VIPE

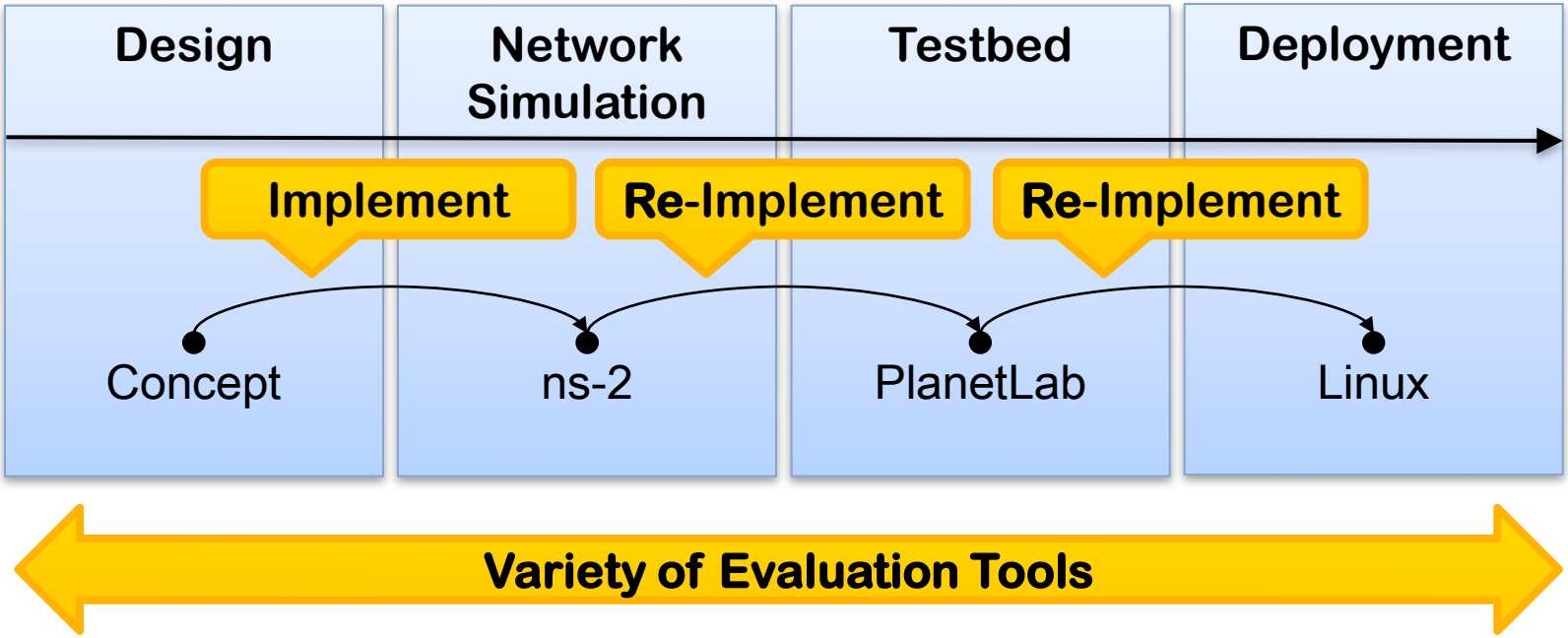
A Virtual Platform for Network Experimentation

Olaf Landsiedel, Georg Kunz, Stefan Götz, Klaus Wehrle

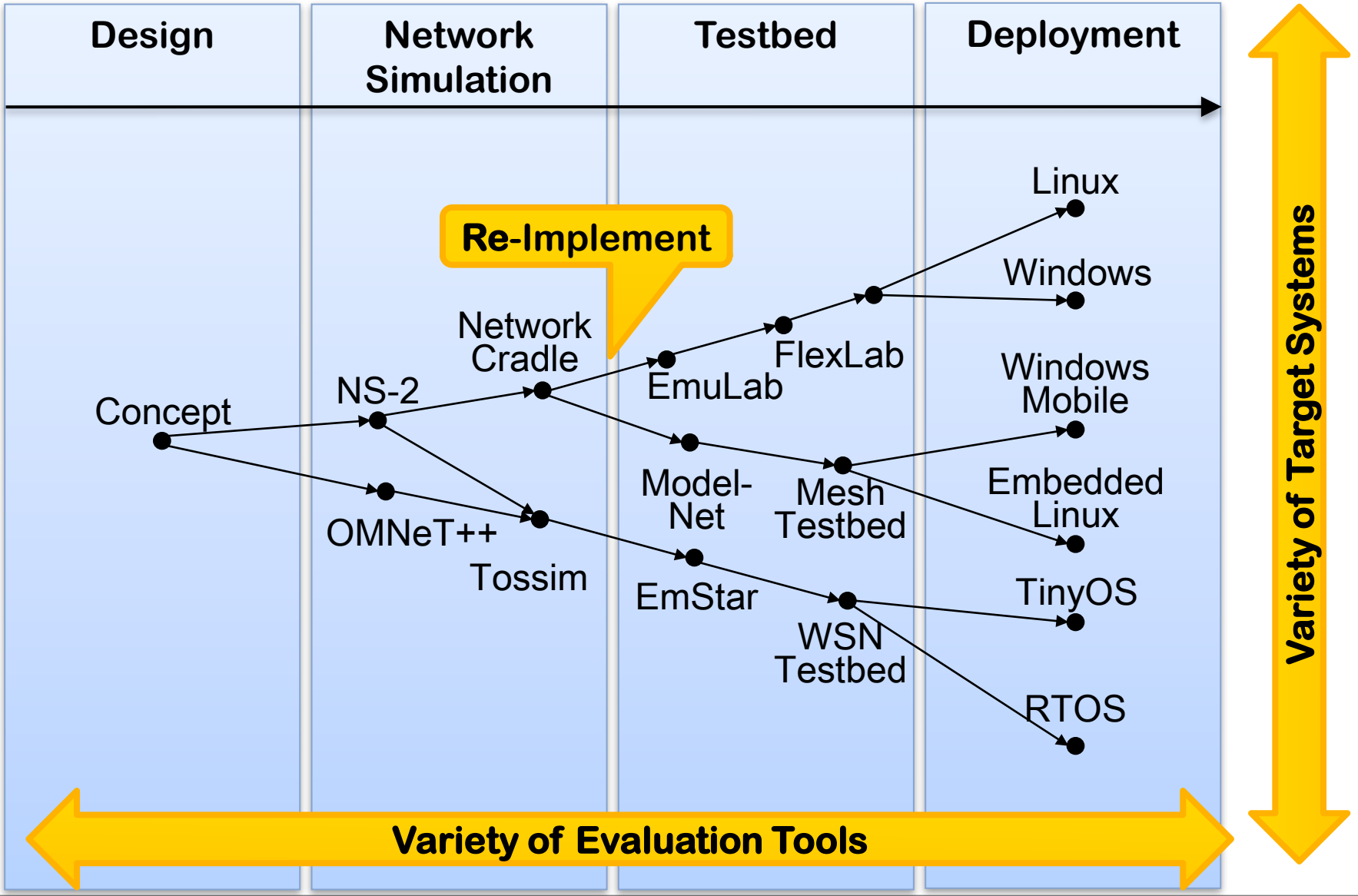
http://ds.cs.rwth-aachen.de

Barcelona / SIGCOMM VISA, August 2009

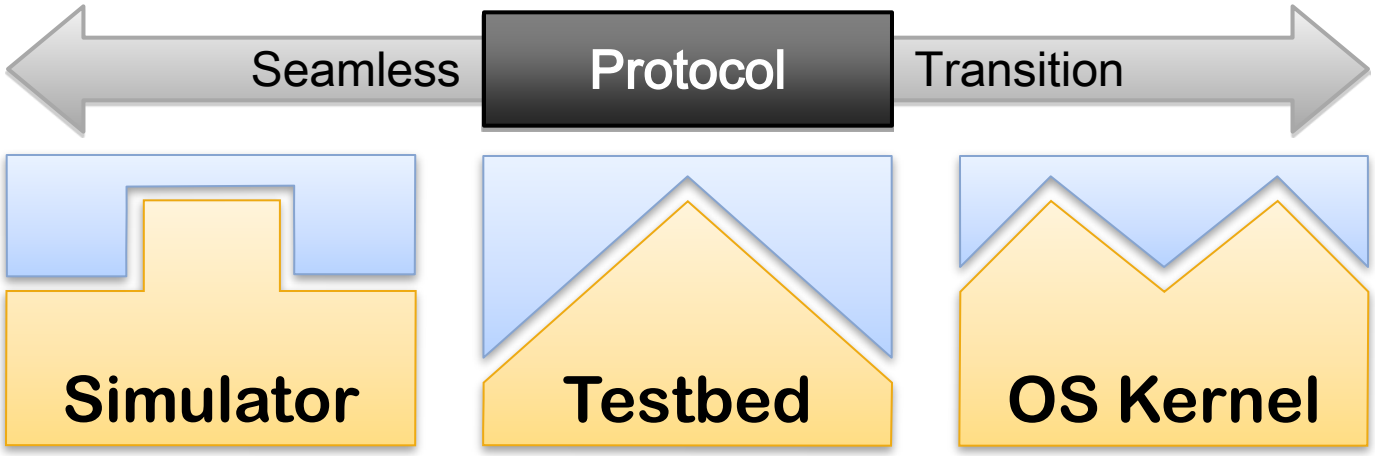
Motivation


Design
Network Simulation
Testbed
Deployment
Implement
Re-Implement
Re-Implement
Concept
ns-2
PlanetLab
Linux
Variety of Evaluation Tools

Motivation


Design
Network Simulation
Testbed
Deployment
Re-Implement
Concept
NS-2
OMNeT++
Network Cradle
Tossim
EmuLab
FlexLab
Model-Net
Mesh Testbed
EmStar
WSN Testbed
Linux
Windows
Windows Mobile
Embedded Linux
TinyOS
RTOS
Variety of Target Systems
Variety of Evaluation Tools

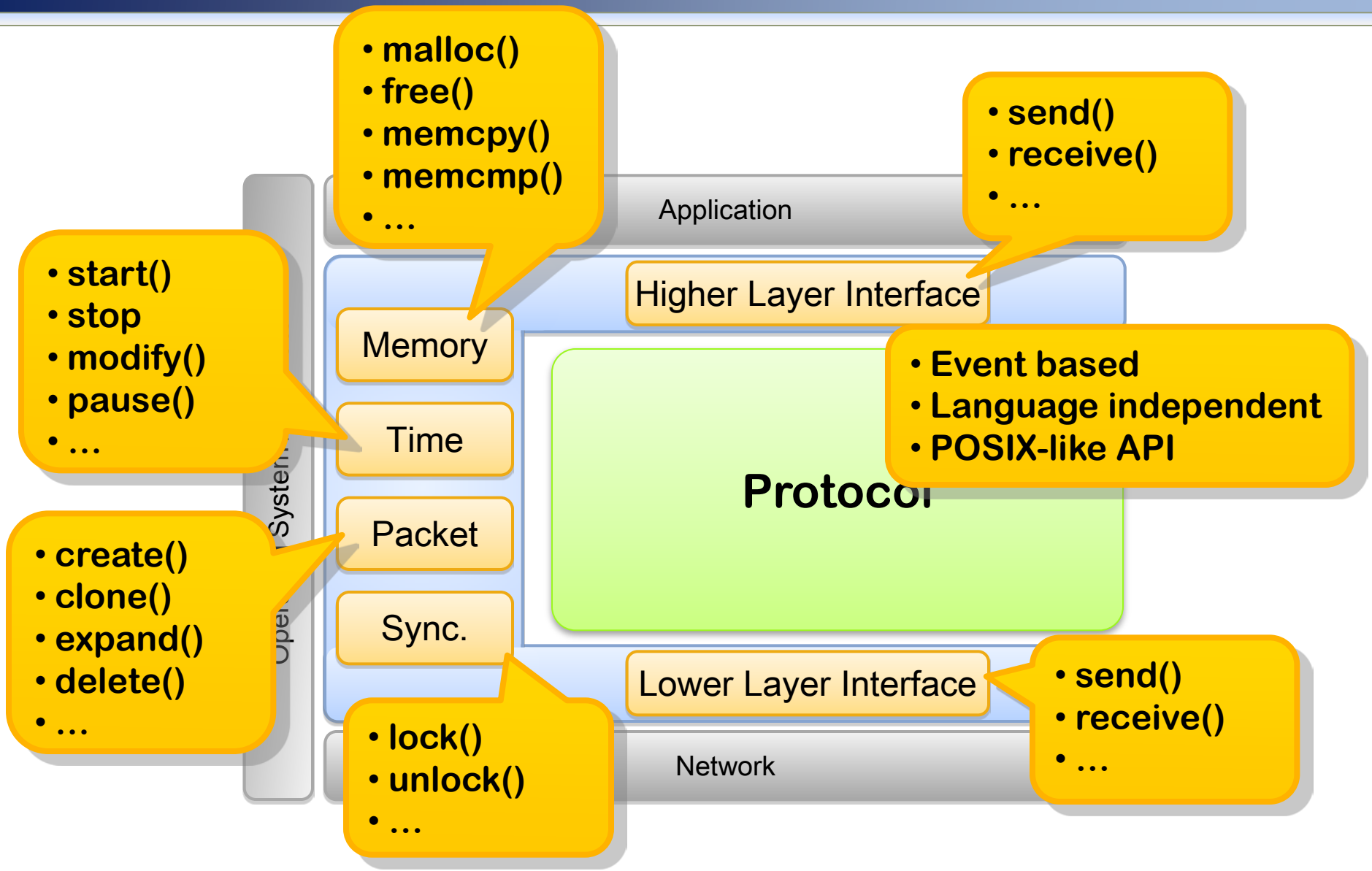
Concept



- **Challenge**
  - Completeness vs. complexity of abstraction

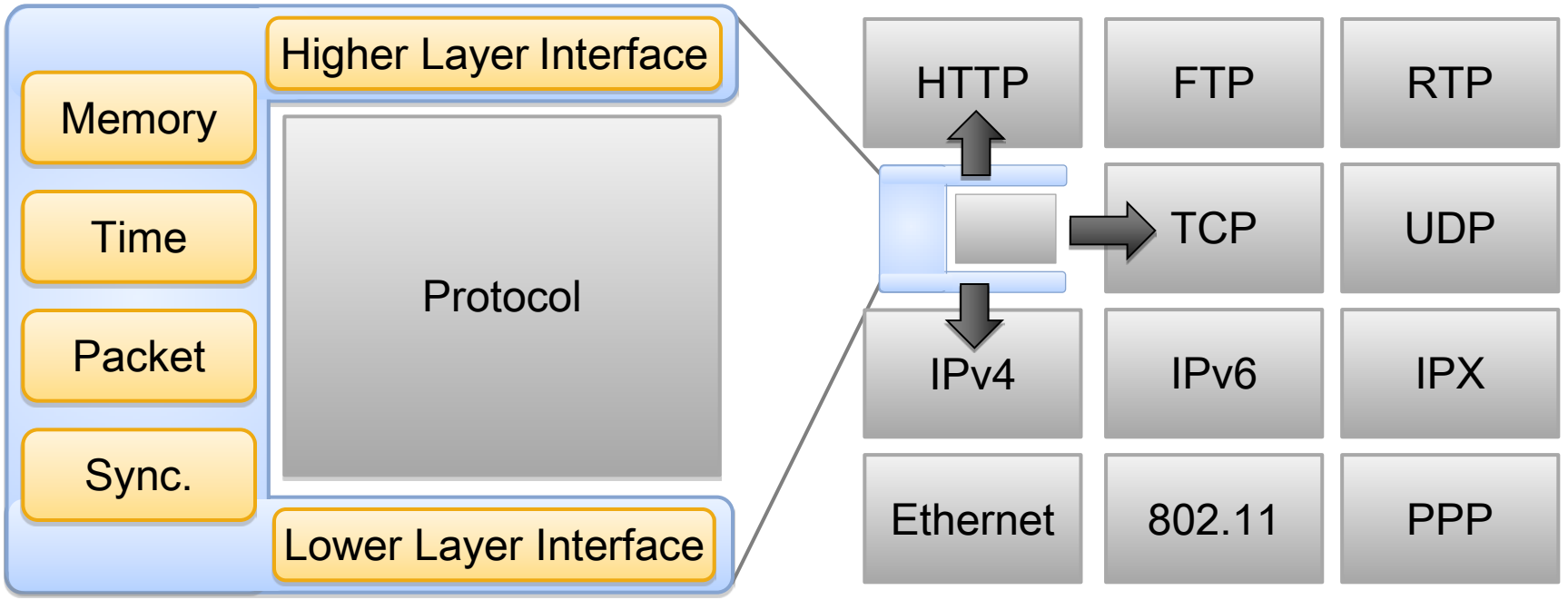
⇨ **Our goal:** Lightweight abstraction

Design

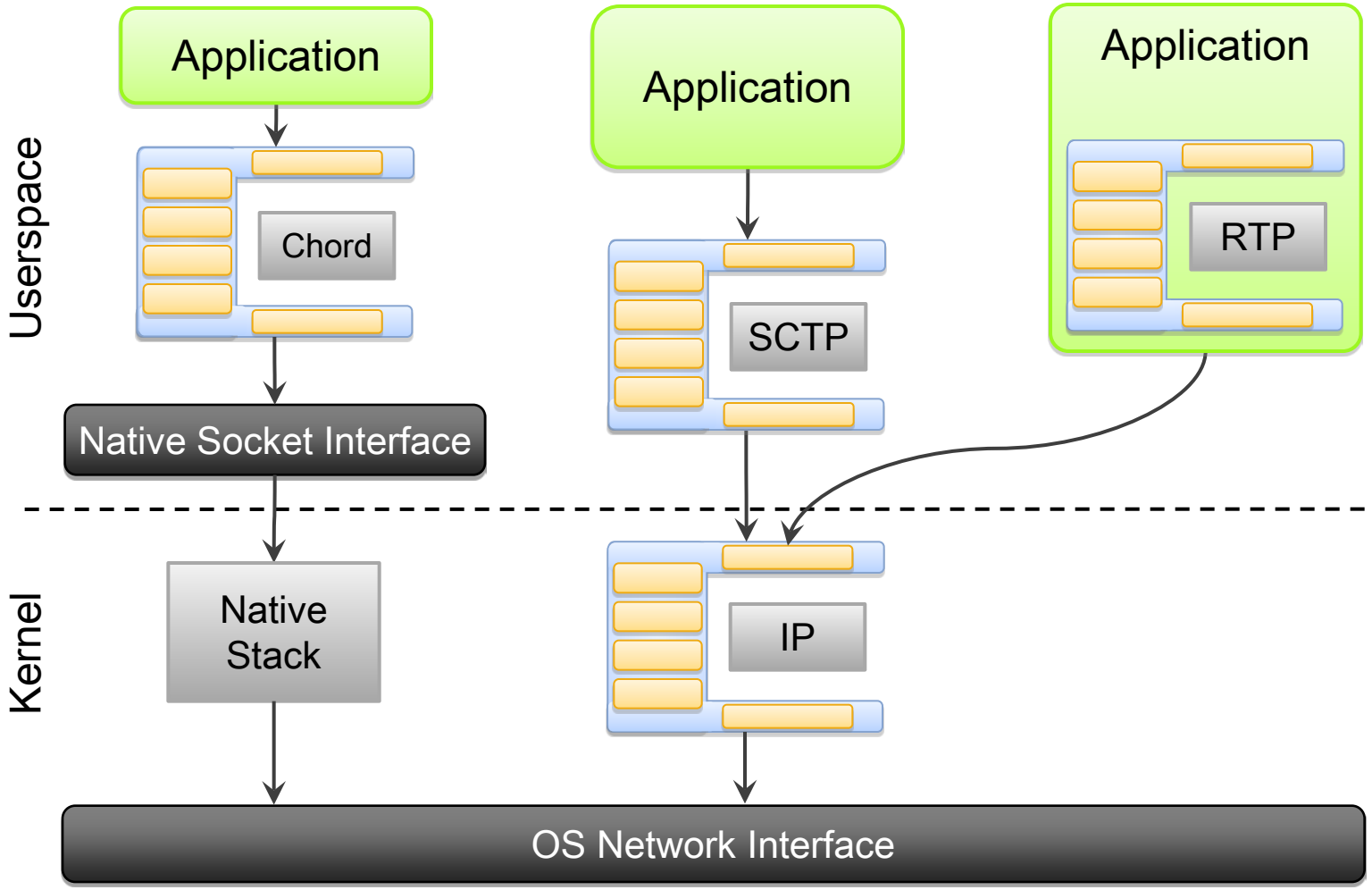

• malloc()
• free()
• memcpy()
• memcmp()
• …
• send()
• receive()
• …
Application
Higher Layer Interface
• start()
• stop
• modify()
• pause()
• …
Memory
Time
Packet
Sync.
Protocol
• Event based
• Language independent
• POSIX-like API
• create()
• clone()
• expand()
• delete()
• …
Lower Layer Interface
• send()
• receive()
• …
• lock()
• unlock()
• …
Network

Protocol Integration

- **Flexible Placement**
  - Layer independent
  - Utilize existing protocols / simulation model



Use Cases

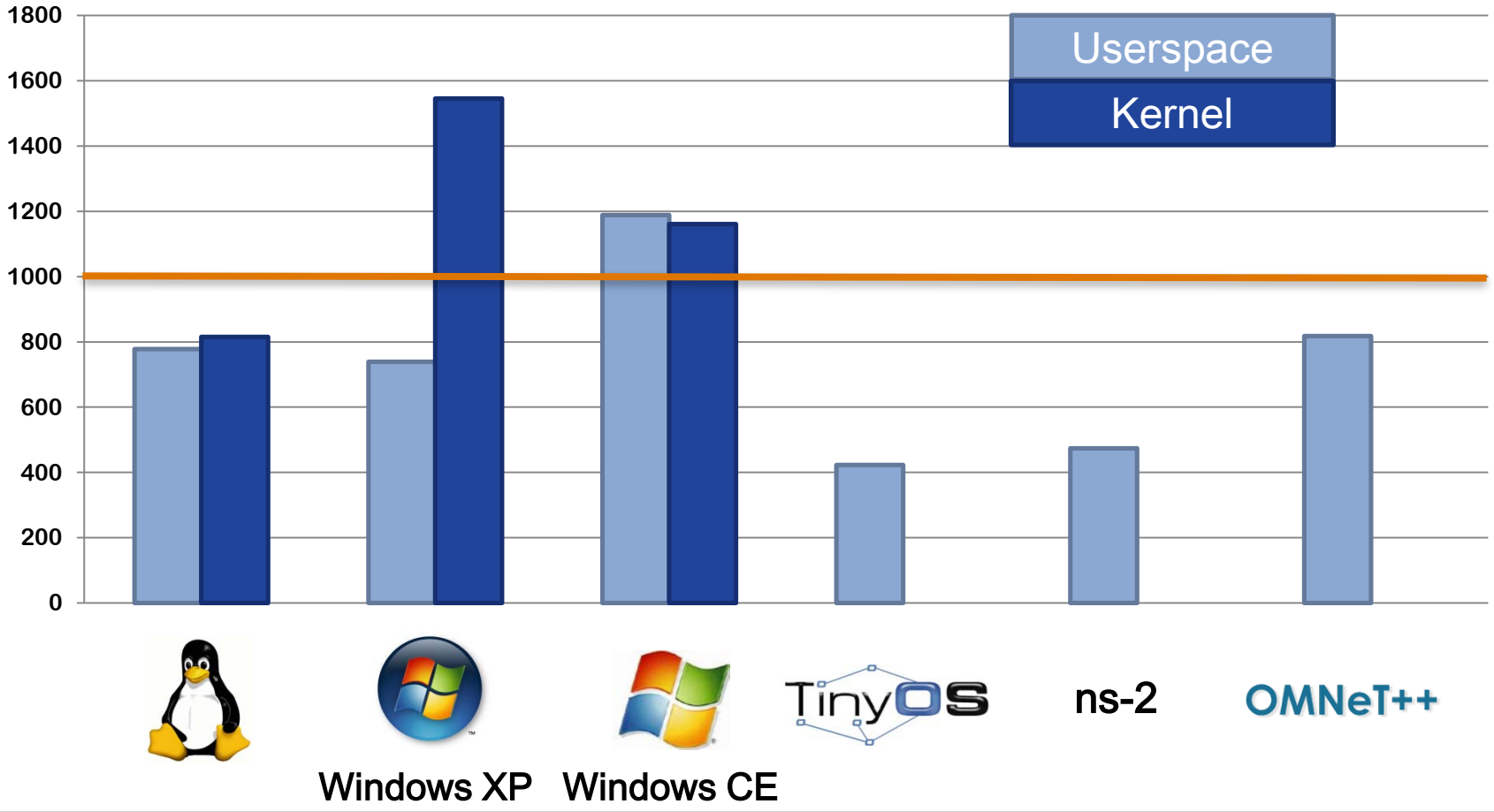

Application
Application
Application
Userspace
Chord
SCTP
RTP
Native Socket Interface
Native
Stack
IP
Kernel
OS Network Interface

Runtime Overhead

- **Benchmarks of Virtual Resources**
  - CPU time-stamp counter

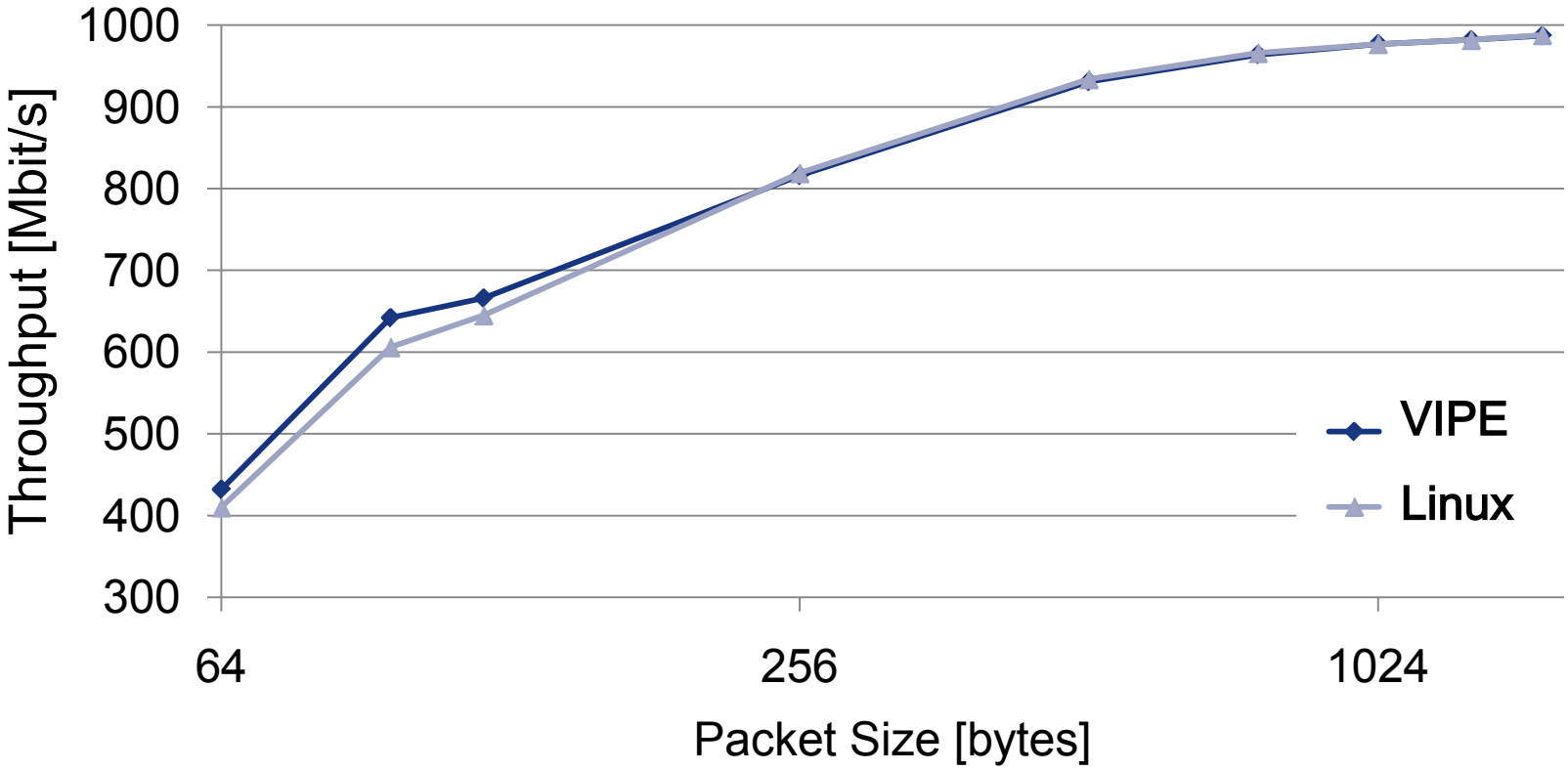
<table>
<tr><th></th><th colspan="2">Windows XP</th><th colspan="2">Linux 2.6.22</th></tr>
<tr><th></th><th>Kernel</th><th>Userspace</th><th>Kernel</th><th>Userspace</th></tr>
<tr><td>Memory</td><td colspan="4" rowspan="3">0%
macros and function inlining</td></tr>
<tr><td>Synchronization</td></tr>
<tr><td>Timer</td></tr>
<tr><td>Device <code>send()</code></td><td rowspan="4">14.5%</td><td colspan="3" rowspan="4">≤ 0.2%
utilize platform specific
data structures</td></tr>
<tr><td>Device <code>receive()</code></td></tr>
<tr><td>Packet <code>create()</code></td><td>34.2%</td></tr>
<tr><td>Packet <code>delete()</code></td><td>9.7%</td></tr>
</table>

- **Implementation Complexity**
  - Lines of Codes



Runtime Performance

- **Macro-Benchmark**
  - Routing performance
  - Based on VIPE IPv4 implementation



Conclusion

- **VIPE: Virtual Platform for Network Experimentation**
  - Unified programming environment
  - Seamless transition between platforms

⇨ Tight feedback loop
⇨ **Evolve** NOT re-implement protocols

- **Lightweight Architecture**
  - Based on best practices in systems design

⇨ Small porting effort
⇨ Low performance overhead

Thank you for your attention.