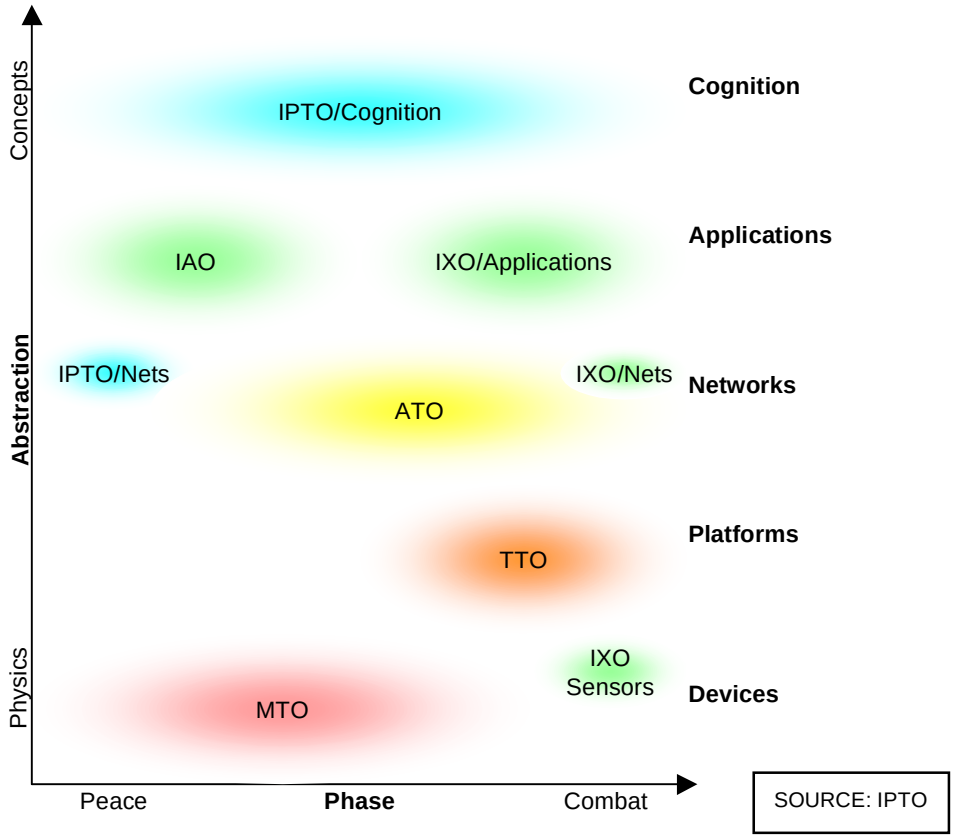
The Knowledge Plane Program

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)

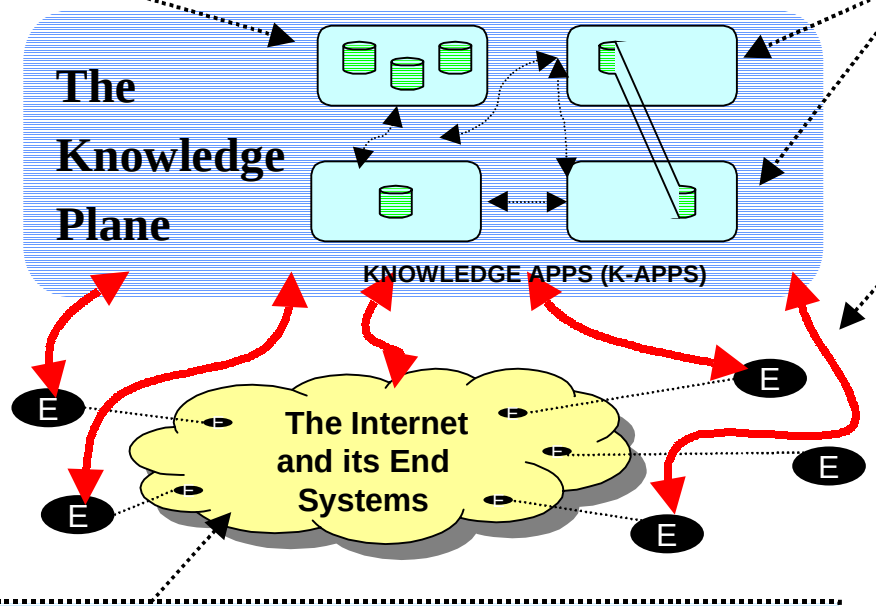
The Knowledge Plane Program

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

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



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

= Assertions, queries, requests, observations

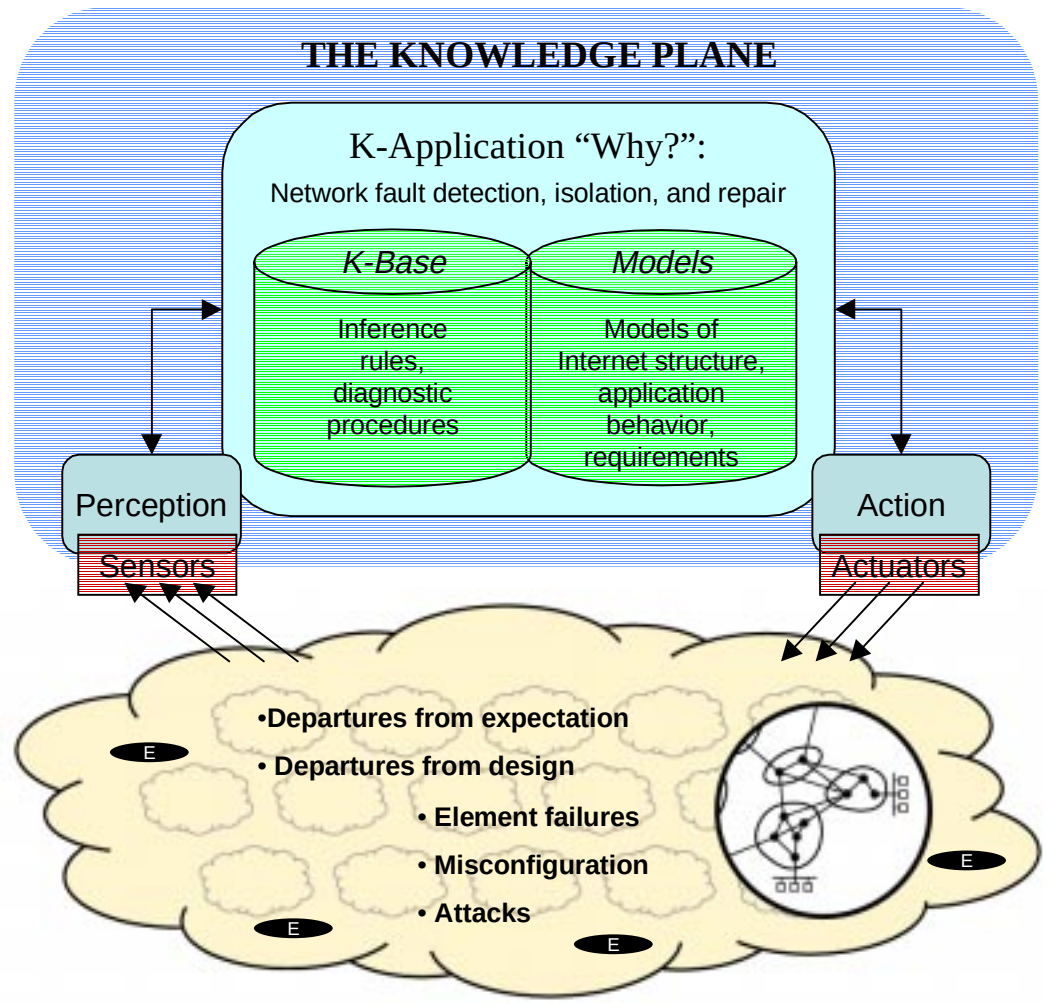
E = End system

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

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

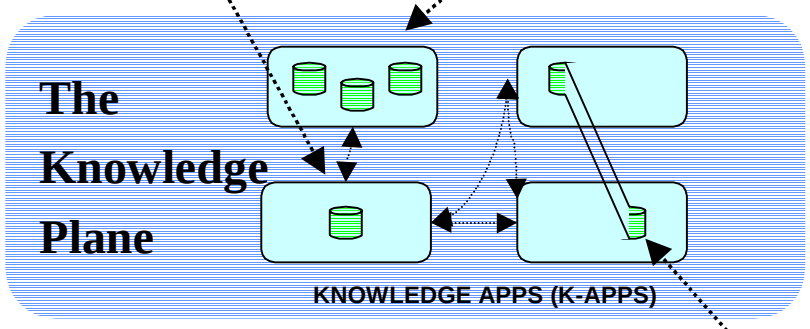
More Challenging

The Interplay Between K-Apps

There is a relationship between learning and collective cognition

Models from one setting can be re-used in another
- Firewall configuration models can inform intrusion detection

Knowledge gained in one setting can be used in another
- Topology reports can inform traffic engineering



Distributed instances of a an application could share knowledge ("collective cognition")
- Observations made by one ISP could help diagnose problems in another

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

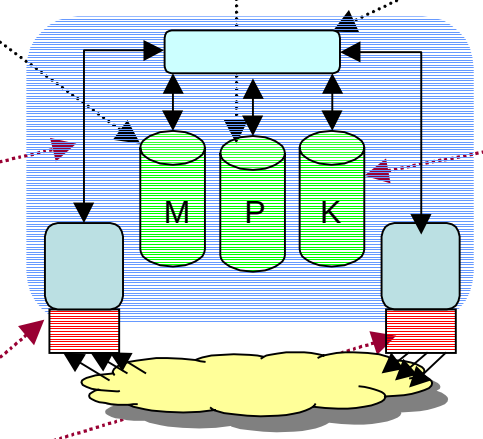
Elements of the Knowledge Plane

The Knowledge Plane's "central nervous system" is the first thing to understand

Developing extensible, compositional, and distributed operational models of the Internet

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

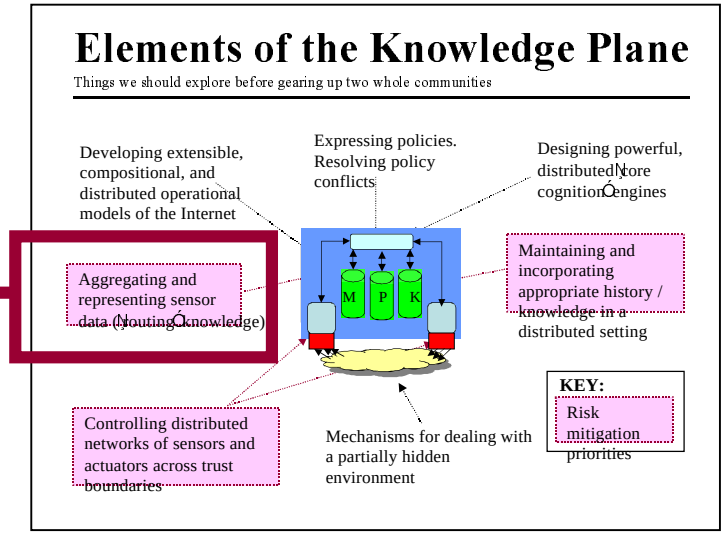
KEY:

Risk mitigation priorities

“Routing” Knowledge

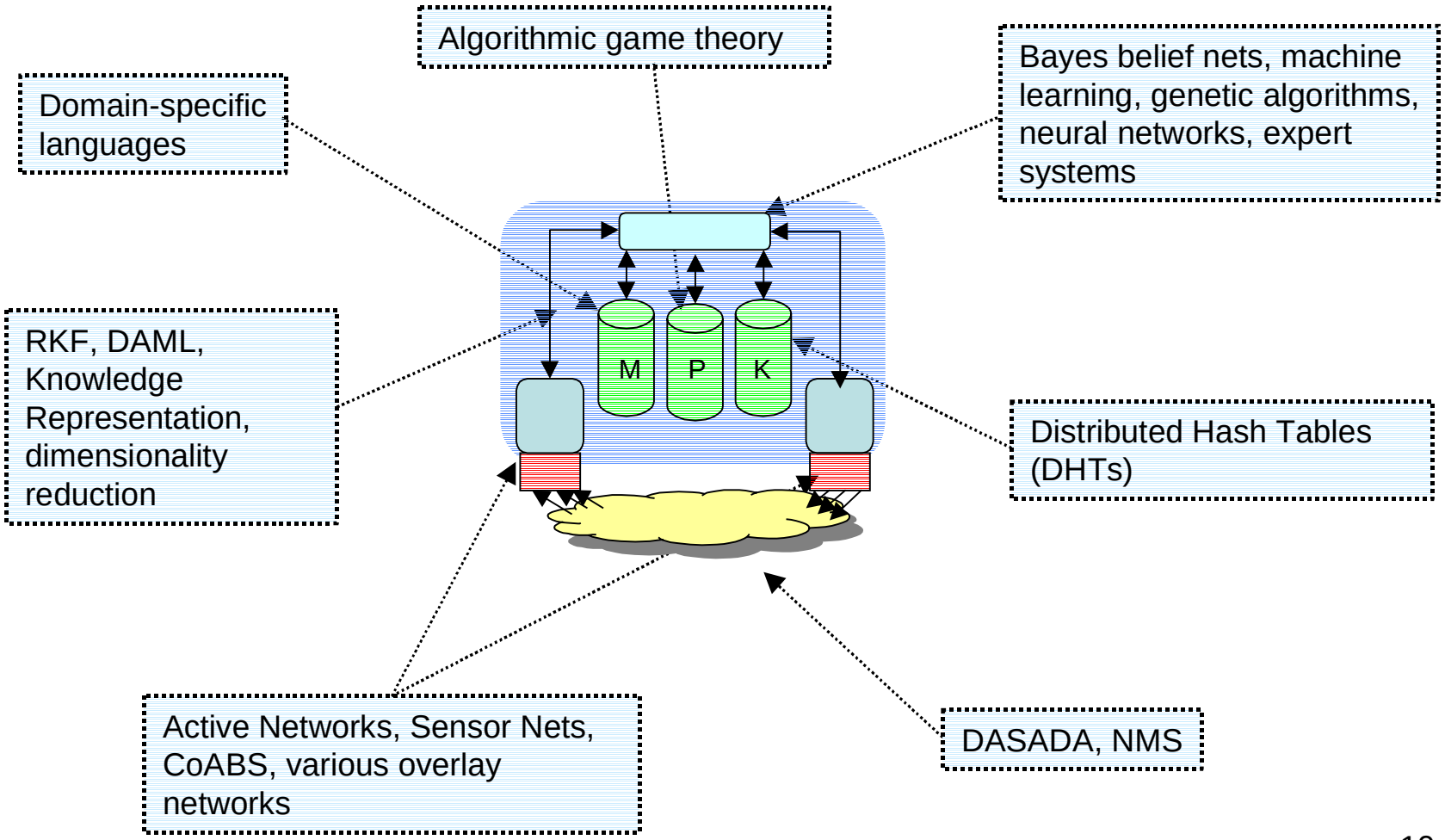
Exploring one aspect of the Knowledge Plane in detail

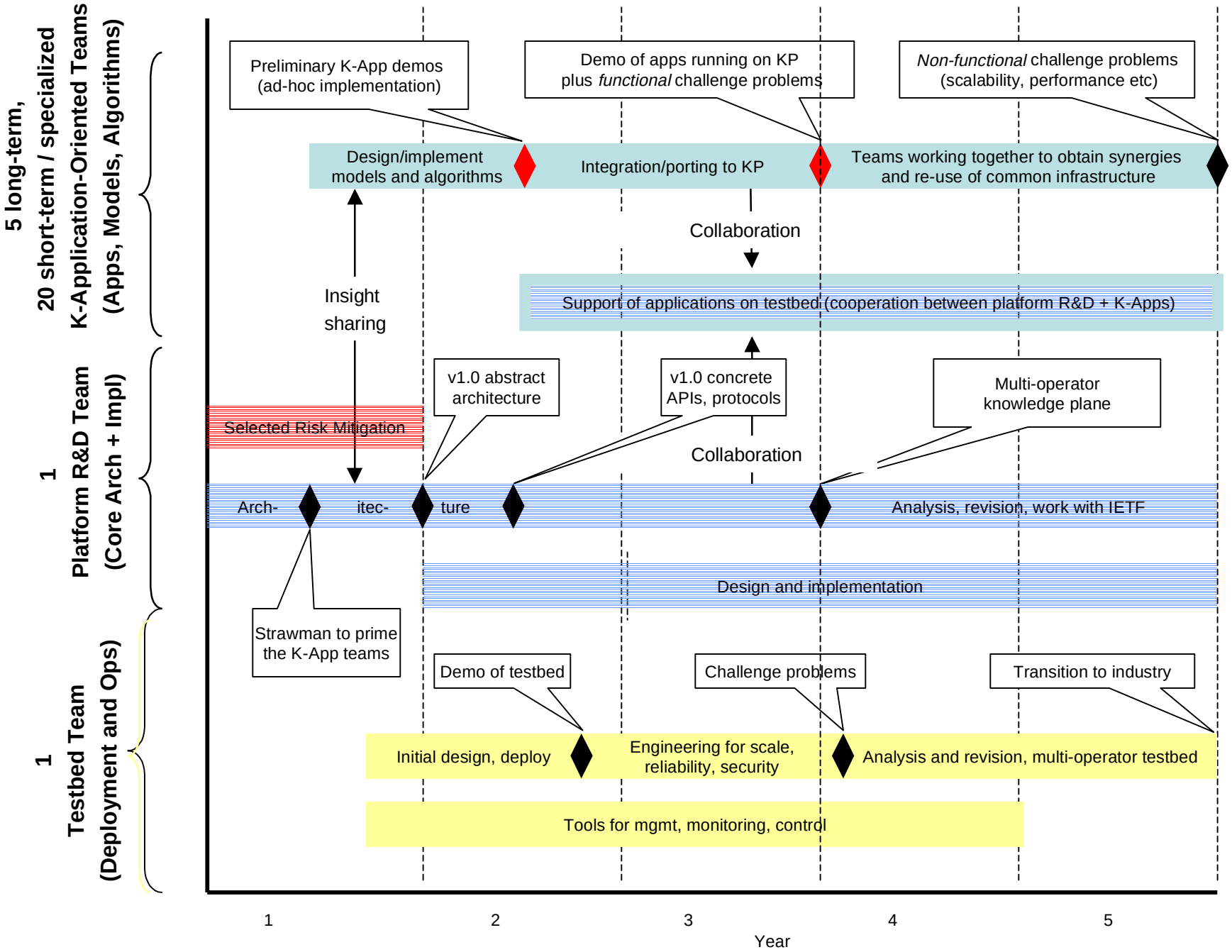
- The right knowledge at the right place at the right time
  - Route observations to cascade of “think points”
    - Aggregation and grade-based filtering.
  - Route explanations to observers and control points.
  - Route grades backwards, knowledge forward.
  - Pre-position explanations and responses
- 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



Technology Foundations

Why we know enough to embark on this project




5 long-term,
20 short-term / specialized
K-Application-Oriented Teams
(Apps, Models, Algorithms)
1
Platform R&D Team
(Core Arch + Impl)
1
Testbed Team
(Deployment and Ops)
Preliminary K-App demos
(ad-hoc implementation)
Demo of apps running on KP
plus functional challenge problems
Non-functional challenge problems
(scalability, performance etc)
Design/implement
models and algorithms
Integration/porting to KP
Teams working together to obtain synergies
and re-use of common infrastructure
Collaboration
Insight
sharing
Support of applications on testbed (cooperation between platform R&D + K-Apps)
v1.0 abstract
architecture
v1.0 concrete
APIs, protocols
Multi-operator
knowledge plane
Selected Risk Mitigation
Collaboration
Arch-
itec-
ture
Analysis, revision, work with IETF
Design and implementation
Strawman to prime
the K-App teams
Demo of testbed
Challenge problems
Transition to industry
Initial design, deploy
Engineering for scale,
reliability, security
Analysis and revision, multi-operator testbed
Tools for mgmt, monitoring, control
1
2
3
4
5
Year

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.

"Profession/Profile: George H. Heilmeier" article by Joshua Shapiro in _IEEE Spectrum_ 1994 June, as summarized at http://www.rdrop.com/~cary/html/creed.html#Heilmeier

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

Internet Strength	Corresponding Weakness	Illustrations
Successful focus on adaptive, resilient packet forwarding and routing	Goal of a unified, coherent "management space" has not yet been addressed	Existence of professional guilds for highly-trained network technicians. Routers configured individually. Un-integrated, ineffective proprietary management suites and standards.
Decentralized architecture	Applications and tools have only a local perspective	Fault isolation is very hard. Detecting worm propagation requires aggregated knowledge. Traffic engineering requires aggregated knowledge.
Multiple administrative domains	Inevitable stakeholder conflicts and failures to coordinate can be problematic	Flash crowds, operator error, new applications, hot potato routing and policy conflicts can cause problems that are hard to understand and address
Simple, transparent core and undifferentiated packets	No application-specific support	The Internet does not know what applications are running or what their individual needs are. Can't distinguish legitimate traffic from undesirable traffic

Example K-Application:

Fault detection/isolation/repair

Challenge Level	Action	Hard problems	Examples	Learning	How to evaluate progress
Quite Challenging (5 year)	Fault isolation and repair based on distributed observations and effects	Distinguishing between faults/attacks and benign patterns. Finding appropriate representations of the patterns. Using effectors across trust/administrative boundaries	Attack pushback across administrative boundaries	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)	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
Challenging (3 year)	Fault isolation based on active distributed observations	Scheduling/planning appropriate and efficient non-local observations	An extended "Why" application capable of e.g. diagnosing the loss of packets fragmented at a NAT point or routing problems across autonomous systems		
Challenging (18 month)	Fault isolation based on passive distributed	Fault isolation in general	"Why?" application		

Cognitive Networks Workshop

19-20 Nov. 2002, The Ritz-Carlton, Washington DC

- 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

- Netvision2012, Dec. 15-16, 2002. Dallas, TX
  - Significant focus on network configuration and management. Further discussions of the Knowledge Plane
- End-to-End Research Group, January 6-7, 2003. Berkeley, CA
  - 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

Challenge Level	Action	Hard problems	Examples	Learning	How to evaluate progress
Quite Challenging (5 year)	Multiple administrative domains, complex routing	Trust, policy negotiation, representation, plus all of the below	Configuring and managing a full ISP	[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	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.
Challenging (4 year)	Single administrative domain, complex protocols, complex internal topology	Dealing with policies, plus all of the below	Routed LAN with internal partitions / firewalls		
Challenging (3 year)	Single administrative domain, complex protocols	Representation of policies, goals, network abstractions. Deriving configuration both from environmental analysis and generative techniques based on representations	Routed LAN with multiple egress		
Moderate (2 year)	Single administrative domain, simple protocols		Unrouted LAN		

Example K-Application:

Reducing BGP convergence time

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

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

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

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)
- King (http://www.icir.org/vern/imw-2002/imw2002-papers/198.pdf), an infrastructure-free alternative to IDMaps

Example K-Application:

Routing observations / shadowing

Challenge Level	Action	Hard problems	Examples	Learning	How to evaluate progress
???	Route filtering	Determine whether an announcement can be discarded or not (to avoid problems ensuing from big routing tables)	Need global awareness to understand whether a filtered route is actually “covered” by another route	???	% reduction in routing table sizes
???	Inferring relationships between Ass based on routing	Making observations and inferences about routing paths, for instance that X is a customer of Y, or that X and Y are peers	(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	Need to “learn” difference between anomalies and legitimate quirks	Utility in avoiding provider errors. Increased efficiency and use of bandwidth in topology-aware applications
???	AS traceroute	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		???	% of hops in a traceroute that can be explained

Who isn't solving the problem

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

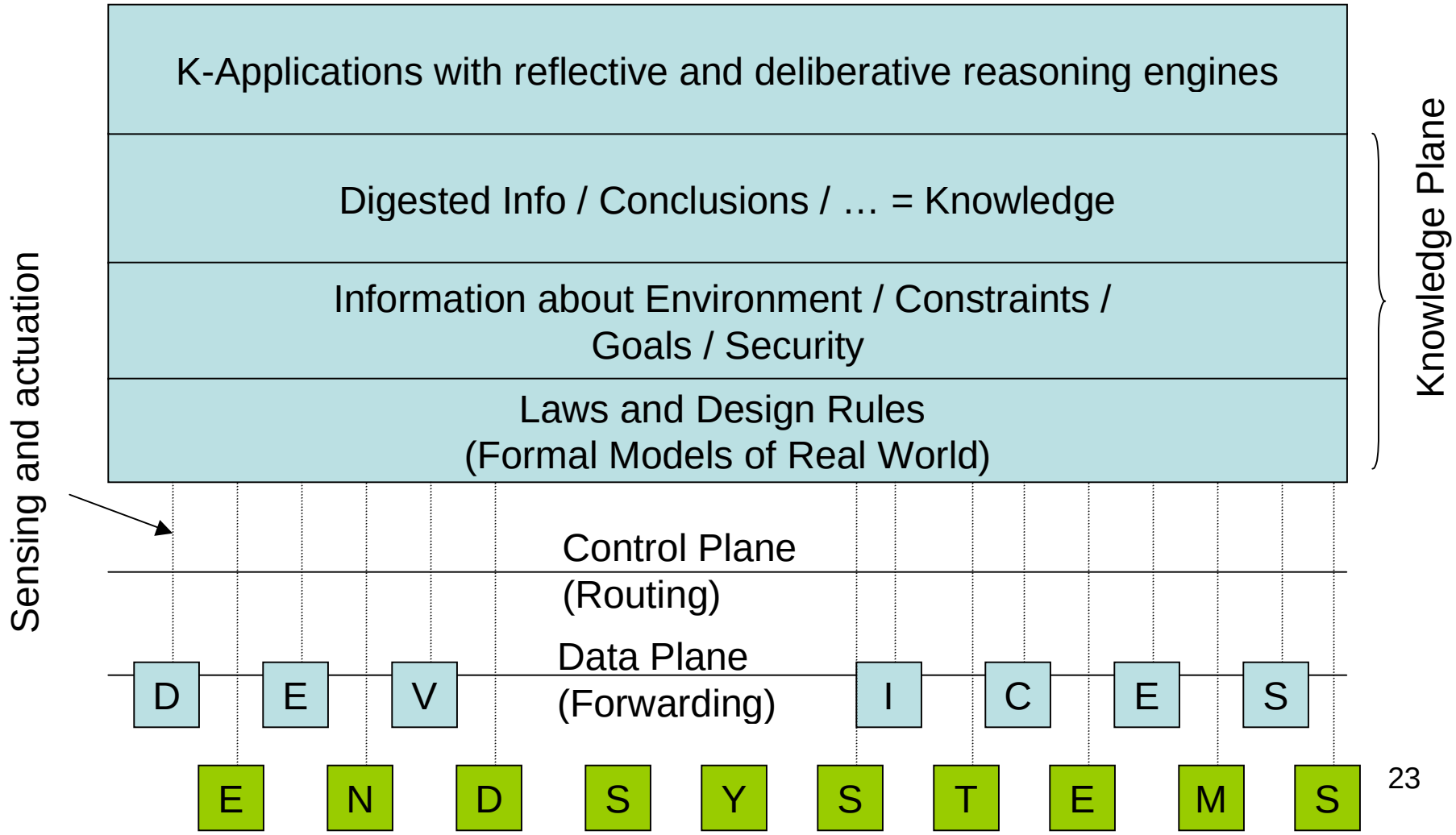
Gap: Open Cross-Vendor and Inter-Operator IP Network Management Support

*Q: is this characterization fair to FTN?

KP Security from the Start

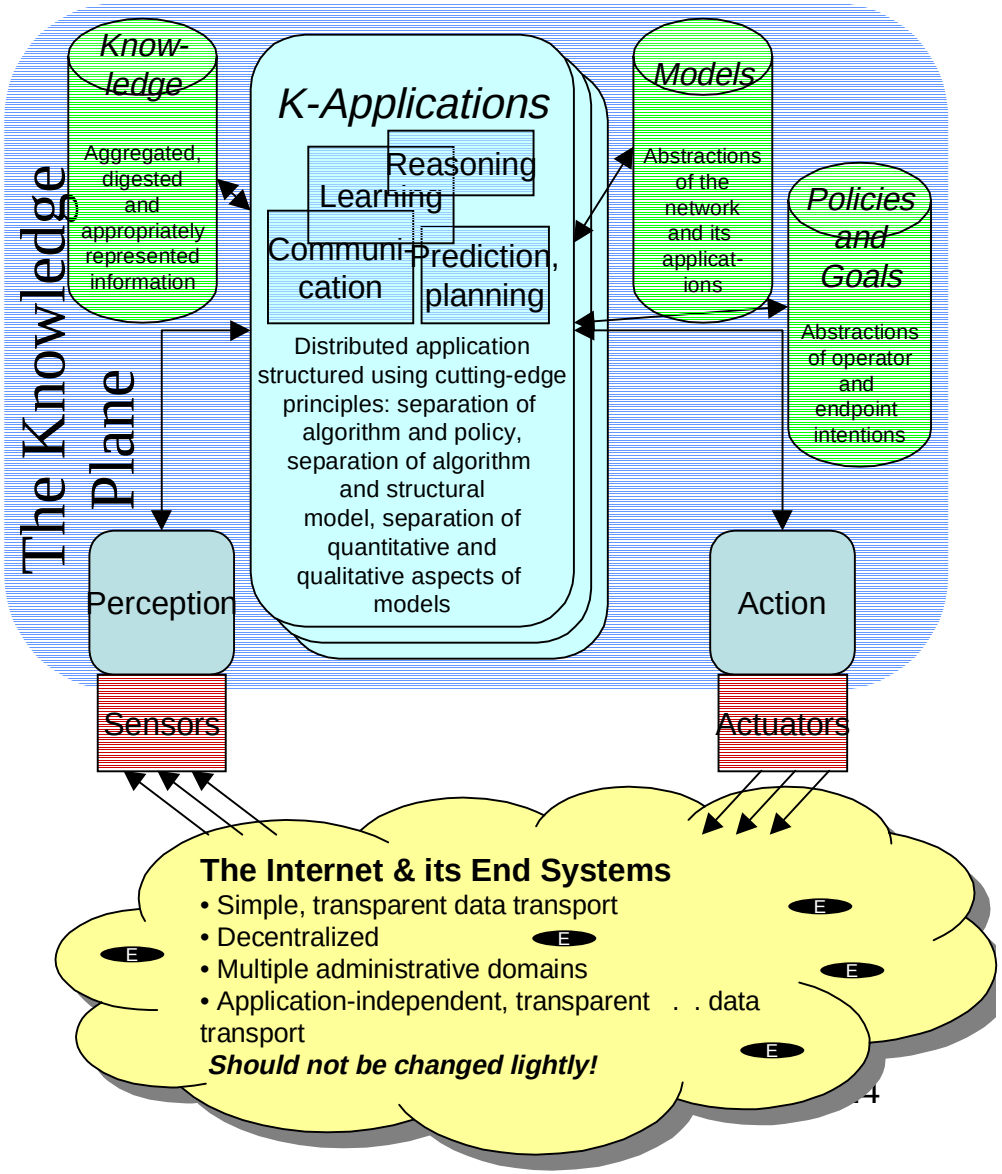
What needs to be built in from the start	Factors, challenges, approaches
Identity	Determining whether identity should be managed in parallel and independently from real-world identity
Authentication and trust	Observations need to be authenticated with varying degrees of confidence for different applications. 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
Algebras of trust	It will be necessary to confer/delegate authority at times. Emerging frameworks like SD3, keynote may prove a useful foundation
Accounting for malicious or mistaken observations	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

Another perspective on the KP



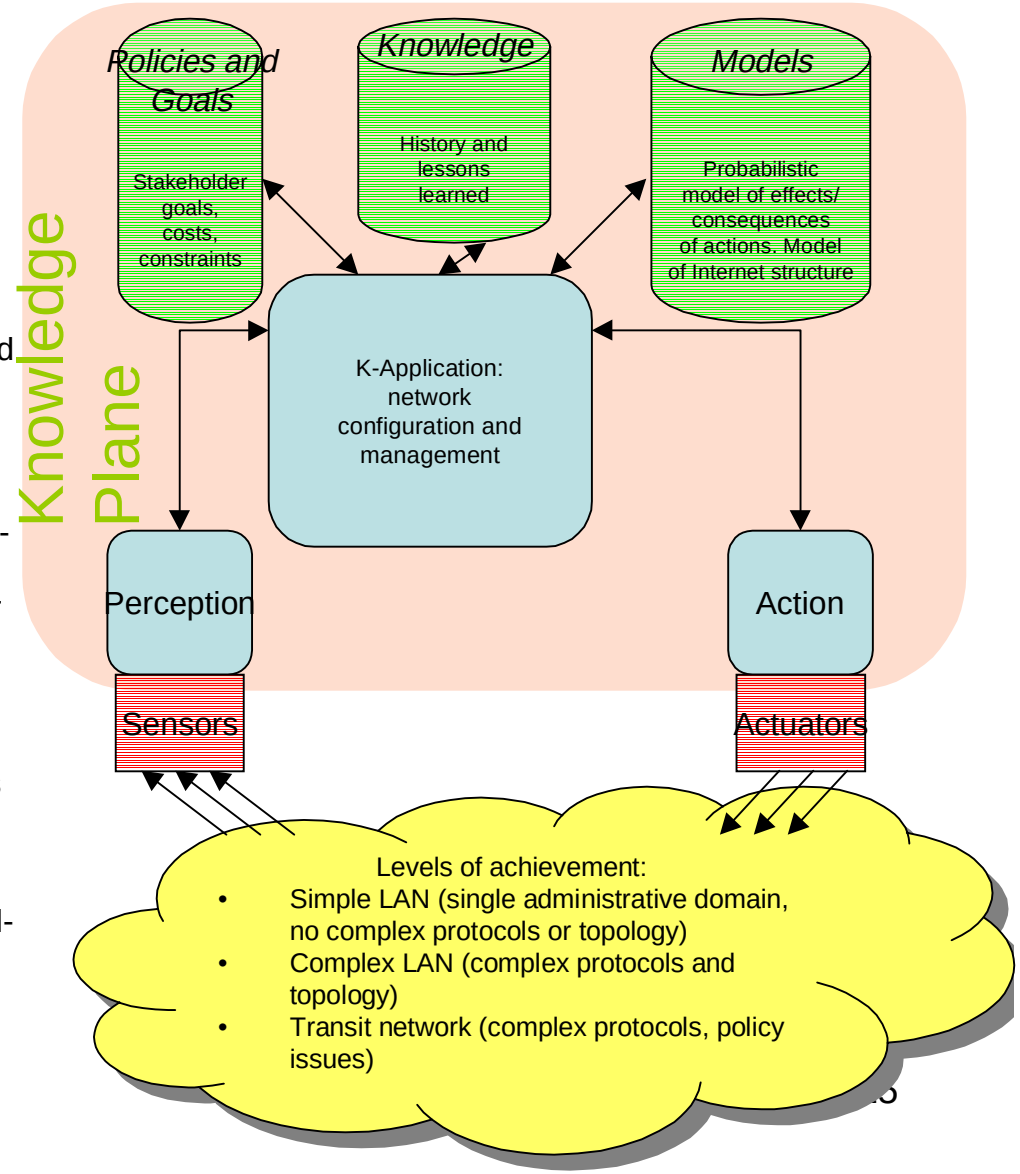
Why cognitive systems?

- 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



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 *a priori*
- Personal and commercial privacy