**In-VIGO**

J. Fortes

Advanced Computing and Information Systems Laboratory
University of Florida

Europar 2005

---

**Putting it all together: GUI Application example**

Role-Based Access Control enables Single Sign-On to VNC X-window

1: User request
2: query (user data, compute server)
3: get VM or shadow account
4: copy/access user data
5: start application
6: return handler to user (URL)
7: VNC X-window

Total: 16 seconds
Total: 7.5 seconds

Physical server pool P

Data Server D1

Data Server D2

CMS/Portal Engine

Front end 'F'

isolation via shadow accounts or virtual machines

2.5 seconds
2.5 seconds
50ms
50ms
9.5 seconds
4 seconds
4 seconds
1 second

ACIS Advanced Computing and Information Systems Laboratory
In-VIGO Virtual Networks - ViNe

- IP overlay on top of the Internet
- Operation similar to site-to-site VPN
- Designed to address issues that VPN does not solve:
  - High administrative overhead: manageable only with few sites (i.e., for all joins/leaves every network administrator needs to be involved in reconfiguration)
  - VPN firewalls need to have a static public IP address: no support for multi-level NAT and ISPs that do not offer static addresses.

Each participating host is configured with an additional IP address in ViNe space (IP aliasing)
- Packets with destination in ViNe space are directed to VRs for routing in ViNe space.
**ViNe – Local communication**

- Local communication is kept local in both Physical and Virtual space.
- ViNe does not interfere with physical communication.
- Virtual space can be used only when needed.

**ViNe – Communication in virtual space**

Original, unmodified packet VH1→VH2 is delivered.

Example: VH1 sends a packet to VH2.

ViNe packet is encapsulated with an additional header for transmission in physical space: B→A/(VH1→VH2)

VRB looks up its routing table. In this case the table indicates that the packet should be forwarded to "A".

ViNe header is stripped off for final delivery.

Packet with header VH1→VH2 is directed to VRB.
ViNe – Firewall/NAT traversal

- VRs connected to the public network proxy (queue) packets to VRs with limited connectivity. The latter open connection to the queue VR to retrieve packets.
- VRs with limited connectivity are not used when composing routing tables. Routing tables are made to direct packets to queue VRs.
- The approach supports multi-level NAT.
- The approach also works under DHCP since the changing IP is not considered for routing.

ViNe – Self-organization

- Routing tables are created/destroyed as needed (e.g., join/leave of sites, creation of a new virtual network, etc).
- VRs communicate with each other to exchange routing information.
- All VR-to-VR communication is digitally and cryptographically authenticated.
- VR authentication is similar of that used in SSH protocol: each VR has one VR host certificate. PKI is used to establish session keys.
- Communication of sensitive information (e.g., routing tables, VRs host certificates) is encrypted.
- Administrator of a joining site is involved only during the setup/configuration phase. Joins/leaves do not need administrators' intervention.
**ViNe – Security**

- ViNe does not route packets to the Internet.
- Internet routers do not route packets to ViNe (ViNe uses the private IP space).
- Site specific security policies are not changed by enabling ViNe. A minimal change may be required to allow ViNe traffic (in the private IP space).
- ViNe packets are IP packets, so they can be under the same inspection (e.g., by firewalls and intrusion detection systems) of regular network traffic.

**ViNe – Performance**

- Low virtualization (tunneling) overhead: 0 – 5% depending on available processing power of involved VRs and available physical network performance, when firewall/NAT traversal is not required.
- When firewall/NAT traversal is required the allocation of VRs to proxy/queue traffic, processing power of VRs, and available physical network performance impact ViNe performance. Degradation between 10 – 50% was observed in initial experiments. Optimizations on VR allocation is under investigation.
ViNe – Performance

Physical Network Performance

<table>
<thead>
<tr>
<th></th>
<th>Throughput (Mbps)</th>
<th>Round-trip latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Florida</td>
<td>24.1</td>
<td>20.4</td>
</tr>
<tr>
<td>Purdue University</td>
<td>27.8</td>
<td>17.5</td>
</tr>
<tr>
<td>Northwestern University</td>
<td>6.82</td>
<td>63.1</td>
</tr>
</tbody>
</table>

ViNe – No NAT/firewall traversal required

<table>
<thead>
<tr>
<th></th>
<th>Throughput (Mbps)</th>
<th>Round-trip latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Florida</td>
<td>25.5</td>
<td>19.8</td>
</tr>
<tr>
<td>Purdue University</td>
<td>29.3</td>
<td>16.7</td>
</tr>
<tr>
<td>Northwestern University</td>
<td>8.26</td>
<td>27.7</td>
</tr>
</tbody>
</table>

ViNe – Performance – NAT/firewall traversal

Physical Network Performance

<table>
<thead>
<tr>
<th></th>
<th>Throughput (Mbps)</th>
<th>Round-trip latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Florida</td>
<td>24.1</td>
<td>20.4</td>
</tr>
<tr>
<td>Purdue University</td>
<td>27.0</td>
<td>15.3</td>
</tr>
<tr>
<td>Northwestern University</td>
<td>27.8</td>
<td>17.5</td>
</tr>
</tbody>
</table>

ViNe – NAT/firewall traversal required

<table>
<thead>
<tr>
<th></th>
<th>Throughput (Mbps)</th>
<th>Round-trip latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Florida</td>
<td>31.1</td>
<td>12.0</td>
</tr>
<tr>
<td>Purdue University</td>
<td>29.3</td>
<td>15.8</td>
</tr>
<tr>
<td>Northwestern University</td>
<td>6.82</td>
<td>63.1</td>
</tr>
</tbody>
</table>
ViNe – Performance – NAT/firewall traversal

<table>
<thead>
<tr>
<th>Physical Network Performance</th>
<th>University of Florida to Purdue University</th>
<th>University of Florida to Northwestern University</th>
<th>Northwestern University to Purdue University</th>
</tr>
</thead>
<tbody>
<tr>
<td>Round-trip latency (ms)</td>
<td>24.1</td>
<td>20.4</td>
<td>27.8</td>
</tr>
<tr>
<td>Throughput (Mbit/s)</td>
<td>27.8</td>
<td>17.5</td>
<td>6.82</td>
</tr>
<tr>
<td>ViNe–NAT/firewall traversal required</td>
<td>37.5</td>
<td>11.8</td>
<td>29.8</td>
</tr>
<tr>
<td>Throughput (Mbit/s)</td>
<td>29.8</td>
<td>15.2</td>
<td>8.30</td>
</tr>
<tr>
<td>Round-trip latency (ms)</td>
<td>37.5</td>
<td>11.8</td>
<td>29.8</td>
</tr>
<tr>
<td>Throughput (Mbit/s)</td>
<td>29.8</td>
<td>15.2</td>
<td>8.30</td>
</tr>
</tbody>
</table>

ViNe – Performance in LAN environment

TCP throughput and round-trip latency for different CPU configurations, in a LAN environment (1 Gbit/s)
**Application virtualization**

- Users
- Grid Computer Resource(s)
- Virtual Application 1 Service
  - Service Interface 1
  - Instance of Virtual Application 1
  - Instance 1 of Physical Application 1
- Virtual Application 2 Service
  - Service Interface 2
  - Instance of Virtual Application 2
  - Instance 2 of Physical Application 1
- Operating System
- Libraries
- Application Binary
- Grid Storage Resource(s)
- Application Execution Environment
- Web Portal

**Virtual Application Monitor**
- Regular Service
- Restricted Service
- Composite Service
- Augmented Service

**Grid-enabling unmodified applications**

- Enabler provides
  - Command-line syntax
  - Application-related labels
  - Parameter(s), type-set values, entire applications
  - Resource and execution environment metadata
  - Architecture, OS libraries, environment variables
- Grid-services are created, deployed and possibly customized using
  - Generic Application Service (GAP)
  - Virtual Application Service (VAS)
- Grid-user interacts with the virtual application through a Web-portal to execute applications on virtualized resources
**Command syntax example**

```
dineroIV -informat D
  -informat d
  -informat p
  -informat P
  -informat b

-skipcount U
-flushcount U
-maxcount U
-stat-interval U
-tracefile

-on-trigger A
-off-trigger A
-stat-idcombine
```

- Argument with a set of values
- Group of arguments
- Group of arguments that “depends” on another group
- Optional argument
- Argument with a value constrain (an hexadecimal address)
- Exclusive set of arguments

- **Level N “expands” into**

```
UniDataOpts
InstrOpts
```

- Excluser set of arguments

- **UniDataOpts and InstrOpt further “expand”**

---

**Enabling Service and VAS Generator Service**

Virtual Application Enabling

- **Enabler**
  - **HTTPS**
  - Portal Interface

- **SPS**
  - Command-line syntax, command-line description in natural language, resource requirements and execution environments settings

- **Enabling Service**
  - **SOAP**

- **XSD**

- **VAS Generator Service**
  - **PROP**
  - Resource requirements and execution environments settings in properties file format.

Application Generation and Utilization

- **User**
  - **HTTPS**
  - Portal Interface

- **WSDL**
  - Virtual Application Service description, importing the XML Schemas.

- **Java**
  - Input parameters types.

- **Java + WSDD**
  - Service implementation and deployment descriptor.

- **VM Service**
- **VFS Service**
- **ViNe Service**

---

**Tools**

**Protocols**

**Other Frameworks**

**VAS Framework**
In-VIGO approach to Single Sign-On (SSO)

- In-VIGO users do not “own” resources
- Resources are accessed by In-VIGO middleware on behalf of users
- This approach enables:
  - Reuse of local identities on resources (short-lived user identities)
  - Dealing with any kind of user credential representation or resource access scheme
  - Easy user creation/removal
In-VIGO approach to SSO

- Role Based Access Control (RBAC)
  - Entities/resources in In-VIGO have a set of permission groups which define operations
  - Users are grouped into roles (user classes)
  - In-VIGO administrator defines the mapping between roles and permission groups

![Diagram showing RBAC roles and permissions]

Implementation

- Role Based Access Control
- Short-lived User Identities

In-VIGO assigns localuserY to start application on machineX on behalf of user1
Later In-VIGO may assign the same localuserY on machineX to run another application on behalf of user2
Is user1’s data exposed to user2?
VDF S support

- Short-term ID to user data mapping: Virtual Distributed File System

Implementation

- Multiple grid users to short-term ID: Collaborative environments using VNC