The Knowledge Plane Program
A new way to “think” about the Internet

J. Christopher Ramming
January 31, 2003
Current IPTO, DARPA themes

- Vision of cognition as the next major paradigm shift in computing
  - Creating new computing capabilities
  - Fundamentally altering the growth curve of system complexity
- Concern over risks of increased reliance on networks
  - The role of the network is growing more quickly than our ability to manage and protect it
  - Network-centric warfare has promise and peril
  - The civilian economy is alternately helped and hurt by the Internet

**Key Idea:** The Internet Knowledge Plane as a basis for making progress in cognition (an opportunity of national importance) while exploring a new vision for network architecture (a problem of national importance)
The Knowledge Plane Program
Anticipated accomplishments by 2009

1. We have enabled generalized **learning and reasoning** with new cognitive system techniques
   - Separation of algorithm, policy, goals, and knowledge
   - New models and approaches to modeling

2. New techniques for **collective (distributed) cognition** have led to synergies across multiple “K-Apps” e.g. agents in a heterogeneous trust environment
   - Shared structural models
   - Market-based mediation mechanisms
   - Attention to privacy and security

3. Visions of networking 2012 become feasible because we have learned to **manage ever-growing complexity**
   - Applications have new abilities to “peer into” and leverage the Internet via the KP
   - We have “solved” internetwork management

4. The Internet, one of our most complex and successful distributed systems, is recognized to have had the **attributes needed to fully explore cognitive techniques**
   - Multiple administrative domains, co-opetition amongst stakeholders, inevitability of partial knowledge, need to support “naïve” users, global setting
   - A research community with deep experience in complex distributed systems

5. The science developed in the KP context is **exported to other domains**
   - We have new understanding of complex distributed systems
   - We have developed new general-purpose cognitive techniques

= Assertions, queries, requests, observations

= End system
Benefits of the Knowledge Plane

- New “collective cognitive” mechanisms for supporting cooperation and learning
- A coherent management infrastructure for the Internet that does not compromise its strengths
- Additional military benefits: quick deployment, more effective networks, and reduced reliance on human experts
  - Will provide a standards-based solution to military requirements concerning self-configuring, self-managing networks
  - Will provide vastly improved network diagnostic capabilities in battlefield conditions
  - Will provide applications with topology and route awareness that can be used to improve efficiency and/or robustness
  - Might provide an infrastructure for responding effectively to modern, fast-acting Worms (a “reach” goal)
Sample K-Application: “Why?”
Fault management is illustrative of key issues in cognition and networking

- Description of the “Why?” program
  - “Why?” explains and (in the long run) fixes network abnormalities
  - Relevant data is represented, routed, and aggregated in the Knowledge Plane
  - Information “features” are analyzed using modern probabilistic models, inference engines
  - Actuation in better-than-human time

- How is it done today
  - Ad-hoc, out-of-band sharing of human-readable information between operators
  - Low-level tools like “ping” and “traceroute”

- What’s new
  - Observations from multiple vantage points
  - Collective action to resolve problem
  - Mixed-mode distributed learning
  - Framework for privacy, security, and marketplaces of data
  - Endpoint participation and knowledge sharing
“Why?” Progress and Metrics
Sequentially raising the bar

<table>
<thead>
<tr>
<th>Cognition Scope</th>
<th>Networking Scope</th>
<th>Challenge Descriptions</th>
<th>Metrics Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Why?” in the host learning and response to surprise. Immunity from malicious manipulation</td>
<td>Exhibition of observations from endpoint</td>
<td>Fault injection at endpoint. Limited error types. Malicious observations to simulate mistakes</td>
<td>MTTD. Quality of explanation. Number of error types handled. Rate of improvement.</td>
</tr>
<tr>
<td>“Why?” in the local area Ability to generate new strategies, not just explanations</td>
<td>Ability to generate observations from multiple vantage points</td>
<td>More complex faults, and with selective disabling of “the usual” sensors</td>
<td>…PLUS “Cost”/efficiency of strategies. Rate of improvement. Ease of improvement. Impact of malicious action</td>
</tr>
<tr>
<td>“Why?” in the wide area Market-based cooperation protocols, trust, security</td>
<td>Sense &amp; actuate across administrative domains</td>
<td>Inter-AS style faults (use a testbed), this time with artificially imposed privacy and security concerns</td>
<td>…PLUS MTTR</td>
</tr>
</tbody>
</table>

More Challenging
The Interplay Between K-Apps

There is a relationship between learning and collective cognition

Potential K-Apps

- Passive “Why?” (fault detection, isolation)
- Active “Why?” (including repair)
- “Vital Signs” (monitoring for, e.g. worms)
- “Shadow Routing” (exploring alternatives and impact prior to decisions and commitment)
- “Inferring Hidden Facts” (such as physical topology)
- “ISP Status & Benchmarks” (open, methodologically sound)
- “Topology Reporting” (for CDNs, PSTN gateway location, route control)
- “Wide-area management support” (Route filtering, inferring AS relationships, AS traceroute)
- “Web of Trust” (Adding judgment to authentication)

Insights

- Separations of concern enable independent evolution as well collective cognition
  - Perhaps K-Apps will expose knowledge structures, not just behaviors
- The issues of learning and collective cognition are both deeply affected by trust boundaries
  - Perhaps K-Apps will handle trust boundaries using similar mechanisms in both cases
Elements of the Knowledge Plane

The Knowledge Plane’s “central nervous system” is the first thing to understand

- Developing extensible, compositional, and distributed operational models of the Internet
- Aggregating and representing sensor data (“routing” knowledge)
- Controlling distributed networks of sensors and actuators across trust boundaries
- Expressing policies. Resolving policy conflicts
- Mechanisms for dealing with a partially hidden environment
- Designing powerful, distributed “core cognition” engines
- Maintaining and incorporating appropriate history / knowledge in a distributed setting
- Risk mitigation priorities

KEY:
“Routing” Knowledge
Exploring one aspect of the Knowledge Plane in detail

- The right knowledge at the right place at the right time
  - Route observations to cascade of “think points”
    - Aggregation and grade-based filtering.
  - Route explanations to observers and control points.
  - Route grades backwards, knowledge forward.
  - Pre-position explanations and responses
- Multi-mode routing
  - Diffusion as well as query/response
  - Attribute-based routing
  - Routing through a virtual KP topology as well as real topology
  - Inter-domain as well as intra-domain

Elements of the Knowledge Plane
Things we should explore before getting up two whole communities

- Developing extensible, compositional, and distributed operational models of the Internet
- Expressing policies. Resolving policy conflicts
- Designing powerful, distributed core cognition engines
- Aggregating and representing sensor data (routing knowledge)
- Maintaining and incorporating appropriate history / knowledge in a distributed setting
- Controlling distributed networks of sensors and actuators across trust boundaries
- Mechanisms for dealing with a partially hidden environment

KEY:
- Risk mitigation priorities
Technology Foundations
Why we know enough to embark on this project

Domain-specific languages

Bayes belief nets, machine learning, genetic algorithms, neural networks, expert systems

Active Networks, Sensor Nets, CoABS, various overlay networks

Algorithmic game theory

Distributed Hash Tables (DHTs)

RKF, DAML, Knowledge Representation, dimensionality reduction

DASADA, NMS
11

12 3 4 5

Preliminary K-App demos (ad-hoc implementation)

Design/implement models and algorithms

Demo of apps running on KP plus functional challenge problems

Integration/porting to KP

Teams working together to obtain synergies and re-use of common infrastructure

Collaboration

Support of applications on testbed (cooperation between platform R&D + K-Apps)

Multi-operator knowledge plane

Selected Risk Mitigation

v1.0 abstract architecture

v1.0 concrete APIs, protocols

Analysis, revision, work with IETF

Design and implementation

Initial design, deploy

Engineering for scale, reliability, security

Analysis and revision, multi-operator testbed

Tools for mgmt, monitoring, control

Year

1 2 3 4 5

Platform R&D Team
(Core Arch + Impl)

20 short-term / specialized
K-Application-Oriented Teams
(Apps, Models, Algorithms)

5 long-term, specialized

Testbed Team
(Deployment and Ops)

1

1

Insight sharing

Collaboration

Strawman to prime the K-App teams

Demo of testbed

Challenge problems

Transition to industry
Backup materials and experimental slideware
Heilmeier's catechism

To evaluate research activities at Darpa, Heilmeier formulated a set of questions that so well expresses the fundamentals of his beliefs that he seriously refers to it as his "catechism." He later taught it to his research "novitiates" at Texas Instruments and now enforces its use at Bellcore. Like a preflight checklist, his catechism provides a routine for safely and successfully launching a research project:

What are you trying to do? Articulate your objectives using absolutely no jargon.
How is it done today, and what are the limitations of current practice?
What's new in your approach and why do you think it will be successful?
Who cares? If you're successful, what difference will it make?
What are the risks and the payoffs?
How much will it cost? How long will it take?
What are the midterm and final "exams" to check for success?

Heilmeier attributes much of his success to his imposition of a disciplined thought process on project management. It allowed him to curb and clarify both the enthusiasms of his researchers and the resource demands of his managers.
The Knowledge Plane
A new vision for network architecture

Problem: In order for the Internet to play the role we envision in 2012, we will need to rethink our approach to managing its complexity and exposing its capabilities. We must address the weaknesses of the Internet without diluting its strengths.

Solution: A cognitive overlay on the Internet
- The Knowledge Plane, open management infrastructure to sense and control network functions
- Specific K-Applications that use this infrastructure to address longstanding problems in network management, fault detection/isolation/repair, and other areas
- An architecture that forges a stronger Internet by supporting cooperation across multiple administrative domains (issues of trust, security, and market mechanisms)
- An architecture that exploits the cognitive metaphor and develops general-purpose cutting-edge techniques for reasoning, representation, learning

<table>
<thead>
<tr>
<th>Internet Strength</th>
<th>Corresponding Weakness</th>
<th>Illustrations</th>
</tr>
</thead>
<tbody>
<tr>
<td>resilient packet forwarding and routing</td>
<td>&quot;management space&quot; has not yet been addressed</td>
<td></td>
</tr>
<tr>
<td>Decentralized architecture</td>
<td>Applications and tools have only a local perspective</td>
<td>Fault isolation is very hard. Detecting worm propagation requires aggregated knowledge. Traffic engineering requires aggregated knowledge.</td>
</tr>
<tr>
<td>Multiple administrative domains</td>
<td>Inevitable stakeholder conflicts and failures to coordinate can be problematic</td>
<td>Flash crowds, operator error, new applications, hot potato routing and policy conflicts can cause problems that are hard to understand and address</td>
</tr>
<tr>
<td>Simple, transparent core and undifferentiated packets</td>
<td>No application-specific support</td>
<td>The Internet does not know what applications are running or what their individual needs are. Can't distinguish legitimate traffic from undesirable traffic</td>
</tr>
</tbody>
</table>
## Example K-Application:

Fault detection/isolation/repair

<table>
<thead>
<tr>
<th>Challenge Level</th>
<th>Action</th>
<th>Hard problems</th>
<th>Examples</th>
<th>Learning</th>
<th>How to evaluate progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quite Challenging (5 year)</td>
<td>Fault isolation and repair based on distributed observations and effects</td>
<td>Distinguishing between faults/attacks and benign patterns. Finding appropriate representations of the patterns. Using effectors across trust/administrative boundaries</td>
<td>Attack pushback across administrative boundaries</td>
<td>Allow diagnostic rule base to grow and shrink in an “open source” fashion to ensure gradual improvement. Use AI/ML techniques that train themselves over time, exhibit algorithmic evolution, and generally have a separation of concerns that enables various aspects to evolve independently (Bayes Nets for example)</td>
<td>Enumerate the faults that can be reliably detected / isolated / repaired. Maintain user metrics concerning % of successfully treated problems. In testbed scenarios, use fault injection techniques (possibly extending to red/blue team exercises for attack scenarios). Reduction in mean-time-to-detect, mean-time-to-repair, and in the long run, proactive detection and prevention of impending failures</td>
</tr>
<tr>
<td>Challenging (3 year)</td>
<td>Fault isolation based on active distributed observations</td>
<td>Scheduling/planning appropriate and efficient non-local observations</td>
<td>An extended “Why” application capable of e.g. diagnosing the loss of packets fragmented at a NAT point or routing problems across autonomous systems</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Challenging (18 month)</td>
<td>Fault isolation based on passive distributed</td>
<td>Fault isolation in general</td>
<td>“Why?” application</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Cognitive Networks Workshop

- David Clark – MIT
- Michael Kearns – University of Pennsylvania
- Bob Braden – USC-ISI
- Deborah Estrin – UCLA
- Craig Partridge – BBN
- Larry Peterson – Princeton
- J. Christopher Ramming
- Stefan Savage – UC-San Diego
- Scott Shenker – UC-Berkeley
- Jonathan M. Smith – University of Pennsylvania
- Tom Dietterich – Oregon State
- Satinder Singh – University of Michigan
- Amy Greenwald – Brown
- Jeff Kephart – IBM

Other notable events
  - Significant focus on network configuration and management. Further discussions of the Knowledge Plane
  - First third of the agenda devoted to the Knowledge Plane. Related talks by David Clark, Craig Partridge, Chris Ramming
## Example K-Application:
Network [re]-configuration

<table>
<thead>
<tr>
<th>Challenge Level</th>
<th>Action</th>
<th>Hard problems</th>
<th>Examples</th>
<th>Learning</th>
<th>How to evaluate progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quite Challenging (5 year)</td>
<td>Multiple administrative domains, complex routing</td>
<td>Trust, policy negotiation, representation, plus all of the below</td>
<td>Configuring and managing a full ISP</td>
<td>[Re]configuration, e.g., traffic engineering, requires characterization of patterns and expectations that must be learned over time. Configuration instances can also “learn” from prior experiences of self and others. Must recognize not only changing patterns of traffic, but departures from engineering assumptions</td>
<td>Time / cost / personnel needed for deployment (e.g. 82nd airborne example). Analysis of differential improvements resulting from reconfigurations. Testbed such as Jay Lepreau’s may be useful for offline analysis. Some operators may be persuaded to participate in early deployments.</td>
</tr>
<tr>
<td>Challenging (4 year)</td>
<td>Single administrative domain, complex protocols, complex internal topology</td>
<td>Dealing with policies, plus all of the below</td>
<td>Routed LAN with internal partitions / firewalls</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Challenging (3 year)</td>
<td>Single administrative domain, complex protocols</td>
<td>Representation of policies, goals, network abstractions. Deriving configuration both from environmental analysis and generative techniques based on representations</td>
<td>Routed LAN with multiple egress</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moderate (2 year)</td>
<td>Single administrative domain, simple protocols</td>
<td></td>
<td>Unrouted LAN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example K-Application:
Reducing BGP convergence time

<table>
<thead>
<tr>
<th>Challenge Level</th>
<th>Action</th>
<th>Hard problems</th>
<th>Examples</th>
<th>Learning</th>
<th>How to evaluate progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Challenging</td>
<td>Reducing BGP convergence time by determining appropriate settings for BGP parameters MIN_ROUTE_ADVERT_TIMER and the 5 ROUTE_FLAP_DAMAGE parameters</td>
<td>Gathering and interpreting global knowledge about expected behaviors. Undesirable hysteresis effects.</td>
<td>Cisco’s current default for MIN_ROUTE_ADVERT_TIMER is 30 seconds, but this number was picked out of a hat.</td>
<td>Characterizing normal and desired behavior. Developing models that reflect engineering assumptions. Incorporating gradual changes in industry norms and usage patterns by applications settings.</td>
<td>Measuring and tracking convergence times, possibly using “beacons”, in testbeds and eventually in the real world.</td>
</tr>
</tbody>
</table>

Questions:
• Are there increasingly difficult versions of this problem? Is it exhibited in other settings?
• What exactly is the global knowledge that is needed to adjust the parameters accurately?

Related work:
• IPTO’s NMS
Example K-Application:
Open, accurate ISP benchmarking

<table>
<thead>
<tr>
<th>Challenge Level</th>
<th>Action</th>
<th>Hard problems</th>
<th>Examples</th>
<th>Learning</th>
<th>How to evaluate progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>???</td>
<td>Create an open, accurate, version of existing proprietary ISP benchmarking systems e.g. Keynote</td>
<td>Methodology. Deriving an infrastructure that is re-useable across multiple applications. Should be robust in the face of “gaming”.</td>
<td>Measuring and publishing meaningful statistics about latency and jitter across various ISPs</td>
<td>Interesting goal would be to “learn” optimal placement of sensors</td>
<td>Industry acceptance over proprietary, occasionally unsound alternatives.</td>
</tr>
</tbody>
</table>

Questions:
- Can the methodology problem exhibited by e.g. Keynote and Matrix (Internet Weather Report) be described succinctly? What are the problems?

Related work:
- IDMaps (measures end-to-end latency, not an aggregate ISP evaluation)
**Example K-Application:**
Routing observations / shadowing

<table>
<thead>
<tr>
<th>Challenge Level</th>
<th>Action</th>
<th>Hard problems</th>
<th>Examples</th>
<th>Learning</th>
<th>How to evaluate progress</th>
</tr>
</thead>
<tbody>
<tr>
<td>???</td>
<td>Route filtering</td>
<td>Determine whether an announcement can be discarded or not (to avoid problems ensuing from big routing tables)</td>
<td>Need global awareness to understand whether a filtered route is actually &quot;covered&quot; by another route</td>
<td>???</td>
<td>% reduction in routing table sizes</td>
</tr>
<tr>
<td>???</td>
<td>Inferring relationships between Ass based on routing</td>
<td>Making observations and inferences about routing paths, for instance that X is a customer of Y, or that X and Y are peers</td>
<td>(1) An ISP could detect inappropriate routing requests from a customer, for instance not routing traffic through its customer to an upstream provider (2) CDNs could use this information to better position servers</td>
<td>Need to &quot;learn&quot; difference between anomalies and legitimate quirks</td>
<td>Utility in avoiding provider errors. Increased efficiency and use of bandwidth in topology-aware applications</td>
</tr>
<tr>
<td>???</td>
<td>AS traceroute</td>
<td>Requires observations from multiple vantage points to deduce the role of every hop in a traceroute. For instance, the traceroute may have exchange point hops not listed in the AS path</td>
<td></td>
<td>???</td>
<td>% of hops in a traceroute that can be explained</td>
</tr>
</tbody>
</table>
Who isn’t solving the problem

<table>
<thead>
<tr>
<th>Initiative Type</th>
<th>Example</th>
<th>Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vendor Initiatives</td>
<td>IBM’s Autonomic Computing</td>
<td>On the vendor’s own products</td>
</tr>
<tr>
<td>Industry Initiatives</td>
<td>TMForum’s NGOSS</td>
<td>On paying ISP customers with the implicit goal of differentiation</td>
</tr>
<tr>
<td>Operator Initiatives</td>
<td>AT&amp;T’s “Concept of Zero” and “Concept of One”</td>
<td>On operator’s own network</td>
</tr>
<tr>
<td>IETF</td>
<td>SNMP, Distributed Management, policy based working group</td>
<td>Mired in history, not making progress, or too high-level</td>
</tr>
<tr>
<td>DARPA</td>
<td>FTN</td>
<td>Focus on specific attacks and problems, not the encompassing management challenge*</td>
</tr>
</tbody>
</table>

Gap: Open Cross-Vendor and Inter-Operator IP Network Management Support

*Q: is this characterization fair to FTN?
# KP Security from the Start

<table>
<thead>
<tr>
<th>What needs to be built in from the start</th>
<th>Factors, challenges, approaches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity</td>
<td>Determining whether identity should be managed in parallel and independently from real-world identity</td>
</tr>
<tr>
<td>Authentication and trust</td>
<td>Observations need to be authenticated with varying degrees of confidence for different applications. Can minimize scale problems by creating a hierarchy, and by using correlation and learning to filter out unreliable observations. PGP-style authentication may be more useful than CA-style authentication, since authentication is tied to trust, which involves a judgment call on the part of the endpoint.</td>
</tr>
<tr>
<td>Algebras of trust</td>
<td>It will be necessary to confer/delegate authority at times. Emerging frameworks like SD3, keynote may prove a useful foundation</td>
</tr>
<tr>
<td>Accounting for malicious or mistaken observations</td>
<td>Need to consider that the KP will never yield entirely consistent and complete observations. Two-pronged approach: “webs of trust” to identify trustworthy observations, and correlation techniques to discard untrusted observations. Note that authentication is insufficient; judgment is also needed</td>
</tr>
</tbody>
</table>
Another perspective on the KP

K-Applications with reflective and deliberative reasoning engines

Digested Info / Conclusions / … = Knowledge

Information about Environment / Constraints / Goals / Security

Laws and Design Rules (Formal Models of Real World)

Knowledge Plane

Sensing and actuation

Control Plane (Routing)

Data Plane (Forwarding)

D E V I C E S

E N D S Y S T E M

K-Applications with reflective and deliberative reasoning engines
Why cognitive systems?

- The cognitive metaphor is suggestive of what the Internet currently lacks
  - Standards-based “sensory” mechanisms to collect and aggregate data across administrative boundaries
  - Useful, extensible representations of information about the network
  - Effective models of policies, goals, the network, and its applications
  - Reasoning mechanisms that are not entangled with fixed models and data
  - General-purpose “effectors” capable of controlling large distributed systems in application-specific ways
  - The ability to learn and respond to surprise in order to keep pace with the evolution of applications and threats
Sample K-Application: Network Configuration

• Description
  – High-level formal models of networks are used to generate provably correct configurations
  – The Knowledge Plane “closes the loop” by feeding status back into the configuration engine so that it can make adjustments
  – Existing “adaptive” techniques are augmented with reflective, deliberative mechanisms
• How is it done today
  – An guild has emerged to individually tune router parameters
  – At best, configurations are generated with ad-hoc scripts (100K lines in the core routers!)
  – Network management standards are too low-level (SNMP) or too abstract (NGOSS)
  – Industry initiatives are parochial (vendor- and provider-specific solutions)
• What’s new
  – Deliberative and reflective reasoning engines to augment existing reactive/adaptive mechanisms
  – Emphasis on high-level formal models
  – Relationship between formal models and real-time feedback from system

Levels of achievement:
• Simple LAN (single administrative domain, no complex protocols or topology)
• Complex LAN (complex protocols and topology)
• Transit network (complex protocols, policy issues)
DARPA-Hard Problems / Risks

• Applying learning algorithms to network problems
  – Translation to the networking world is non-trivial
  – Non-stationary world
  – Untrustworthy inputs (game the system)
  – Closed-loop feedback (learning through knowledge rating)
  – Knowing what to measure
• Scalability of representation, routing, reasoning,…
  – Finding a representation for observations
  – Naming the observation (discovering observations)
  – Aggregation of like/unlike information
  – Reasoning about observations/events at a distance
• Not making matters worse (auto-immune problem)
  – Stability and convergence (self-stabilization in this domain)
  – Avoiding new vulnerabilities
• Dealing with un-trusted and malicious entities
  – Trust input (below), aggregation, reasoning, actuators (who’s driving)
  – Ownership, administrative decentralization
• Scalability to (growing) Internet size
• Trust, security, policy
  – New vulnerabilities introduced by the KP must be address \textit{a priori}
• Personal and commercial privacy