15-744: Computer Networking

L-15 Changing the Network

Adding New Functionality to the Internet

- Overlay networks
- Active networks
- Assigned reading
  - Active network vision and reality: lessons from a capsule-based system
  - Delay tolerant networks (pages 1-4)
- Optional reading
  - Resilient Overlay Networks

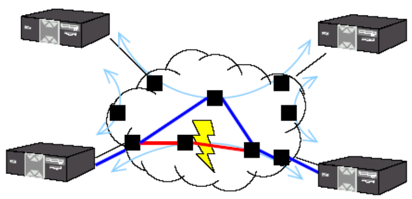
Clean-Slate vs. Evolutionary

- Successes of the 80s followed by failures of the 90's
  - IP Multicast
  - QoS
  - RED (and other AQMs)
  - ECN
  - …
- Concern that Internet research was dead
  - Difficult to deploy new ideas
  - What did catch on was limited by the backward compatibility required

Outline

- Overlay Routing
  - Overview
  - RON: Slides Akshay
- Active Networks
- Delay tolerant networks

The Internet Ideal



- Dynamic routing routes around failures
- End-user is none the wiser

Lesson from Routing Overlays

End-hosts are often better informed about performance, reachability problems than routers.

- End-hosts can measure path performance metrics on the (small number of) paths that matter
- Internet routing *scales well*, but at the cost of performance

Overlay Routing

- Basic idea:
  - Treat multiple hops through IP network as one hop in “virtual” overlay network
  - Run routing protocol on overlay nodes
- Why?
  - For performance and resilience – can run more clever protocol on overlay to avoid congested links and failures → RON
  - For functionality – can provide new features such as multicast, active processing, IPv6 → DTN, active nets

Overlay for Features

- How do we add new features to the network?
  - Does every router need to support new feature?
  - Choices
    - Reprogram all routers → active networks
    - Support new feature within an overlay
  - Basic technique: tunnel packets
- Tunnels
  - IP-in-IP encapsulation
  - Poor interaction with firewalls, multi-path routers, etc.

Examples

- IP V6 & IP Multicast
  - Tunnels between routers supporting feature
- Mobile IP
  - Home agent tunnels packets to mobile host's location
- QOS
  - Needs some support from intermediate routers → maybe not?

9

Outline

- Overlay Routing
  - Overview
  - RON: Slides Akshay
- Active Networks
- Delay tolerant networks

10

How Robust is Internet Routing?

- Slow outage detection and recovery
- Inability to detect badly performing paths
- Inability to efficiently leverage redundant paths
- Inability to perform application-specific routing
- Inability to express sophisticated routing policy

Paxson 95-97	• 3.3% of all routes had serious problems
Labovitz 97-00	• 10% of routes available < 95% of the time • 65% of routes available < 99.9% of the time • 3-min minimum detection+recovery time; often 15 mins • 40% of outages took 30+ mins to repair
Chandra 01	• 5% of faults last more than 2.75 hours

11

Routing Convergence in Practice

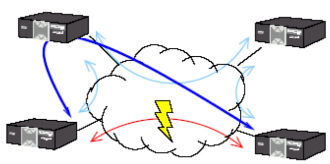
Time	Prefix	Type	AS Path	Localpref	MED	Community
2005/11/01 00:06:23	195.78.38.0/23	A	174 5400 20703 28773			174:21100 16631:1000
2005/11/01 00:06:39	195.78.38.0/23	A	3356 5400 20703 28773			3356:2 3356:100 3356:123 3356:500 3356:2064 5400:46
2005/11/01 00:06:45	195.78.38.0/23	W				

- Route withdrawn, but stub cycles through backup path…

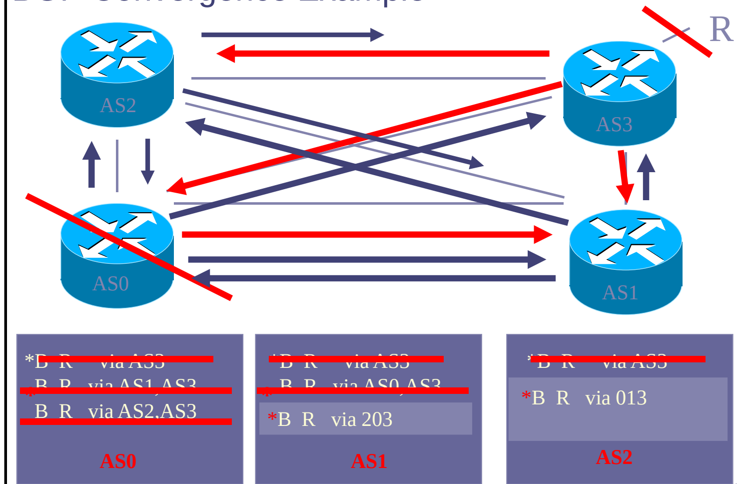
12

Resilient Overlay Networks: Goal

- Increase reliability of communication for a small (i.e., < 50 nodes) set of connected hosts
- Main idea: End hosts discover network-level path failure and cooperate to re-route.



BGP Convergence Example

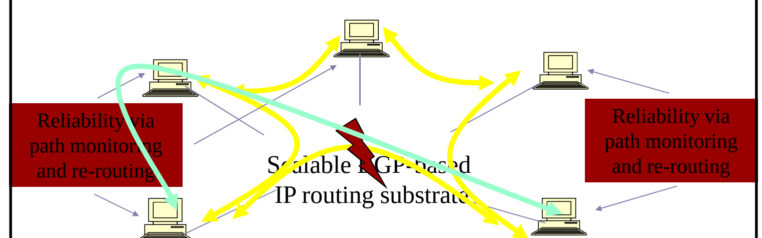


The RON Architecture

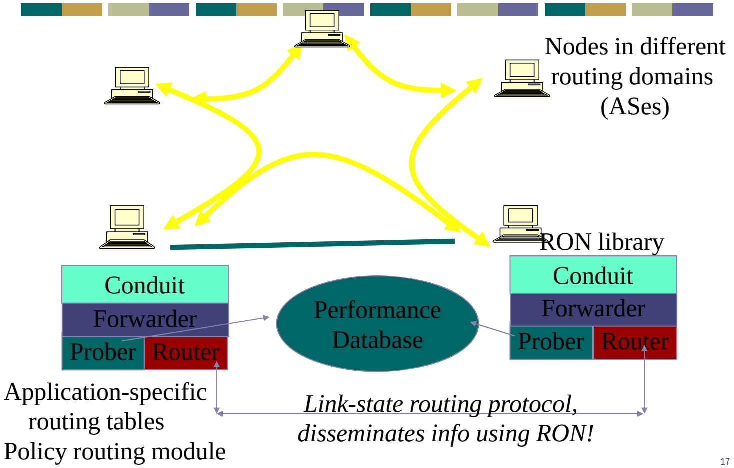
- Outage detection
  - Active UDP-based probing
    - Uniform random in [0,14]
    - $O(n^2)$
  - 3-way probe
    - Both sides get RTT information
    - Store latency and loss-rate information in DB
- Routing protocol: Link-state between overlay nodes
- Policy: restrict some paths from hosts
  - E.g., don't use Internet2 hosts to improve non-Internet2 paths

RON: Routing Using Overlays

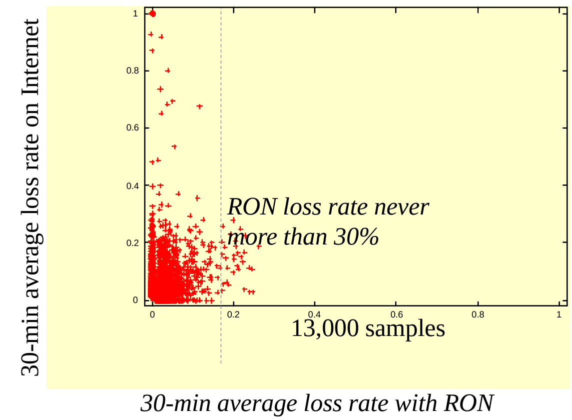
- Cooperating end-systems in different routing domains can conspire to do better than scalable wide-area protocols
- Types of failures
  - Outages: Configuration/op errors, software errors, backhoes, etc.
  - Performance failures: Severe congestion, DoS attacks, etc.



RON Design



RON greatly improves loss-rate



An order-of-magnitude fewer failures

30-minute average loss rates

Loss Rate	RON Better	No Change	RON Worse
10%	479	57	47
20%	127	4	15
30%	32	0	**0**
50%	20	0	**0**
80%	14	0	**0**
100%	10	0	**0**

6,825 “path hours” represented here
12 “path hours” of essentially complete outage
76 “path hours” of TCP outage
RON routed around all of these!
One indirection hop provides almost all the benefit!

Main results

- RON can route around failures in ~ 10 seconds
- Often improves latency, loss, and throughput
- Single-hop indirection works well enough
  - Motivation for another paper (SOSR)
  - Also begs the question about the benefits of overlays

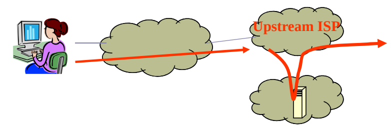
Open Questions

- Scaling
  - Probing can introduce high overheads
  - Can use a subset of $O(n^2)$ paths → but which ones?
- Interaction of multiple overlays
  - End-hosts observe qualities of end-to-end paths
  - Might multiple overlays see a common “good path”
  - Could these multiple overlays interact to create increase congestion, oscillations, etc.?
    - Selfish routing

21

Efficiency

- Problem: traffic must traverse bottleneck link both inbound and outbound

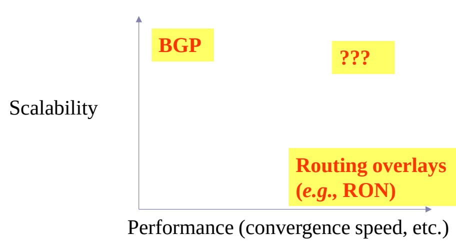


- Solution: in-network support for overlays
  - End-hosts establish reflection points in routers
    - Reduces strain on bottleneck links
    - Reduces packet duplication in application-layer multicast (next lecture)

22

Scaling

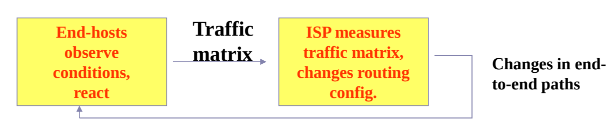
- Problem: $O(n^2)$ probing required to detect path failures. Does not scale to large numbers of hosts.
- Solution: ?
  - Probe some subset of paths (which ones)
  - Is this any different than a routing protocol, one layer higher?



23

Interaction of Overlays and IP Network

- Supposed outcry from ISPs: “Overlays will interfere with our traffic engineering goals.”
  - Likely would only become a problem if overlays became a significant fraction of all traffic
  - Control theory: feedback loop between ISPs and overlays
  - Philosophy/religion: Who should have the final say in how traffic flows through the network?



24

Benefits of Overlays

- Access to multiple paths
  - Provided by BGP multihoming
- Fast outage detection
  - But…requires aggressive probing; doesn't scale

Question: What benefits does overlay routing provide over traditional multihoming + intelligent routing selection

Outline

- Overlay Routing (RON)
- Active Networks
- Delay tolerant networks

Why Active Networks?

- Traditional networks route packets looking only at destination
  - Also, maybe source fields (e.g. multicast)
- Problem
  - Rate of deployment of new protocols and applications is too slow
- Solution
  - Allow computation in routers to support new protocol deployment

Active Networks

- Nodes (routers) receive packets:
  - Perform computation based on their internal state and control information carried in packet
  - Forward zero or more packets to end points depending on result of the computation
- Users and applications can control behavior of the routers
- End result: network services richer than those by the simple IP service model

Why not IP?

- Applications that do more than IP forwarding
  - Firewalls
  - Web proxies and caches
  - Transcoding services
  - Nomadic routers (mobile IP)
  - Transport gateways (snoop)
  - Reliable multicast (lightweight multicast, PGM)
  - Online auctions
  - Sensor data mixing and fusion
- Active networks makes such applications easy to develop and deploy

29

Variations on Active Networks

- Programmable routers
  - More flexible than current configuration mechanism
  - For use by administrators or privileged users
- Active control
  - Forwarding code remains the same
  - Useful for management/signaling/measurement of traffic
- “Active networks”
  - Computation occurring at the network (IP) layer of the protocol stack → capsule based approach
  - Programming can be done by any user
  - Source of most active debate

30

Case Study: MIT ANTS System

- Conventional Networks:
  - All routers perform same computation
- Active Networks:
  - Routers have same runtime system
- Tradeoffs between functionality, performance and security

31

System Components

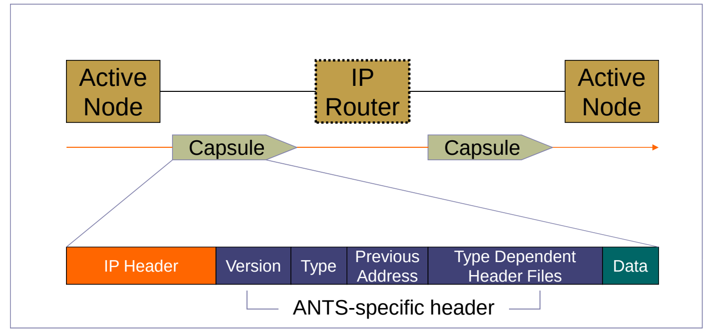
- Capsules
- Active Nodes:
  - Execute capsules of protocol and maintain protocol state
  - Provide capsule execution API and safety using OS/language techniques
- Code Distribution Mechanism
  - Ensure capsule processing routines automatically/dynamically transfer to node as needed

32

Capsules

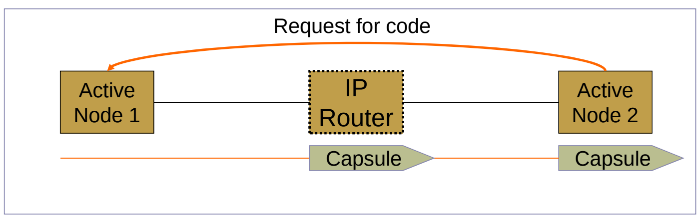
- Each user/flow programs router to handle its own packets
  - Code sent along with packets
  - Code sent by reference
- Protocol:
  - Capsules that share the same processing code
- May share state in the network
- Capsule ID (i.e. name) is MD5 of code

Capsules



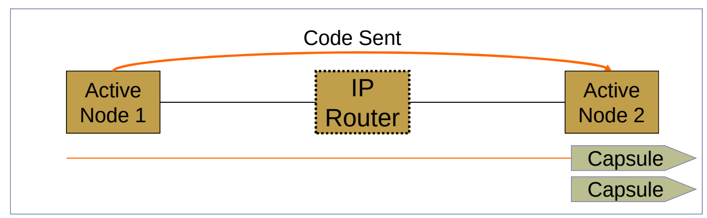
- Capsules are forwarded past normal IP routers

Capsules



- When node receives capsule uses “type” to determine code to run
- What if no such code at node?
  - Requests code from “previous address” node
  - Likely to have code since it was recently used

Capsules



- Code is transferred from previous node
  - Size limited to 16KB
  - Code is signed by trusted authority (e.g. IETF) to guarantee reasonable global resource use

Research Questions

- Execution environments
  - What can capsule code access/do?
- Safety, security & resource sharing
  - How isolate capsules from other flows, resources?
- Performance
  - Will active code slow the network?
- Applications
  - What type of applications/protocols does this enable?

Functions Provided to Capsule

- Environment Access
  - Querying node address, time, routing tables
- Capsule Manipulation
  - Access header and payload
- Control Operations
  - Create, forward and suppress capsules
  - How to control creation of new capsules?
- Storage
  - Soft-state cache of app-defined objects

Safety, Resource Mgt, Support

- Safety:
  - Provided by mobile code technology (e.g. Java)
- Resource Management:
  - Node OS monitors capsule resource consumption
- Support:
  - If node doesn't have capsule code, retrieve from somewhere on path

Applications/Protocols

- Limitations
  - Expressible → limited by execution environment
  - Compact → less than 16KB
  - Fast → aborted if slower than forwarding rate
  - Incremental → not all nodes will be active
- Proof by example
  - Host mobility, multicast, path MTU, Web cache routing, etc.

Discussion

- Active nodes present lots of applications with a desirable architecture
- Key questions
  - Is all this necessary at the forwarding level of the network?
  - Is ease of deploying new apps/services and protocols a reality?

41

Outline

- Active Networks
- Overlay Routing (RON)
- Delay tolerant networks
  - Slides Abhinava

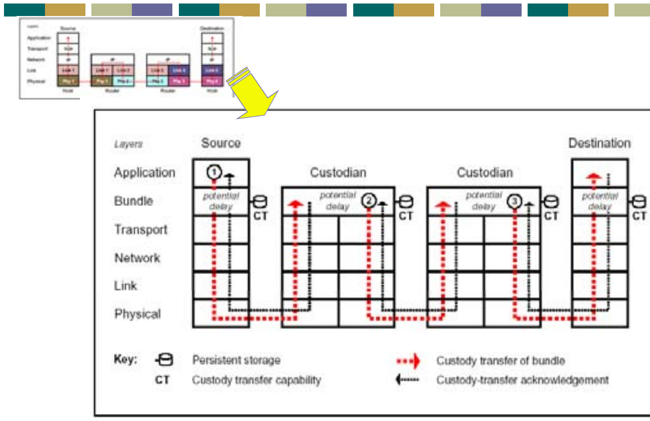
42

Delay-Tolerant Networking Architecture

- Goals
  - Support interoperability across 'radically heterogeneous' networks
  - Tolerate delay and disruption
    - Acceptable performance in high loss/delay/error/disconnected environments
    - Decent performance for low loss/delay/errors
- Components
  - Flexible naming scheme
  - Message abstraction and API
  - Extensible Store-and-Forward Overlay Routing
  - Per-(overlay)-hop reliability and authentication

43

Disruption Tolerant Networks



44

Disruption Tolerant Networks

Naming Data (DTN)

- Endpoint IDs are processed as names
  - refer to one or more DTN nodes
  - expressed as Internet URI, matched as strings
- URIs
  - Internet standard naming scheme [RFC3986]
  - Format: <scheme> : <SSP>
- SSP can be arbitrary, based on (various) *schemes*
- More flexible than DOT/DONA design but less secure/scalable
  - Data-centric networking approaches iscussed later in the course

Naming

- Support ‘radical heterogeneity’ using URI’s:
  - {scheme ID (allocated), scheme-specific-part}
  - associative or location-based names/addresses optional
  - Variable-length, can accommodate “any” net’s names/addresses
- Endpoint IDs:
  - multicast, anycast, unicast
- Late binding of EID permits naming flexibility:
  - EID “looked up” only when necessary during delivery
  - contrast with Internet lookup-before-use DNS/IP

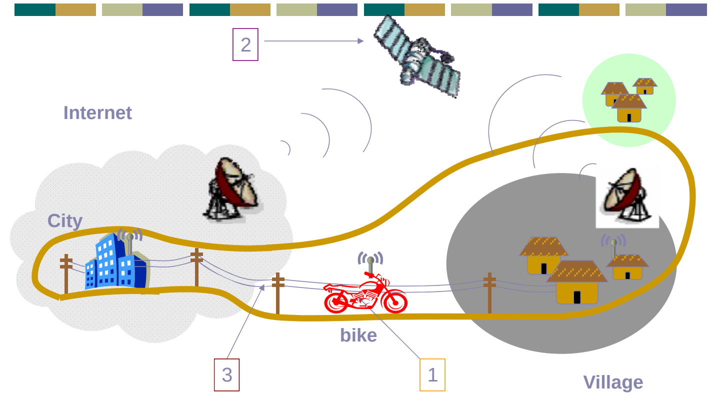
Message Abstraction

- Network protocol data unit: <u>bundles</u>
  - “postal-like” message delivery
  - coarse-grained CoS [4 classes]
  - origination and useful life time [assumes sync’d clocks]
  - source, destination, and respond-to EIDs
  - *Options*: return receipt, “traceroute”-like function, alternative reply-to field, custody transfer
  - fragmentation capability
  - overlay atop TCP/IP or other (link) layers [layer ‘agnostic’]
- Applications send/receive <u>messages</u>
  - “Application data units” (**<u>ADUs</u>**) of possibly-large size
  - Adaptation to underlying protocols via ‘convergence layer’
  - API includes persistent registrations

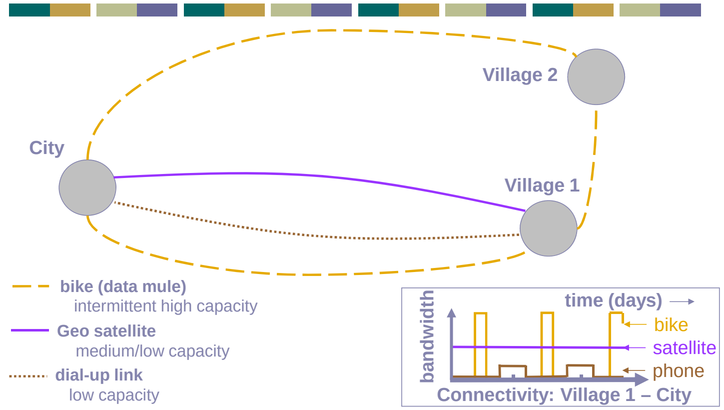
DTN Routing

- DTN Routers form an overlay network
  - only selected/configured nodes participate
  - nodes have persistent storage
- DTN routing topology is a **time-varying** multigraph
  - Links come and go, sometimes predictably
  - Use any/all links that can possibly help (multi)
  - Scheduled, Predicted, or Unscheduled Links
    - May be direction specific [e.g. ISP dialup]
    - May learn from history to predict schedule
- Messages fragmented based on dynamics
  - Proactive fragmentation: optimize contact volume
  - Reactive fragmentation: resume where you failed

Example Routing Problem



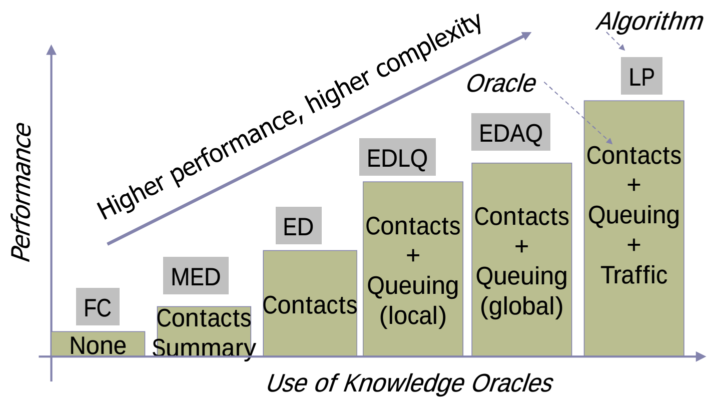
Example Graph Abstraction



The DTN Routing Problem

- *Inputs*: topology (multi)graph, vertex buffer limits, contact set, message demand matrix (w/priorities)
- An ***edge*** is a possible opportunity to communicate:
  - One-way: (S, D, c(t), d(t))
  - (S, D): source/destination ordered pair of contact
  - c(t): capacity (rate); d(t): delay
  - A ***Contact*** is when c(t) > 0 for some period $[i_k, i_{k+1}]$
- Vertices have buffer limits; edges in graph if ever in any contact, multigraph for multiple physical connections
- *Problem*: optimize some metric of delivery on this structure
  - Sub-questions: what metric to optimize?, efficiency?

Knowledge-Performance Tradeoff



Knowledge-Performance Tradeoff

Abbr.	Name	Description	Oracles Used
FC	First Contact	Use any available contact	None
MED	Minimum Expected Delay	Dijkstra with time-invariant edge costs based on average edge waiting time	Contacts Summary
ED	Earliest Delivery	Modified Dijkstra with time-varying cost function based on waiting time	Contacts
EDLQ	Earliest Delivery with Local Queue	ED with cost function incorporating local queuing	Contacts
EDAQ	Earliest Delivery with All Queue	ED with cost function incorporating queuing information at all nodes and using reservations	Contacts and Queuing
LP	Linear Program	-	Contacts, Queuing and Traffic

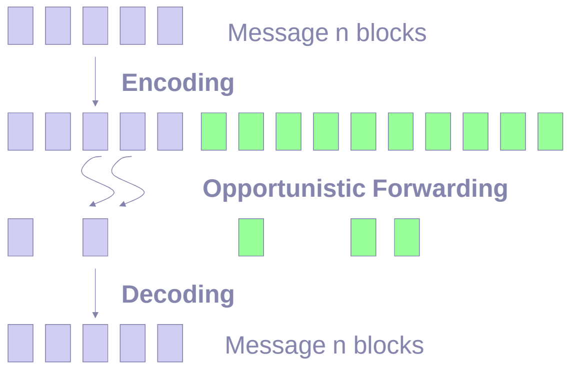
Routing Solutions - Replication

- "Intelligently" distribute identical data copies to contacts to increase chances of delivery
  - Flooding (unlimited contacts)
  - Heuristics: random forwarding, history-based forwarding, predication-based forwarding, etc. (limited contacts)
- Given "replication budget", this is difficult
  - Using simple replication, only finite number of copies in the network [Juang02, Grossglauser02, Jain04, Chaintreau05]
  - Routing performance (delivery rate, latency, etc.) heavily dependent on *"deliverability" of these contacts* (or *predictability of heuristics)*
  - No single heuristic works for all scenarios!

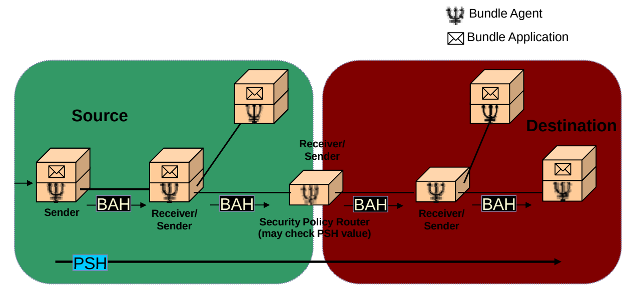
Using Erasure Codes

- Rather than seeking particular "good" contacts, "split" messages and distribute to more contacts to increase chance of delivery
  - Same number of bytes flowing in the network, now in the form of coded blocks
  - Partial data arrival can be used to reconstruct the original message
    - Given a replication factor of *r,* (in theory) any *1/r* code blocks received can be used to reconstruct original data
  - Potentially leverage more contacts opportunity that result in lowest worse-case latency
- Intuition:
  - Reduces "risk" due to outlier bad contacts

Erasure Codes



DTN Security



- Payload Security Header (PSH) end-to-end security header
- Bundle Authentication Header (BAH) hop-by-hop security header

credit: MITRE

So, is this just e-mail?

	naming/ late binding	routing	flow contrl	multi-app	security	reliable delivery	priority
e-mail	Y	N (static)	N(Y)	N(Y)	opt	Y	N(Y)
DTN	Y	Y (exten)	Y	Y	opt	opt	Y

- Many similarities to (abstract) e-mail service
- Primary difference involves routing, reliability and security
- E-mail depends on an underlying layer's routing:
  - Cannot generally move messages 'closer' to their destinations in a partitioned network
  - In the Internet (SMTP) case, not disconnection-tolerant or efficient for long RTTs due to "chattiness"
- E-mail security authenticates only user-to-user

Announcements

- Project report due on Friday – see e-mail
  - Next: project meetings week before Thanksgiving
  - Also: please use office hours!
- Midterms have been graded
  - Median is 69
  - Grade under 60: work on material!
  - Also: pay attention to homeworks!