to an external peering entity.

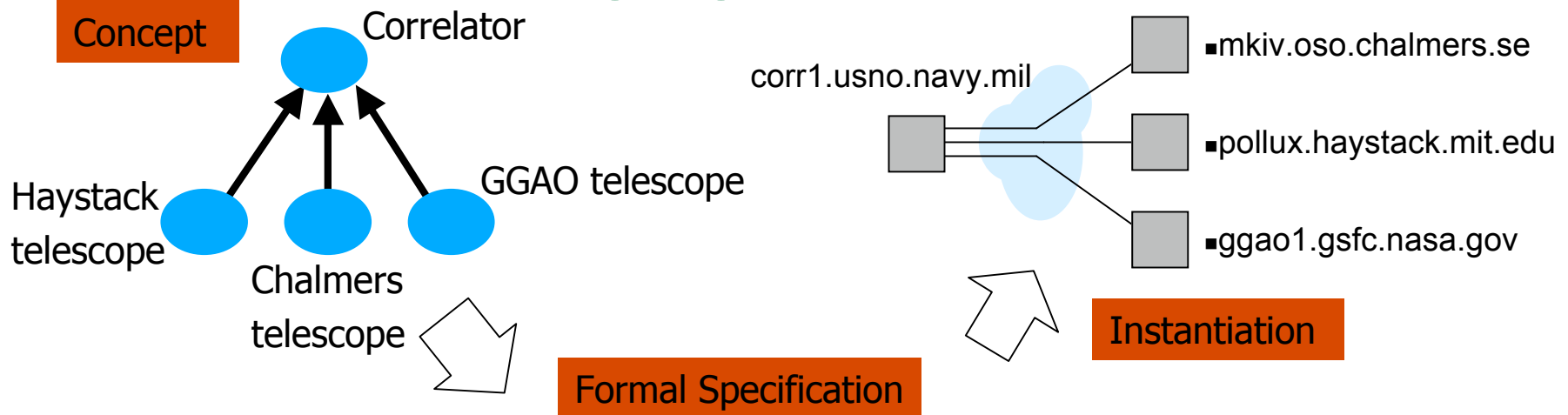
VLSR Abstraction



Application Specific Topologies

- A formalized definition language to describe and instantiate complex topologies
 - Application topologies consist of multiple LSPs that must be instantiated as a whole.
 - Resources must be reserve-able in advance for utilization at some later time
 - By formally defining the application's network requirements
 - Service validation and performance verification can be performed (“wizard gap” issues)

Application Specific Topology Description Language - ASTDL

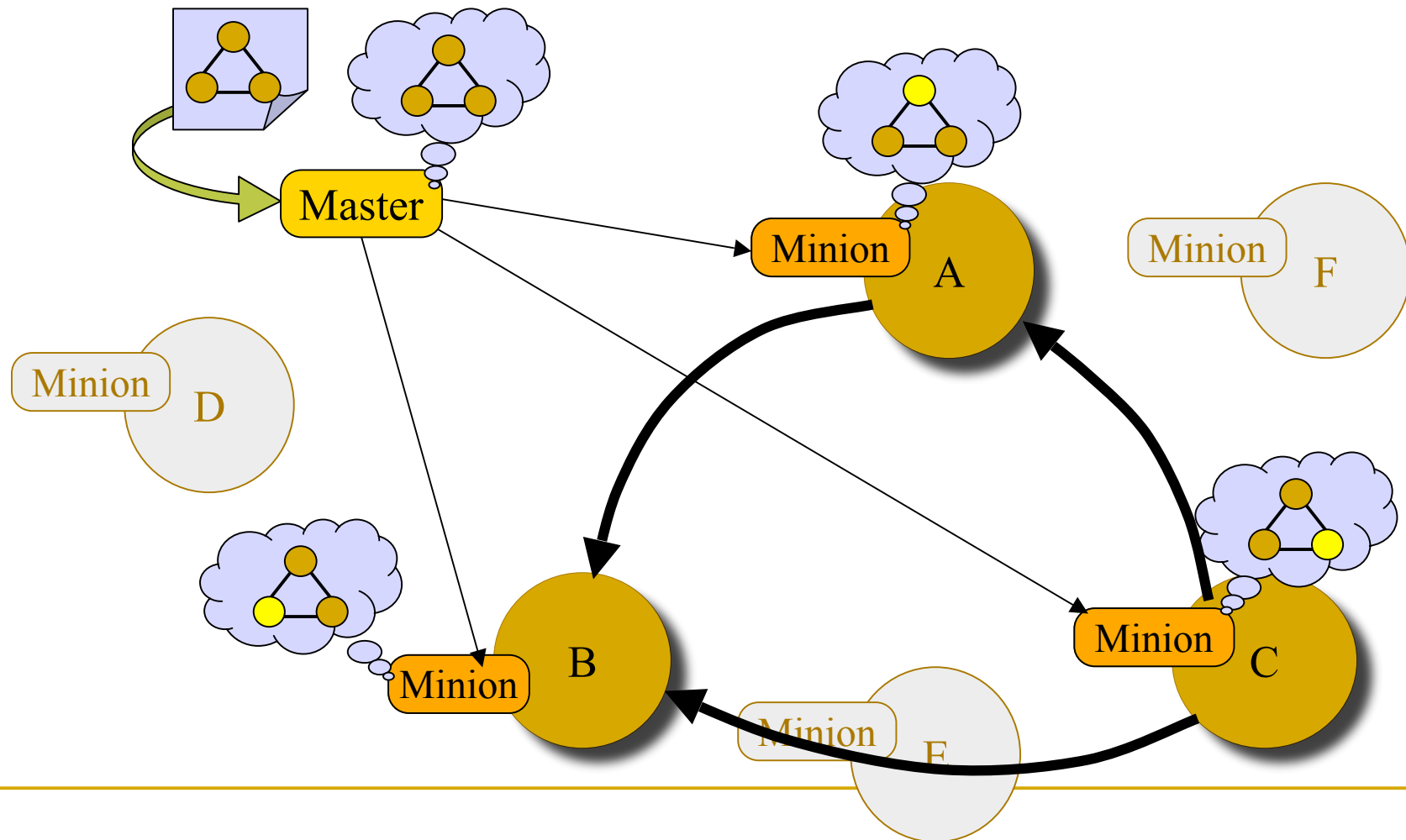


```
Datalink:= { Type=Ethernet; bandwidth=1g;
              SourceAddress=%1::vlbid;   DestinationAddress=%2; }
Topo_vlbi_200406 := {
    Correlator:=corr1.usno.navy.mil::vlbid;           // USNO
    DataLink( mkiv.oso.chalmers.se, Correlator );     // OSO Sweden
    DataLink( pollux.haystack.mit.edu, Correlator );  // MIT Haystack
    DataLink( ggao1.gsfc.nasa.gov, Correlator );     // NASA Goddard
}
```

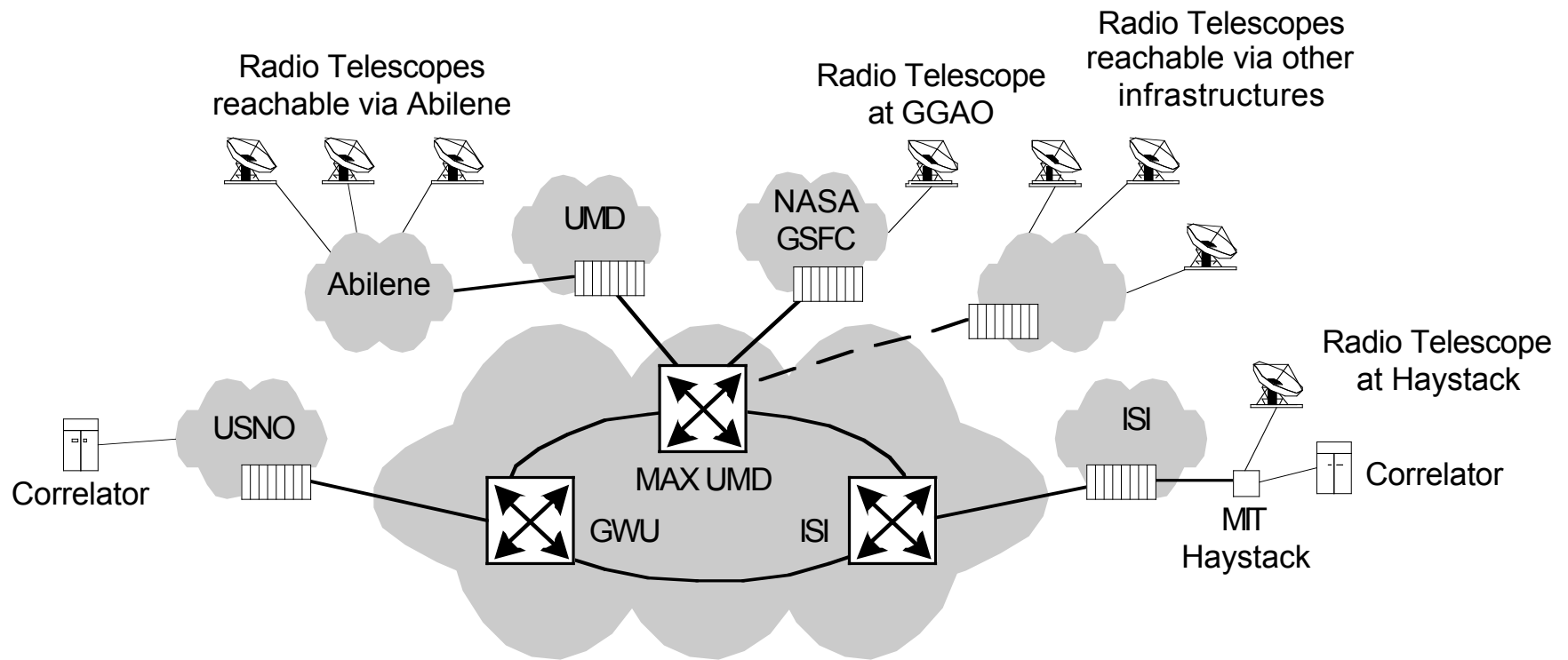
C++ Code invocation example:

```
eVLBI = new ASTDL::Topo( "Topo_vlbi_200406"); // Get the topology definition
Stat = eVLBI.Create();                       // Make it so!
```

The AST Process



E-VLBI Experiment Configuration



Impact on e-VLBI

- A-priori establishments of dedicated channels
 - Meet QoS requirements of real time experiment
 - Bandwidth
 - Acceptable packet loss over various time scales
 - Delay
 - Jitter
- Network configuration at the end systems
 - VLAN setup
 - IP configuration scheme
 - IP Addressing process
 - Monitoring of network performance
- Deployment of VLSR software to switches

e-VLBI – DRAGON Interface

- Integrate e-VLBI
 - Method to request resources
 - Generic for all applications
 - Centralized management and configuration application
 - E-VLBI Access Point (eAXP)
 - XML Profile parameters
 - <network></network>
 - Provide end point network specific information
 - E.g. NIC types, protocols, etc.
 - Presently defining Interaction / Configuration
-

Conclusion

- Continue VSI-E / EGAE
 - Deployment and testing
 - In-house
 - Upgrade software
 - Japan to Haystack
- DRAGON Integration
- Target
 - Real time data transfer using DRAGON
 - IGRID 2005 Conference (Sept. 05)
 - Super Computing 2005 Conference (Nov. 05)