Usage Management In Multi-level Security Environments

Gregory Heileman and Christopher C. Lamb

Department of Electrical and Computer Engineering
University of New Mexico

April 4, 2012

Introduction

1. Information-centric Networking
2. System Overview
3. Theoretical Foundations
4. Implementation Overview
5. Final Steps

Information-centric Networking

Information-centric networking (ICN) is a new approach to internet-scale networks. These networks take advantage of data locality, cache data aggressively, decouple information providers from consumers, and use a content-centric perspective in network design

Similar conceptual approaches:

- **Named Data Objects** — Data objects are the primary data abstraction.
- **API Structure** — Programming interfaces are structured around requesting specific data objects. Can be synchronous or asynchronous.
- **Naming and Security** — Names are tightly and securely bound to content.
- **Caching** — Content is aggressively cached on nodes.

We are most concerned with the first and second characteristics currently.

Current Technology Fail

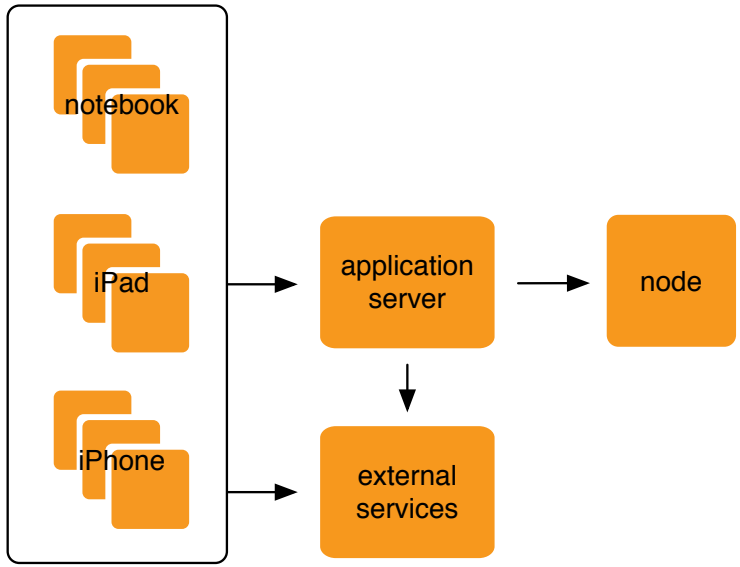
Current internet technologies don't dynamically protect data. Initial design assumtions and implementation characteristics make these approaches difficult.

Why does the internet fail?

- **Strict Layering** — Routing and switching are generally lower level operations (layers 2 and 3 in the OSI model). Content-sensitive routing is a layer 7 (application layer) operation, requiring very expensive hardware to do well.
- **End-to-End Arguments** — Network cores are simple, fast, and dumb. They need to brighten up to evanluate information suitability.
- **Packetization** — Policies and content requires multiple packets. This implies complex window retention logic to handle context splitting, and in at least half of typical cases, context splitting can't be handled at all.

The principles have been effective, but in some cases services *should* be in the network core.

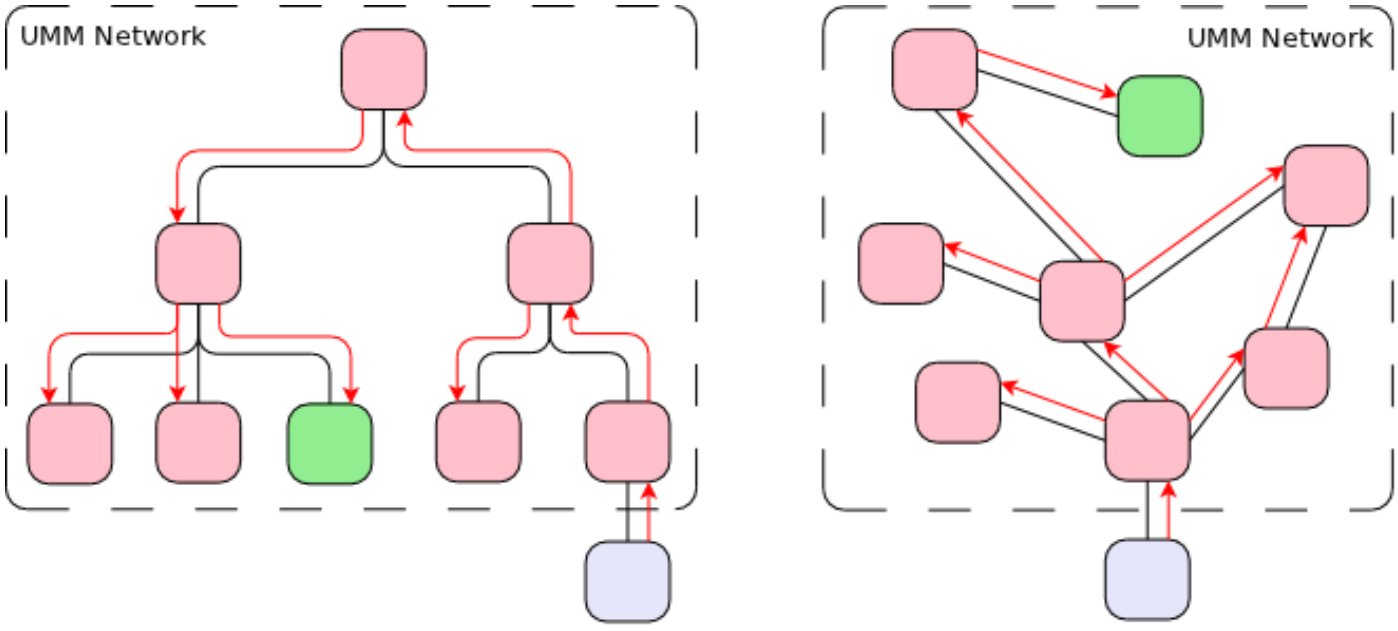
System Overview — Device Perspective



How does the system work?

1. **Request** — An initial request is submitted from an edge device
2. **Reciept** — The request is received by an application server that has access to ICN services via an ICN node and external services of some kind as well.
3. **Dispatch** — The request for information if suitable is dispatched into the ICN.

System Overview — Into the network

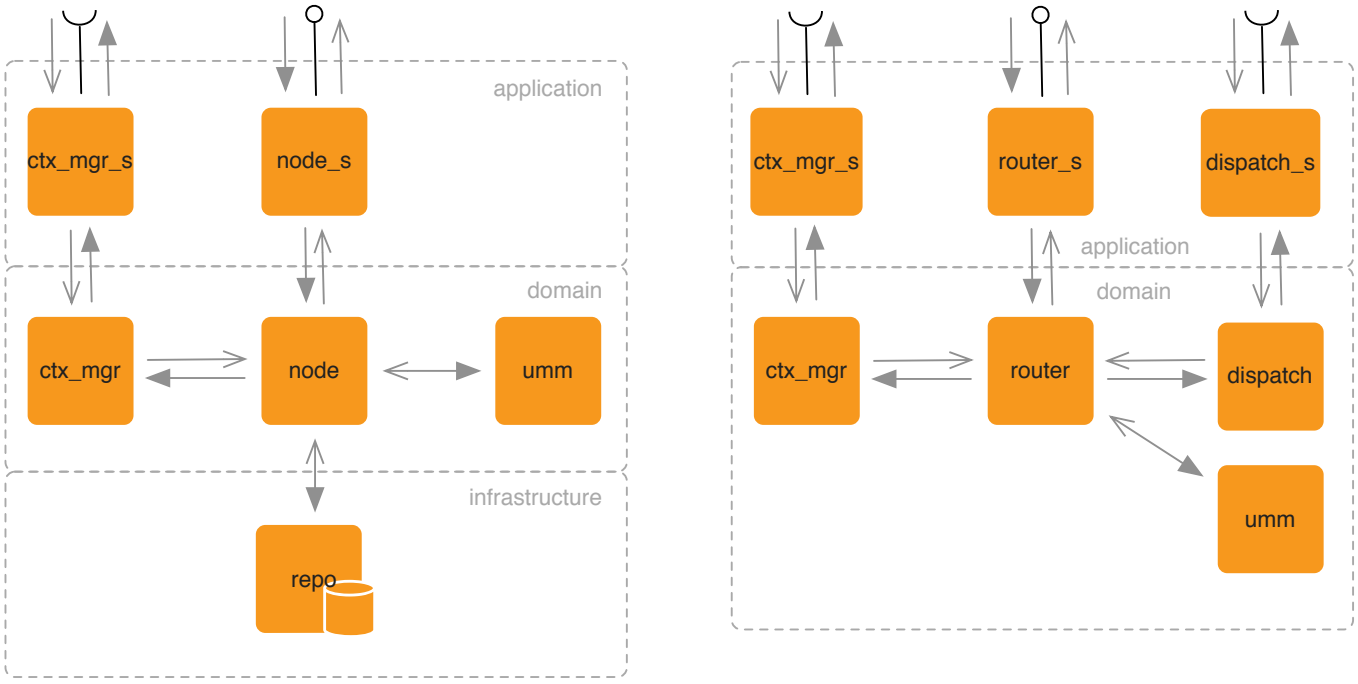


(A) Hierarchical Network (B) Non-hierarchical Network

How is information dispatched through the network? *Well, it depends on the network.*

1. **Submission** — Submit to a node.
2. **Transmission** — Pass through the network, either via a router (if hierarchical) or known peers (non-hierarchical).
3. **Respond** — Nodes respond with content if possible.

System Overivew — Nodes and Routers



(a) Hierarchical Node

(b) Hierarchical Router

Strictly layered architecture within nodes:

- **Application** — REST Interfaces, network access
- **Domain** — Domain functionality, specific node/router logic and UM
- **Infrastructure** — Data storage repositories, data objects and policies

Distributed Security Decisions

In order to make security decions, we must know the *environment* of the *resource* (the data object) and the *subject* (the user).

Fundamental Foundations

In order to understand that our decisions are valid and maintain security in distributed *environments*, we must *prove* that local security decisions provide a compliant global solution.

We make *greedy* with respect to security, and for efficiency we use *dynamic programming*. For these to provide a globally optimal solution, we need to show that the routing problem exibits *optimal structure* and *overlapping subproblems*.

Optimal Substructure

Overlapping Subproblems

Information Protection

Implementation Overview — Environment

Implementation Overview — Capistrano

Implementation Overview — Rest API

Implementation Overview — Information Flow

Implementation Overview — Usage Domain

Implementation Overview — Policy Language

Next Steps