Observations regarding a new architecture

Kevin Fall
Intel Research, Berkeley
18-Sep-2006, Cambridge, UK
Naming

• Things: objects, services, people, roles
• Types of names
  – descriptive (associative) names – “content”
  – location-related names
• Names mapped to routable tag
• Approach using standard namespaces
  – some can be locally computed
  – general names are variable length
  – hierarchical and flat are both useful
Naming (example)

- URI structure
  - \(<namespace> : <ns-specific part>\)
  - http://www.cnn.com
  - isbn:81-7808-101-6

- Can hierarchically decompose as required
- Easily mapped to finite ID using hash
  - if flat routing supported, enough (later)
- Naturally works with anycast
Provenance & Tags

• Generally care less about what entity provided an object than what entity authored it & what it is

• Provenance/tags for objects:
  – verifiable origin and modification lineage
  – content identification
  – handling/dissemination restrictions
    • IP security labels
    • DRM
Addressing & Routing

• “Address” ~ “routing tag”
• $f(name) \rightarrow address$, f may be identity map
• Multiple routing types
  – on names (dns, email, sip?)
  – on variable-len numbers (phone network)
  – on fixed-len numbers (Internet)
• High-level routing requirements
  – operation on given graph [ie. not pure overlay]
  – adjustable using traffic engineering techniques
  – supports expressible policy
• Non-requirements
  – stretch-1 operation
  – hierarchy in addresses
  – only 1 destination address in an ADU
Current Routing Items

• Compact Routing
  – small $O(\sqrt{n})$ tables but not always stretch1
  – extended to topology independence
  – good for scale-free graphs
  – nothing in terms of dynamics

• Overlays that know about underlying graph (VRR)
  – inspired by overlays
  – operates on underlying graph
  – flat space

• Policies for routing
  – algebraic forms

• MANET routing
  – proactive & reactive
  – IP-IP encaps

• Mobile IP
Transport

• Local case: network layer optional
• Fragmentation for two reasons
  – adapting to MTU (current)
  – performance (current for TCP somewhat)
• Application chooses between
  – uncorrected / FEC / ARQ reliability path
  – network coding?
• Storage in network
  – some nodes have more reliable storage
  – can be used to, e.g., offload end node faster
Networking & Storage

• Network hidden behind storage
  – done by AFS / Coda
  – somewhat awkward if interrupted
  – awkward for streaming

• Network hidden behind procedures
  – common RPC services
  – awkward if interrupted
  – awkward for streaming

• Network should understand two types
  – streaming and object ~ ADUs ~ DTN bundles
  – objects mesh with store-and-forward

• *store and forward with support for long-term storage*
ADUs

- ADUs ~ DTN bundles
  - like DTN bundles
- Main features of DTN bundles
  - variable-length src/destination
  - origin time and useful life (time must be sync’d)
  - class of service
  - fragmentation
  - extensions
  - segregation of mutable and non-mutable headers
Security

• Authentication of provenance
  – digital signatures (e.g. IBE) [worry: keygen]
• Protection from transmission disclosure
• Management of unwanted traffic
  – assignment of traffic engineering descriptor
    – ingress filtering of TE descriptor
• Secure notion of time
Layers

• Implementation technique ~ served us well
  – with limited set of protocols
  – and easily ‘abstractable’ link layer

• Issues with layering
  – wrong abstraction (gives rise to tunnels)
  – bad cross-layer interactions
    • ATM cell loss; IP fragmentation; TCP MSS issues; content splitting