

LARK

Bringing Distributed High Throughput Computing to the Network

Todd Tannenbaum

tannenba@cs.wisc.edu

U of Wisconsin-Madison

Garhan Attebury

attebury@cse.unl.edu

U of Nebraska-Lincoln

What is Lark?

- NSF-funded project to take first steps at bridging DHTC and the network layer
- Collaboration between UNL and UW-Madison

Some Lark Goals...

- Develop software for network interaction by leveraging recent advances in **SDN** and the Linux Kernel
- Integrate these advances into HTCondor

One example application...

- At UW-Madison, we want network traffic to pass thru border firewalls.
- *But...* thousands of HTCondor jobs running on large campus compute clusters can overwhelm firewalls.
- Policy: Would like network traffic for trusted compute jobs that need to move a lot of data to bypass firewalls

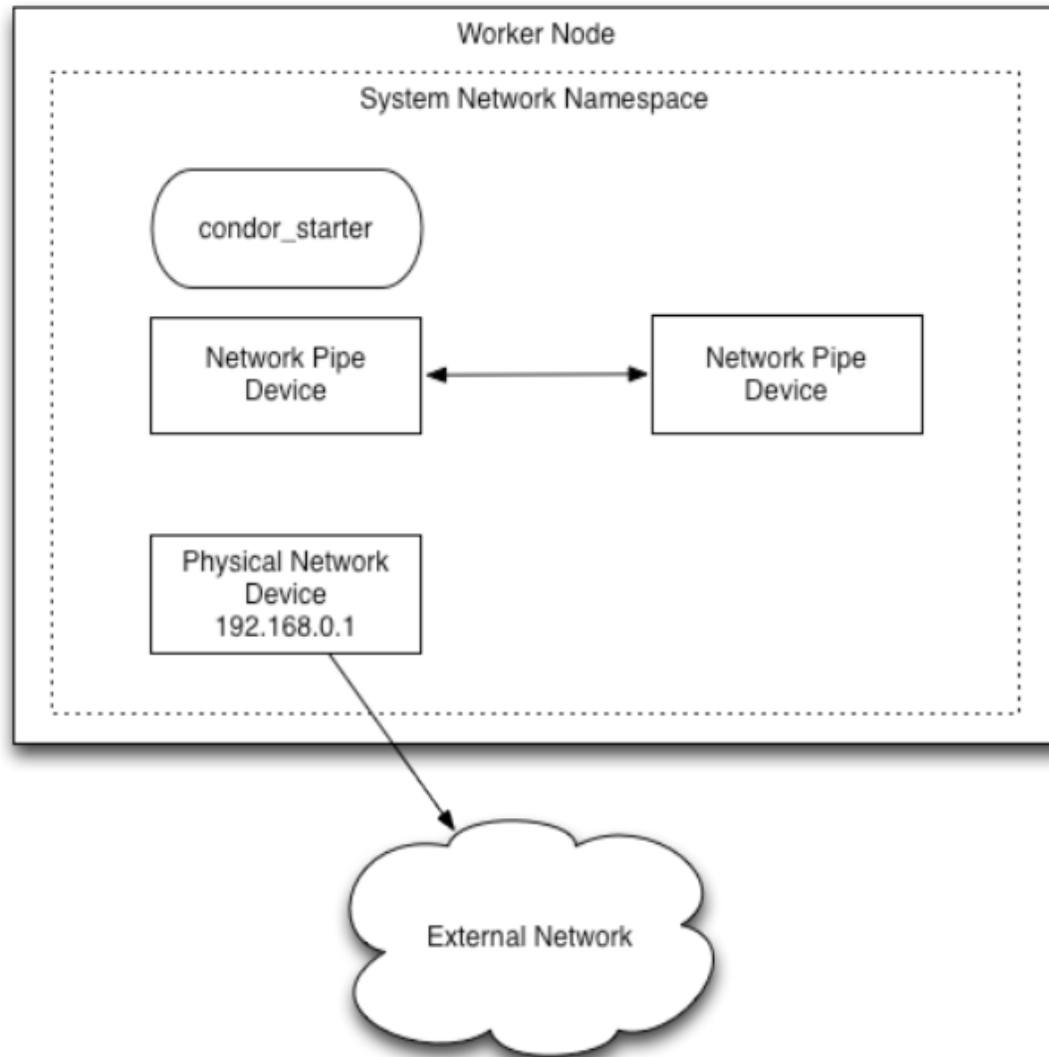
Demonstration

- Goal: Demonstrate HTCondor programming a unique network path per job
- How:
 1. HTCondor creates a per-job network device
 2. Locks the user's application to this device
 3. Communicates with Cisco ONE controller to define a path based on metadata associated with the job

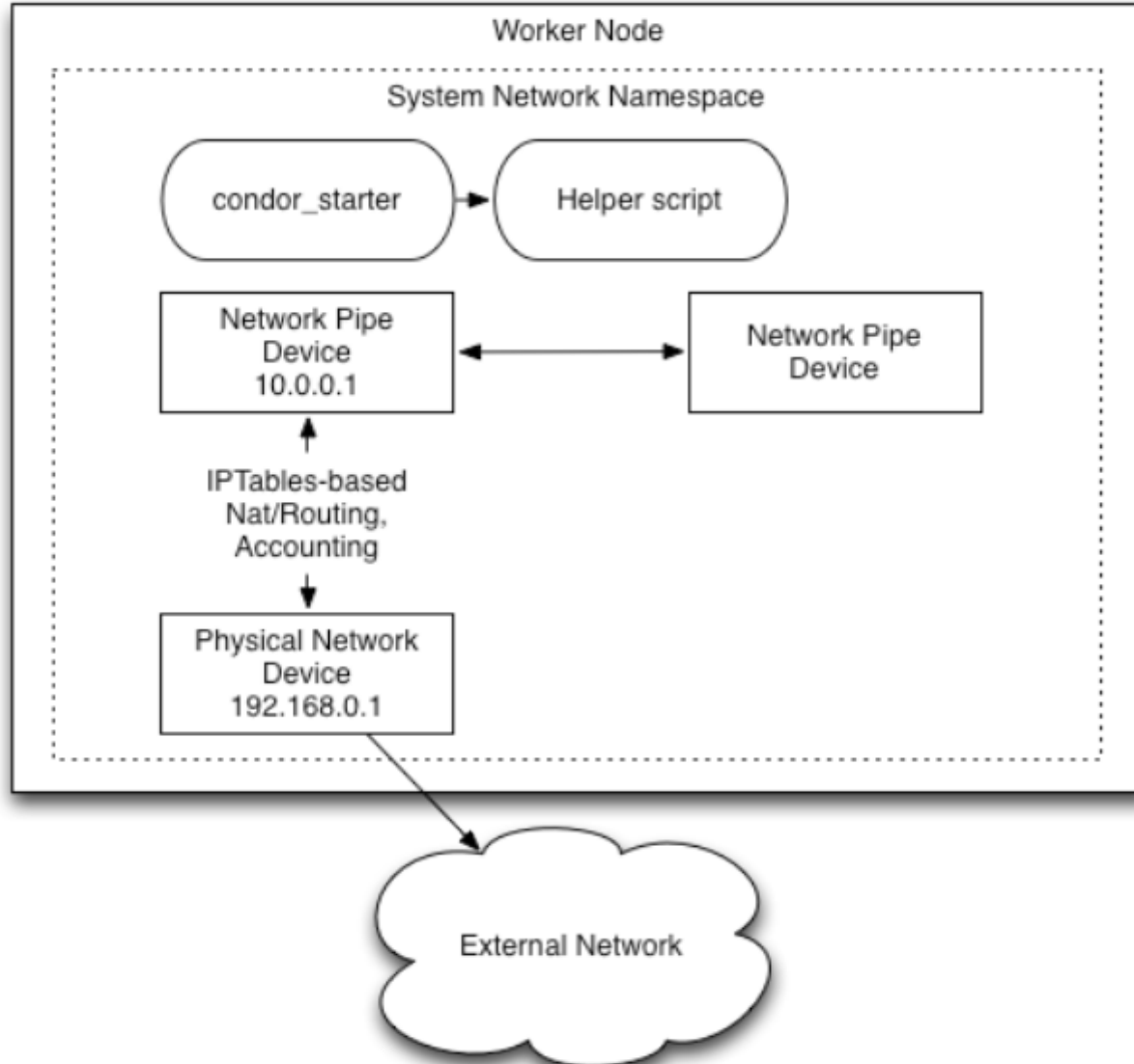
Todd bravely (foolishly?) gambles
with Murphy's Law by performing a
brief live demonstration



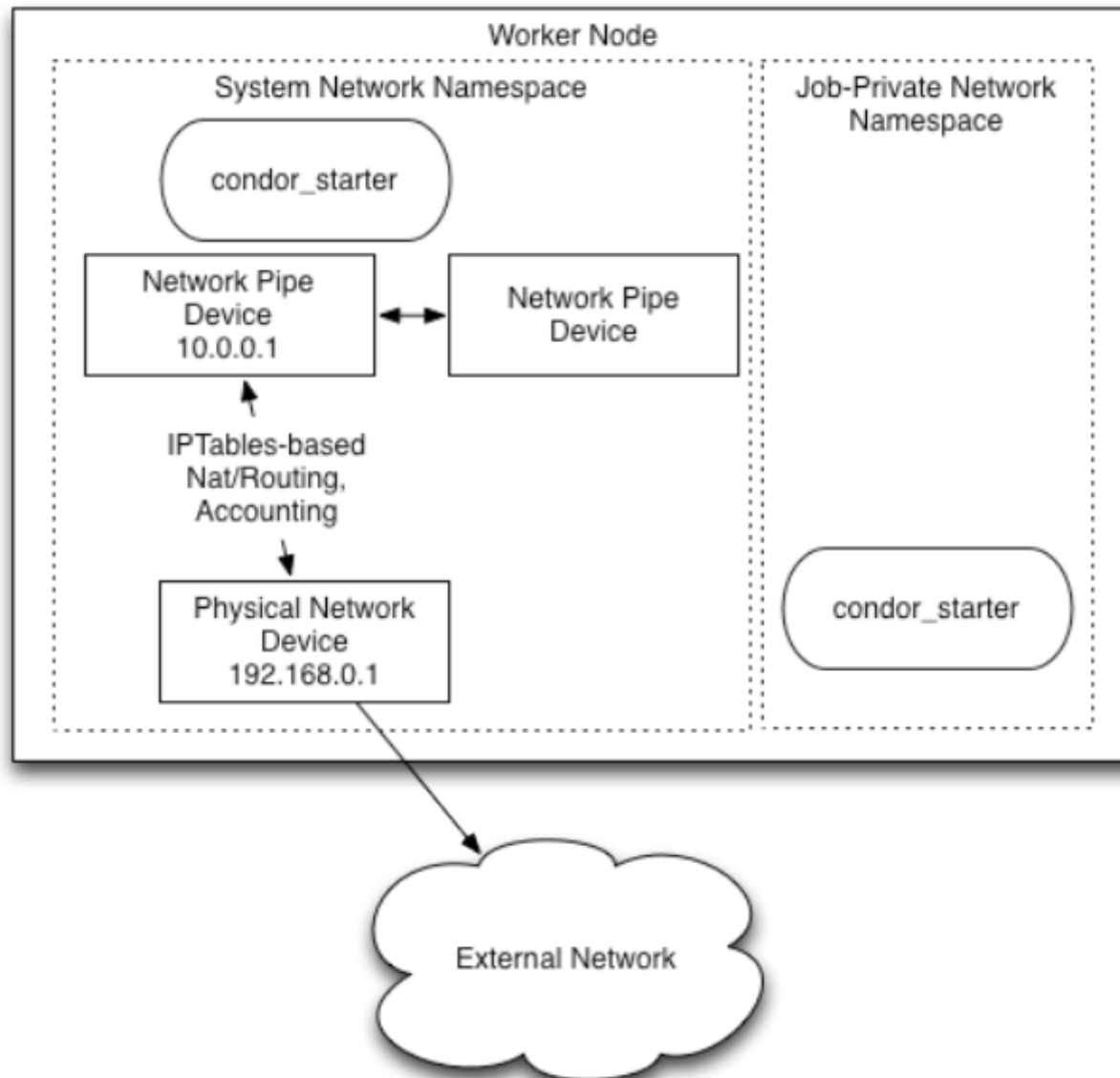
Starter creates network pipes



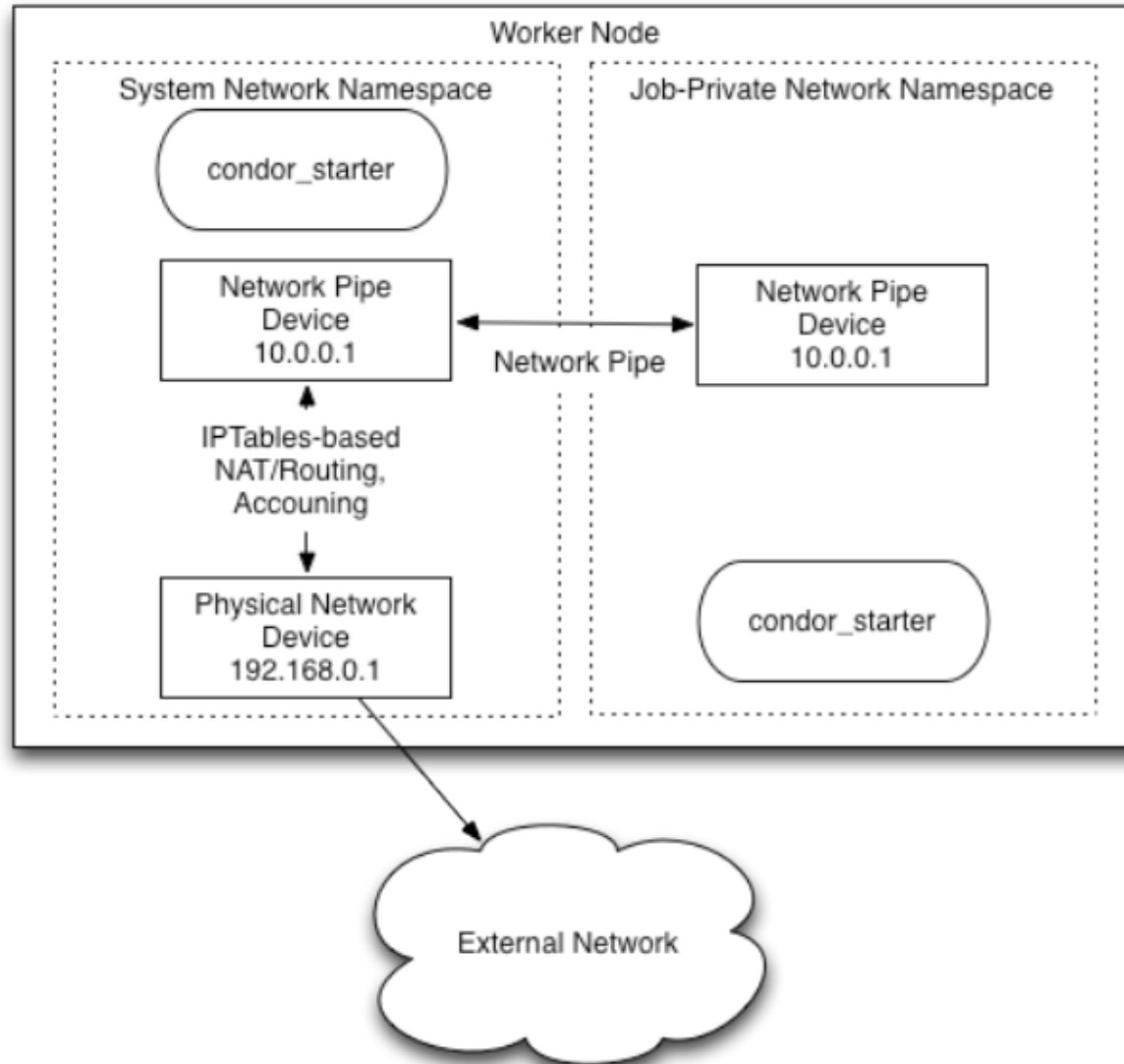
Starter creates a helper process
Helper configures IPTables and assigns addresses



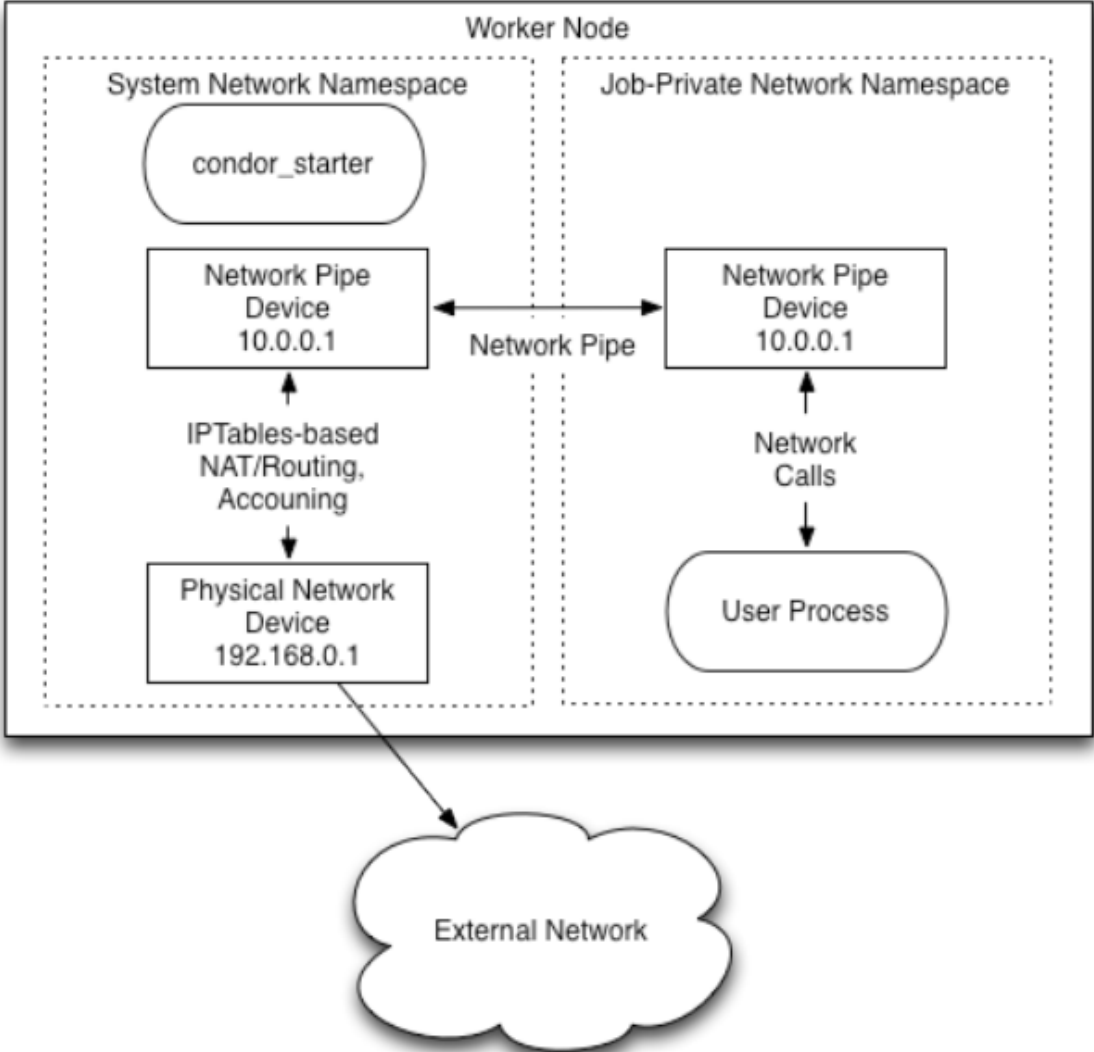
Starter forks new process with new network namespace



Parent starter passes one end of network pipe to network namespace,
Child starter configures routing and IP address



Final Configuration



Besides SDN integration, what are the other Lark goals and activities?

- Networking accounting
- Network policies
- DYNES integration
- perfSONAR integration
- IPv6 and network testbed

Network accounting

- Traditionally CPU hours are primary metric for academic clusters
- Networking not always free (EC2, non I2)
- More data, bigger data
- Interface per job allows detailed and accurate network network accounting

Network policies

- A few options...
 - My job requires no inbound connectivity (NAT)
 - My job needs full connectivity / public IP (Bridge)
 - My job requires port X for ObscureLicenseServer™
 - My job is part of a special workflow (VLANs)
- Security policies

DYNES integration

“Bridging DHTC and the network layer”

- DYNES (Dynamic Network System) allocates, schedules, and prioritizes channels to provide bandwidth reservation to data flows
- API interface allows scheduler to reserve ‘dedicated’ path for workflows

perfSONAR integration

- perfSONAR publishes what it measures
- Collect statistical information (bandwidth, availability, etc...) into ClassAds
- Example: Using information from perfSONAR, Condor can determine expected bandwidth to a remote site and limit a workflow to match

IPv6 and Network Testbed

- Condor has (some) IPv6 support already, needs thorough testing
- Testbeds at both UW and UNL

Project status

- Per job namespaces:
 - IPv4 (NAT + bridging)
 - OpenFlow rules
- perfSONAR:
 - pS metrics in ClassAds
- DYNES:
 - work in progress
- IPv6:
 - basics work in condor, still some work

