Disruption/Delay-Tolerant Networking (DTN) Tutorial

Kevin Fall, PhD
Qualcomm*, Inc.
kfall@qualcomm.com

http://WWW.DTNRG.ORG

Nov 30, 2011 / GFIS 201 / Seoul, South Korea
(this material originally created prior to author's affiliation with Qualcomm)*

What is DTN?

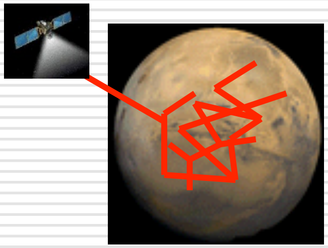
- Network/protocol architecture (in the Internet TCP/IP sense)
  - Not “where do I deploy routers and switches” sense
- Can use TCP/IP for transport, but doesn’t have to
- Some tenets
  - Tolerate very long end-to-end delays and disruption
  - Support more than one (simultaneously-operating) name space
  - Use store-carry-forward (data can be physically moved) of objects
  - Understand that channel and object security are different
- An R&D area
  - Mostly in the government sector so far
  - As an IRTF group, leverages IETF processes & procedures

Outline

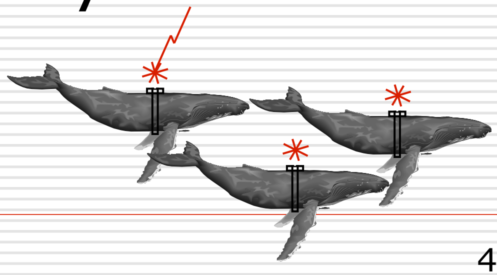
- ***Introduction: The Internet and Challenged Networks***
- *The DTN Architecture*
- *DTN People & Projects*
- *Discussion*

What are Challenged Networks?

- Unusual
  - Containing features or requirements a networking architecture designer would find surprising or difficult to reason about



- Challenged
  - An operating environment making communications difficult
- *Examples*: mobile, power-limited, far-away nodes communicating over heterogeneous, poorly performing, intermittently-available links



RFC1149 : A Challenged Internet

- "…encapsulation of IP datagrams in avian carriers" (i.e. birds, esp carrier pigeons)
- Delivery of datagram:
  - Printed on scroll of paper in hexadecimal
  - Paper affixed to AC by duct tape
  - On receipt, process is reversed, paper is scanned in via OCR

Implementation of RFC1149

CPIP: Carrier Pigeon Internet Protocol

- See http://www.blug.linux.no/rfc1149/

Ping Results

```
Script started on Sat Apr 28 11:24:09 2001
vegard@gyversalen:~$ /sbin/ifconfig tun0
tun0      Link encap:Point-to-Point Protocol
          inet addr:10.0.3.2  P-t-P:10.0.3.1  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:150  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0
          RX bytes:88 (88.0 b)  TX bytes:168 (168.0 b)

vegard@gyversalen:~$ ping -i 900 10.0.3.1
PING 10.0.3.1 (10.0.3.1): 56 data bytes
64 bytes from 10.0.3.1: icmp_seq=0 ttl=255 time=6165731.1 ms
64 bytes from 10.0.3.1: icmp_seq=4 ttl=255 time=3211900.8 ms
64 bytes from 10.0.3.1: icmp_seq=2 ttl=255 time=5124922.8 ms
64 bytes from 10.0.3.1: icmp_seq=1 ttl=255 time=6388671.9 ms

--- 10.0.3.1 ping statistics ---
9 packets transmitted, 4 packets received, 55% packet loss
round-trip min/avg/max = 3211900.8/5222806.6/6388671.9 ms
vegard@gyversalen:~$ exit

Script done on Sat Apr 28 14:14:28 2001
```

Private Addresses

About 1.5 Hrs

High Loss

So What?

- Primary use for the Internet today is content upload/access
- Primary device for accessing information = mobile (by 2015)
- Mobile is a bit different than the “wired” Internet model
  - Performance of connectivity varies significantly over time
  - Latency can be high and asymmetric
  - Different end devices have various levels of “services” / capabilities
  - Capacity is, ultimately, limited

- It’s worth looking at network architectures to support all this
  - Without losing our investment in mobile data and TCP/IP

Internet Architecture

- Key design points
  - Packet abstraction is good
  - Fully-connected routing graph
  - Hierarchical address assignment
  - End-to-end reliability – dumb network
  - Management at the application layer
  - Security and accounting secondary (at ends)

Internet is a Packet Network

- Internet Protocol
  - Abstract IP datagram
    - Fragmentation function adapts its size
  - Globally-unique IP addresses
    - Addresses are hierarchical (prefix-based) to save routing table space and update size
  - Store-and-forward
    - Short-term storage of a few packets
    - Drop on overload (typically “drop tail”)

Internet is Fully-Connected

- Internet Protocol
  - Routing
    - Implemented as an application
    - Finds “best” (single) using prefixes
      - There should be lots of paths available, so pick one
    - No (transport-layer or higher) state in routers (just per-destination next-hops)
  - Drop on failure
    - “No route to host” – failure of the abstraction due to failure of the environmental or operational assumptions

Common Hierarchical Addresses

- Internet Protocol
  - Addresses
    - every interface has a 32-bit [unique] address
    - share a prefix with other nearby machines
      - subnets
      - CIDR and aggregation
  - Consequences
    - too few addresses –> IPv6 and NAT
    - mobility -> indirection
  - IPv6 doesn't change this much
    - But changes enough to not work with IPv4

Reliability is End-to-End

- Fate sharing
  - If one endpoint dies, the other might as well too
    - Consistent with connection abstraction
    - Simple network infrastructure, sophisticated end hosts
    - End hosts should behave
- E2e re-transmission is an appropriate method to combat packet loss

Management at Application Layer

- Control is in-band
  - Subject to same anomalies as regular data
  - Subject to attacks
- Management capabilities depend on which apps are installed/enabled
  - A limited *de-facto* standard set exist
- Management is the last thing to be enabled (e.g., after connectivity)

Security and Accounting

- Security is an “add-on” to Internet
  - Identity is not secured
  - Not implemented at one particular layer
  - Traffic management (filtering) vs end-to-end authentication
    - Filtering limited/fragile, authentication may be burdensome
    - Middlebox problems for e2e protocols
- Accounting
  - Difficult to account for and pay for use
  - Often a distributed data fusion problem

Operational Assumptions

- E2E path doesn't have *really* long delay
  - Reacting to flow control in ½-RTT effective
  - Reacting to congestion in 1-RTT effective
  - Connections open in at most a few seconds
- E2E path doesn't have *really* big, small, or asymmetric bandwidth
- Re-ordering might happen, but not much
- End stations don't cheat
- Links not very lossy (< 1%)
- Connectivity exists through *some* path
  - even MANET routing usually assumes this

Operational Assumptions (cont)

- Hosts are security principals
  - And (historically) rarely lie about who they are
  - And can be equipped with keys ‘easily enough’
- Nodes don’t move around or change addresses
  - assign addresses in hierarchy
  - thought to be important for scalability
- In-network storage is limited
  - not appropriate to store things long-term in network
- End-to-end principle
  - routers are ‘flakier’ than end hosts

Non-Internet-Like Networks

- Random and predictable node mobility
  - Mobile devices (phones, tablets, cars, planes)
  - Military/tactical networks (clusters meeting clusters)
  - Mobile routers w/disconnection (e.g. ZebraNet)
- Big delays, low bandwidth (high cost)
  - Store-and-forward satellites
  - exotic links (NASA DSN, underwater acoustics)
- Big delays, high bandwidth
  - *Data Mules*: buses, mail trucks, carts, etc.

Defining *Challenged* Networks…

- Intermittent/Scheduled/Opportunistic Links
  - Scheduled transfers can save power and help congestion; scheduling for exotic links
- High Error Rates / Low Usable Capacity
  - RF noise, light or acoustic interference, LPI/LPD concerns
- Very Large Delays
  - Natural prop delay could be seconds to minutes
  - If disconnected, may be (effectively) much longer
- Different Network Architectures
  - Different addressing / delivery abstractions
  - (specialized networks might never run IP)

Internet for Challenged Networks?

- What happens when one or more of the operational assumptions doesn’t hold (strongly)?
  - Applications break / communication impossible or unavailable
  - Applications have intolerable performance
  - System is not secure
- Let’s be more specific…

IP Routing May Not Work

- End-to-end path may not exist
  - Lack of many redundant links [there are exceptions]
  - Path may not be discoverable [e.g. fast oscillations]
  - Traditional routing assumes at least one path exists, fails otherwise
- Algorithm solves wrong problem
  - Wireless broadcast media is not an edge in a graph
  - Objective function does not match requirements
    - Different traffic types wish to optimize different criteria
    - Physical properties may be relevant (e.g. power)

IP Routing May Not Work [2]

- Routing protocol performs poorly in environment
  - Topology discovery dominates capacity
  - Incompatible topology assumptions
    - OSPF broadcast model for MANETs
  - Insufficient host resources
    - routing table size in sensor networks
  - Assumptions made of underlying protocols
    - BGP’s use of TCP

What about UDP?

- UDP preserves application-specified boundaries
  - May result in frequent fragmentation
  - Permits out-of-order delivery (no sequencing)
- Delay insensitive [no timers]
  - No provision for loss recovery
- No control loops
  - No flow/congestion control or loss recovery
- Works in simplex/bcast/mcast environment
  - no connections

What about TCP?

- Reliable in-order delivery streams
- Delay sensitive [6 timers]:
  - connection establishment, retransmit, persist, delayed-ACK, FIN-WAIT, (keep-alive)
- Three control loops:
  - Flow and congestion control, loss recovery
- Requires duplex-capable environment
  - Connection establishment and tear-down

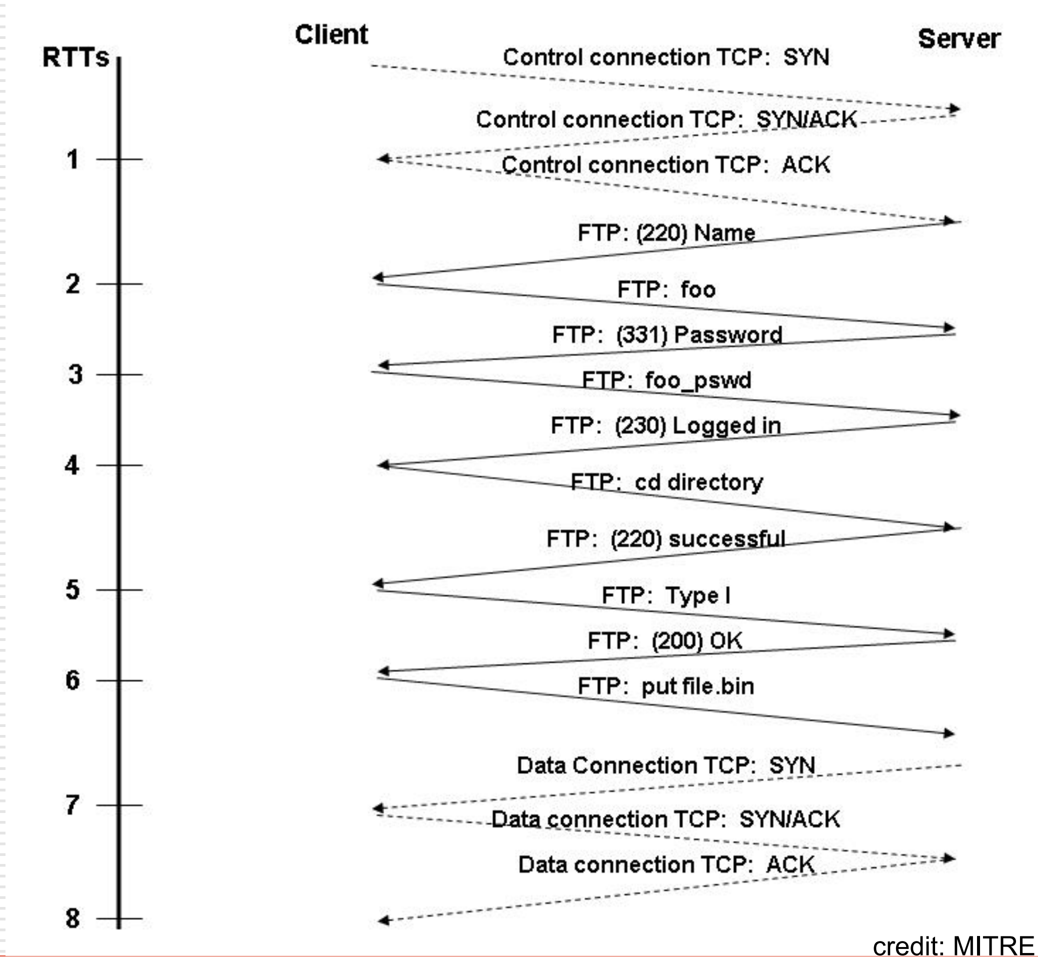
What about DNS?

- Names and the DNS:
  - Names: Administrative assignment (global hierarchy)
  - DNS Distributed Lookup Service
    - Name service frequently located near target
    - Requires ~1RTT or more to perform first mapping
    - Caching helps after that
    - Often a reverse-lookup is also required
- Zone and dynamic updates
- DNS Resolution Failure results in effective application failure or large application delays

What about Applications?

- Most use TCP… ouch
- Detecting failures
  - Many applications have an inactivity timeout used to initiate failure-handling
  - Handling failures often means giving up
- Chattiness
  - Many applications implement layer 7 protocols that require lots of round-trip exchanges
  - Extreme cases drive conversation to stop-and-wait
- Robustness to long delays
  - Most apps aren’t prepared to continue effectively after re-start or other network disruption
  - And its even worse now with VPNs, NATs, etc.

FTP: An example application



credit: MITRE

Applications that are interactive exacerbate channel access problems

What to Do?

- Some problems surmountable using Internet/IP
  - ‘cover up’ the link problems using PEPs
  - Mostly used at “edges,” not so much for transit
- Performance Enhancing Proxies (PEPs):
  - Do “something” in the data stream causing endpoint (TCP/IP) systems to not notice there are problems
  - Lots of issues with transparency– security, operation with asymmetric routing, etc.
  - no really standardized proxy architecture
- Some environments mix heterogeneous technology and *never* have an e2e path

Outline

- *Introduction: The Internet and Challenged Networks*
- ***The DTN Architecture***
- *DTN People & Projects*
- *Discussion*

Delay-Tolerant Networking Architecture Goals

- Support interoperability across ‘radically heterogeneous’ networks
  - Handle differing packet formats
  - Handle differing naming schemes
  - Handle differing temporal assumptions
- Tolerate delay and disruption
  - Acceptable performance in high loss/delay/error/disconnected environments
  - Decent performance for low loss/delay/errors

DTN Architectural Components

- Flexible naming scheme based on URIs
- Store-Carry-Forward Routing Framework
- Extensible, arbitrary length messages
- Endpoint migration (“custody transfer”)
- Data-oriented security model

Naming using URIs

- URIs (RFC3986) – URLs and URNs
  - Reserved strings and characters:
    - **: / ? # [ ] @   (generic delims)**
    - ! $ & ' ( ) * + , ; = (sub-delims)
  - Generic format

    URI = scheme ":" hier-part ["?" query] ["#" fragment]
    hier-part = "//" authority path-abempty
        | path-absolute | path-rootless | path-empty
    Authority = [userinfo "@"] host [":" port]
    Path-abempty (begins with / or is empty)
    Path-absolute (begins with / but not //)
    Path-rootless (begins with a segment)
    Path-empty (empty)
- Example: URN:ISBN:0-395-36341-1

URIs in DTN

- URIs can encode any existing or future network name & address format
- Can use them to identify endpoints (EIDS):
  - multicast, anycast, unicast, security principals
- ***Late binding*** of EID permits naming flexibility and robustness to change:
  - EID "looked up" only when necessary during delivery so can change over long delivery delay
  - contrast with Internet lookup-before-use DNS/IP
- Example: **`dtn:gw.dtn/myapp?a=3`**

DTN PDUs: Bundles

- IPN idea: “bundle” together all necessary ancillary data to complete work unit [ADU]
- Large ADUs allow for network to assign scheduling / buffer resources
  - Proactive fragmentation (e.g. for multiple paths)
- Bundle delivery is mostly best-effort
  - Hard to provide e2e reliability over disrupted paths
  - Apps can request ACKs and/or “custody transfer”
- Extensible format using *blocks* supports experimentation and evolution

Bundles and Blocks

- Bundles are a linear collection of *blocks* (like IPv6 extension headers)
  - First is a required primary block
  - Followed by (extensible set of) other blocks
- Block format shares initial version or ID field
  - Remaining fields are generally variable
  - More difficult processing but greater flexibility

Primary Block Format

Version (1 byte)	Bundle Processing Control Flags (SDNV)
Block Length (SDNV)	
Destination Scheme Offset (SDNV)	Destination SSP Offset (SDNV)
Source Scheme Offset (SDNV)	Source SSP Offset (SDNV)
Report-To Scheme Offset (SDNV)	Report-To SSP Offset (SDNV)
Custodian Scheme Offset (SDNV)	Custodian SSP Offset (SDNV)
Creation Timestamp (SDNV)	
Creation Timestamp Sequence Number (SDNV)	
Lifetime (SDNV)	
Dictionary Length (SDNV)	
Dictionary (byte array)	
Fragment Offset (SDNV, optional)	
Application data unit length (SDNV, optional)	

SDNVs: variable-length values

Offsets support string re-use

Timestamp combines real time and sequence

TTL is real-time offset from creation

Self-Delimiting Numeric Values (SDNVs)

- Variable-length encoding format
  - Avoids hazards of fixed-length fields
  - Represents non-negative integers
  - 1 bit per byte of overhead (plus overflows)

See RFC6256

- High-order bit of each byte: 0 =“end”
  - 1 → 00000001
  - 127 → 01111111
  - 128 → 10000001 00000000
  - 32767 → 11111111 01111111

Synchronized Time

- DTN assumes roughly synchronized time
- Four drivers for this choice
  - Most DTN applications care about time (e.g., when some value is sensed)
  - Space/time DTN routing requires time knowledge
  - Management tasks much easier
  - Time typically provided elsewhere anyhow

DTN Routing

- DTN is an overlay routing network
  - Nodes (fixed or moving) have storage
  - Bundles are routed among DTN nodes
  - Bundles may be fragmented
- DTN routing is a little unusual
  - Multiple paths can be used in parallel
  - Multiple transport encapsulations can be used in parallel
  - Thus, DTN routing involves not just “next hop” but also “next protocol”

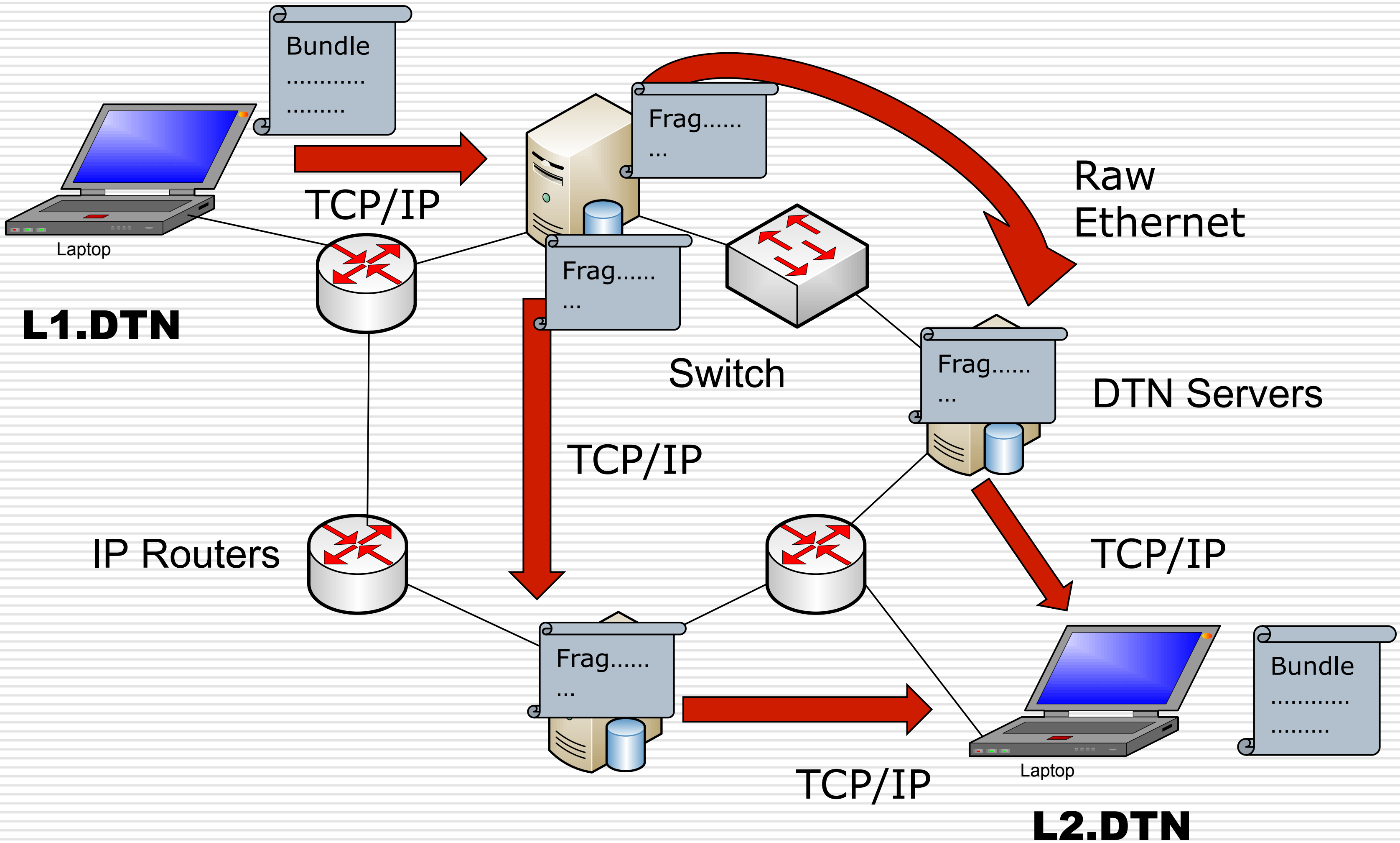
DTN Fragmentation

- Proactive fragmentation
  - Fragmentation performed prior to send
  - Supports filling "contacts"
  - Fragments may be fragmented
- Reactive fragmentation
  - Repackage as a fragment a partially-received fragment
  - Performed at bundle layer in case different hops are needed

Next Hop/Protocol in DTN

- In IP, routing function R(d) gives N (next hop)
  - d is IP destination, N is IP next hop
  - d and N are IP addresses
  - R is a longest matching prefix compare
- In DTN, R(d) gives a matrix M
  - R(d): “best” string match returns at least (N_i, P_i, L_i)
  - N_i is next-hop DTN EID, P_i is next-layer down protocol encap, L_i is next layer down address
  - Multiple matching entries can split (fragment) or replicate bundles

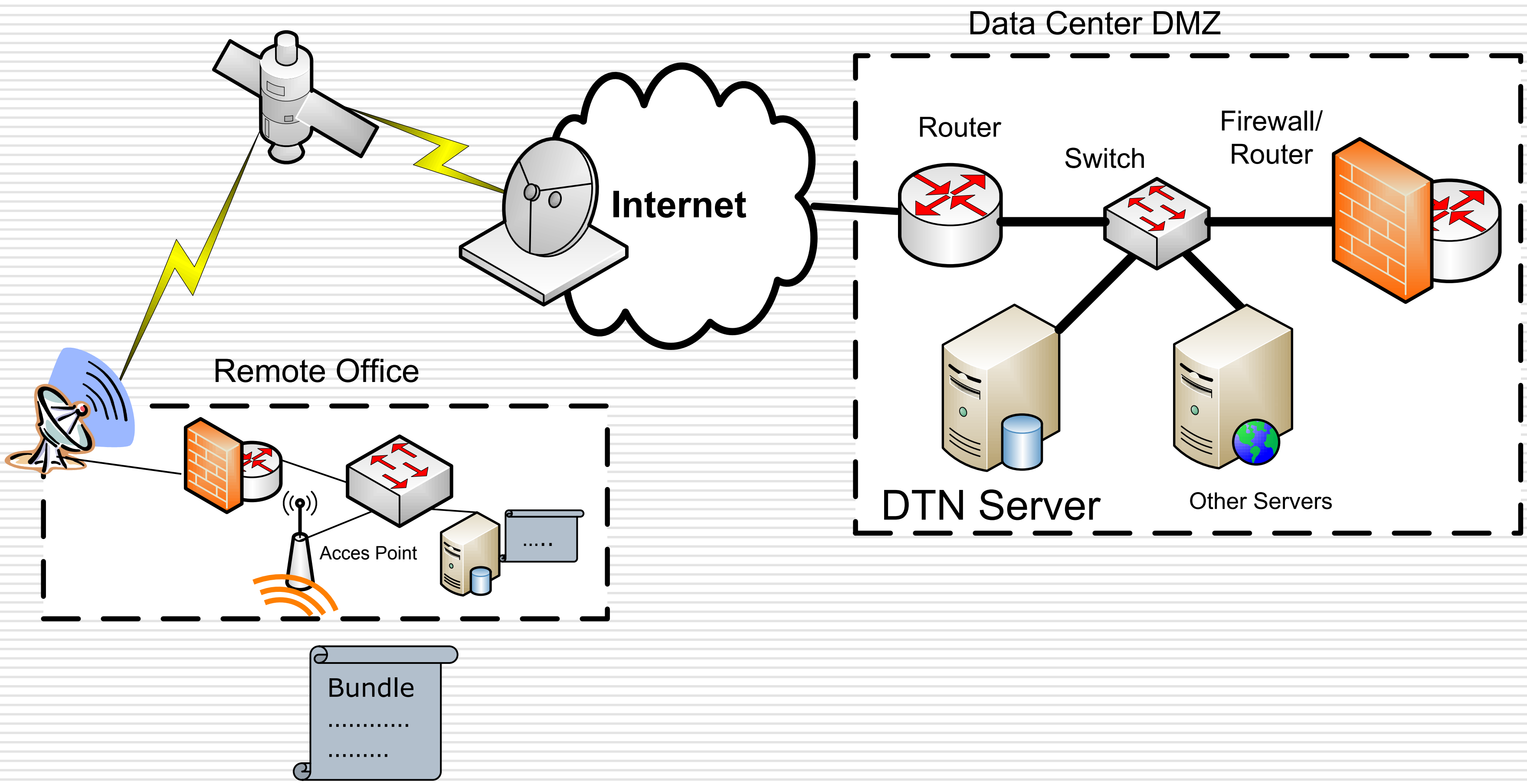
Example: next-hop/protocol



DTN Custody Transfer

- A (optional) transfer of delivery responsibility from one DTN node to others along delivery path(s)
- Avoids problems of poor e2e performance for high-delay lossy networks
  - Frees sender's buffers (relatively) early
  - Useful for low-capability sources
- Custodian nodes in 'good places'
  - E.g. servers in data centers

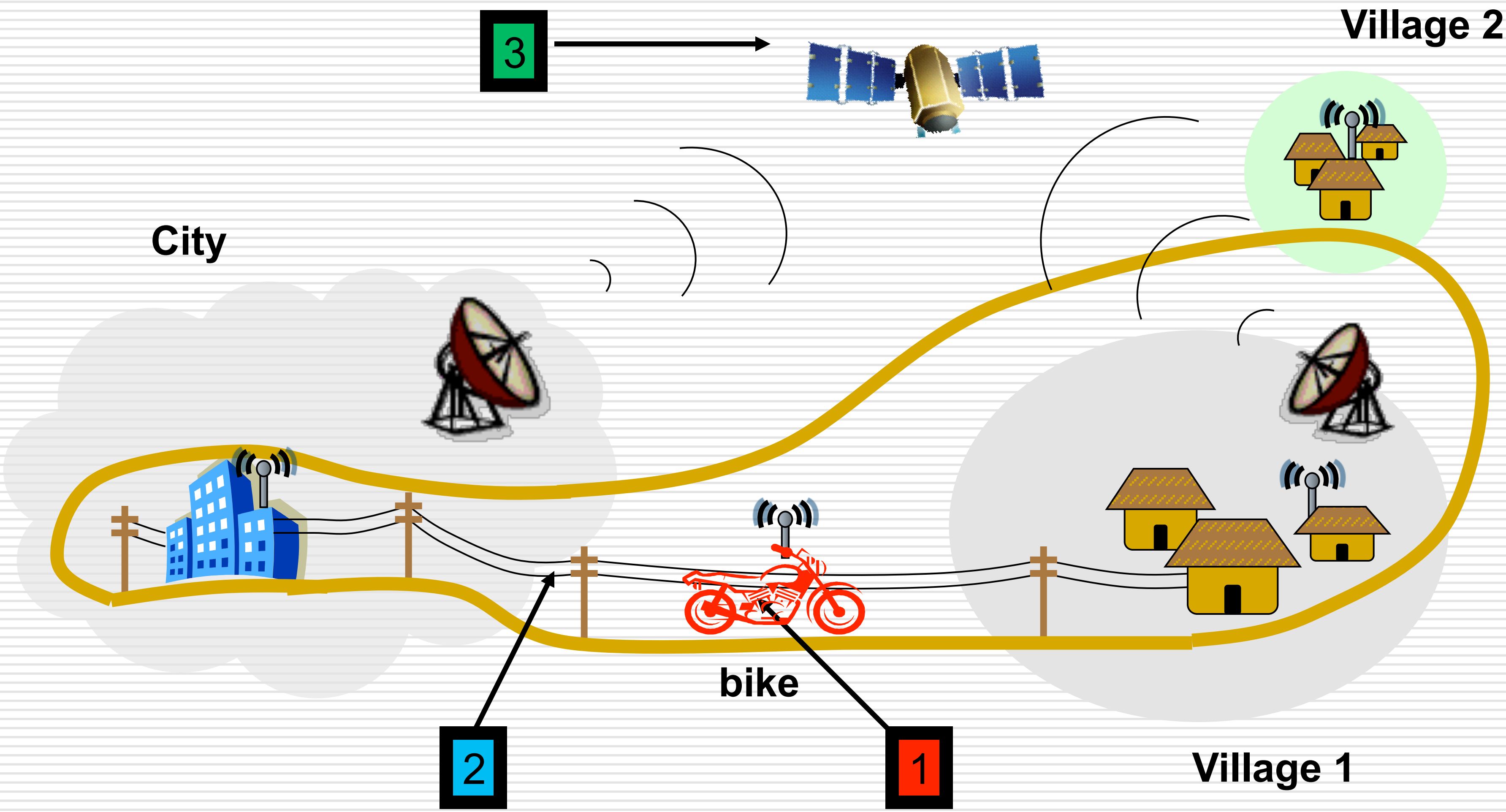
Example: custody transfer



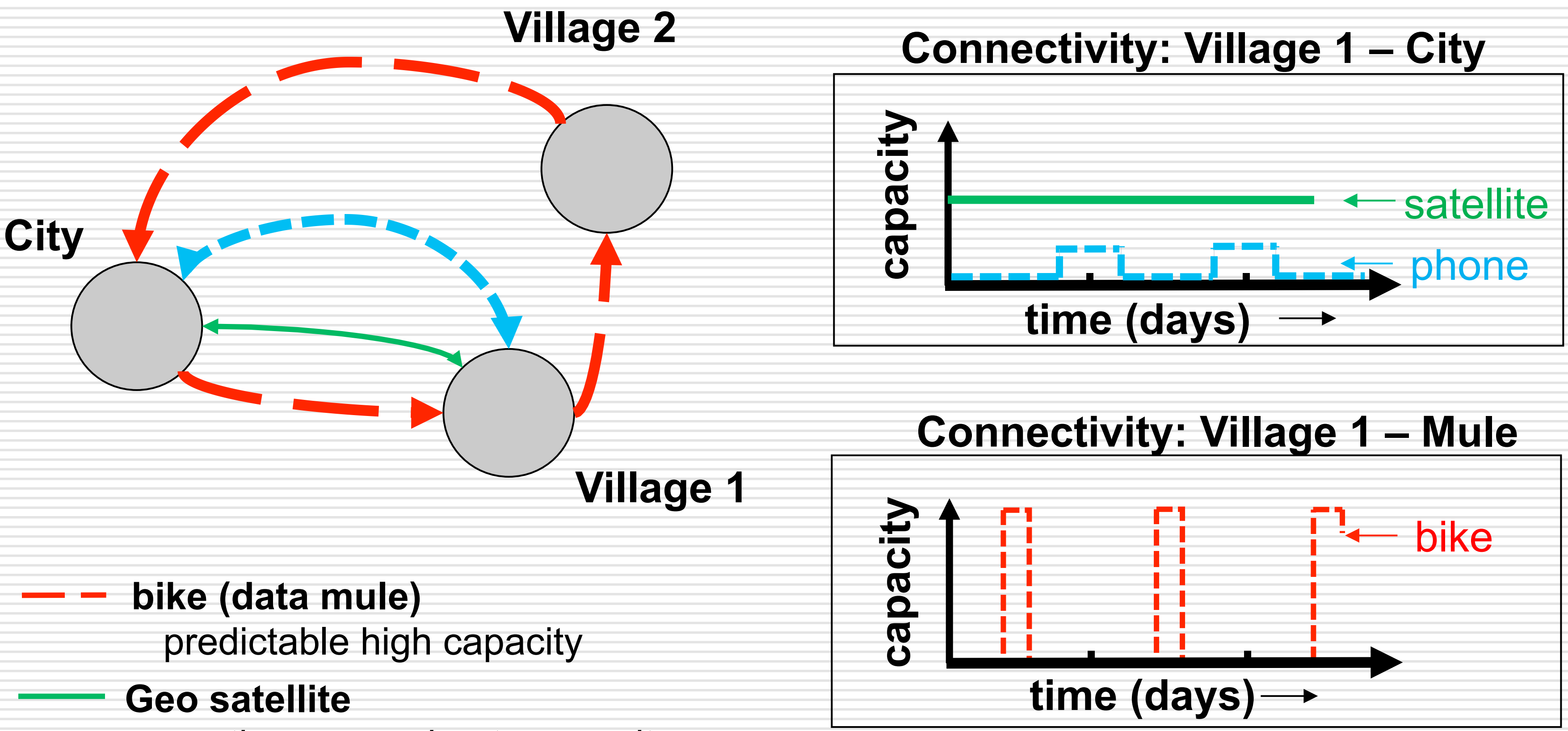
DTN and Mobility

- Mobility patterns induce a set of connectivity opportunities [contacts]
- Contacts have a time-varying bandwidth and delay
- Definition of a contact:
  - $(e_1, e_2, t_s, t_e, C(t), D(t))$
  - e_1, e_2: endpoint identifiers
  - t_s, t_e: contact start and end time (at e_1)
  - $C(t)$: continuous capacity function
  - $D(t)$: continuous delay function

Example: DTN Store-Carry-Forward



Example Graph Abstraction



bike (data mule)
predictable high capacity

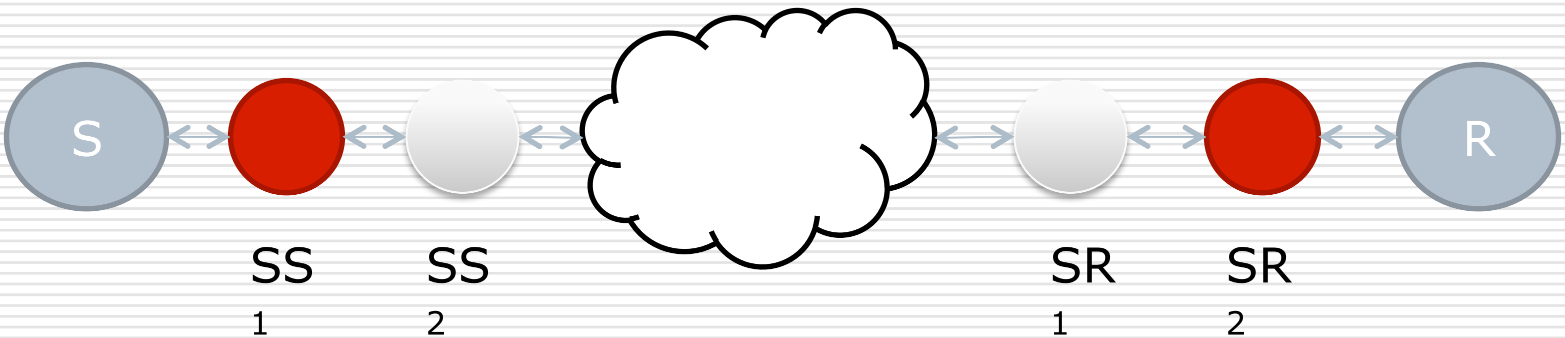
Geo satellite
continuous moderate capacity

dial-up link
on-demand low capacity

DTN Security

- DTN security protects data being transferred and access to transport
- Authentication, confidentiality, and data integrity are integral
- But the environment is a challenge
  - Can’t assume servers are available
  - Link resources can be precious
  - Nodes may move into hostile locations
  - Routing can involve delays and loops
  - Nodes have heterogeneous capabilities

Security Sources/Destinations



- Bundle sender/receiver distinguished from security sender/receiver (for service *k*)
  - Heterogeneous capabilities (e.g. crypto, keying)
  - Different security needs based on topology

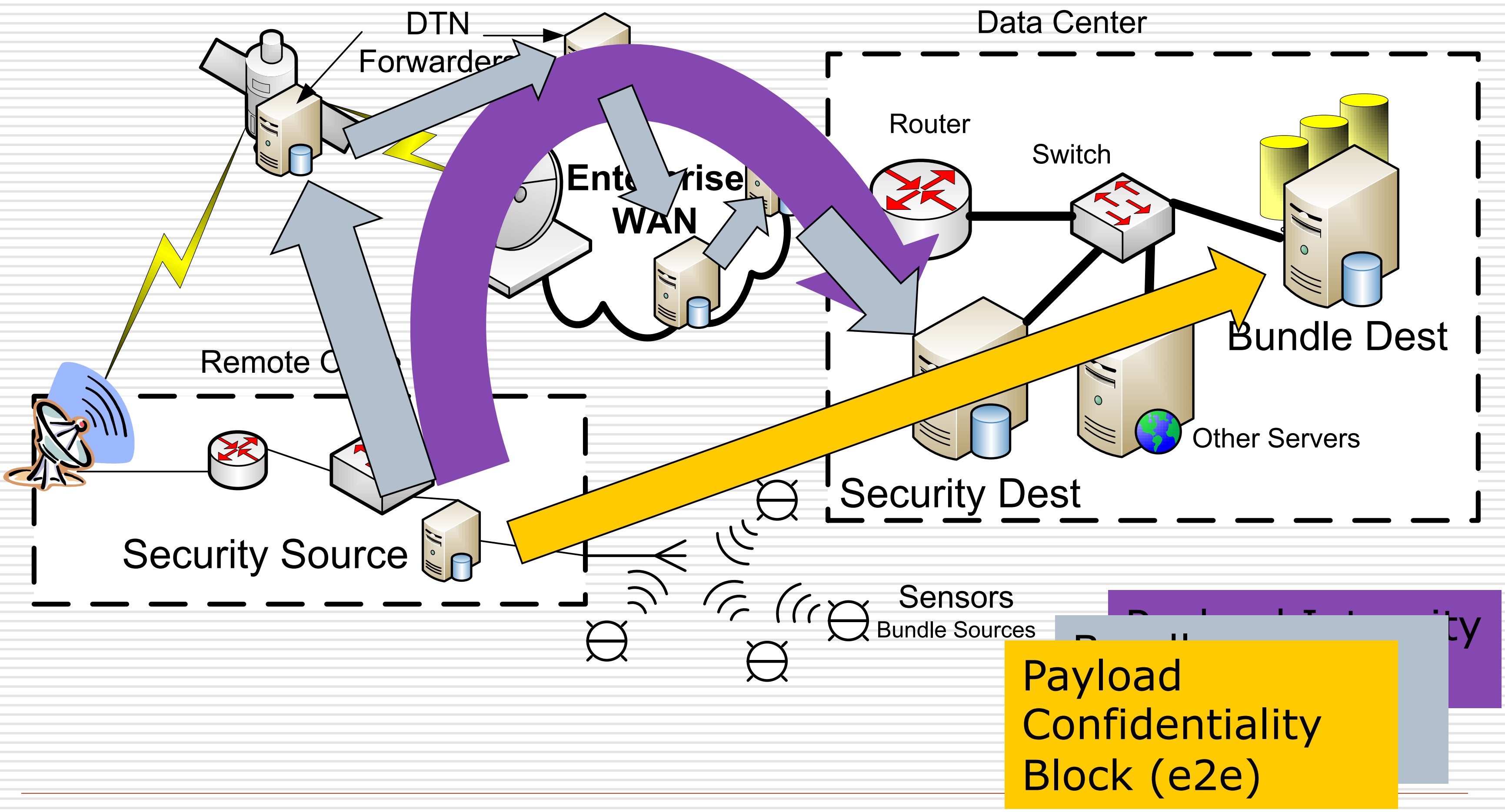
DTN Threats

- Other layers (non-DTN nodes)
- Unauthorized resource consumption
- Denial of service
- Confidentiality and integrity attacks
- Traffic storms
- Free-riding on legitimate traffic

DTN Security Blocks

- Integrity and Authentication
  - BAB – bundle authentication block
    - Hop-by-hop between forwarders
    - Indicates upstream router & data is ok
  - PIB – payload integrity block
    - "End-to-end" between PIB sources/dests
    - Indicates payload and sender is ok
- Confidentiality
  - PCB – payload confidentiality block
    - Supports encryption of payload
    - "End-to-end" between PCB sources/dests

Example: bundle security blocks



Abstract Security Blocks

<table>
<tr><td>Type</td><td>Flags (SDNV)</td><td>EID reference list
(composite, if present)</td></tr>
<tr><td colspan="2">Length (SDNV)</td><td>Ciphersuite ID (SDNV)</td></tr>
<tr><td colspan="2">Ciphersuite Flags (SDNV)</td><td>Correlator (SDNV, if present)</td></tr>
<tr><td>Param Length
(SDNV)</td><td colspan="2">Ciphersuite param data ……</td></tr>
<tr><td>Result Length
(SDNV)</td><td colspan="2">Security result data ……</td></tr>
</table>

Ciphersuite Flags:

Bit#: 6 5	4	3	2	1	0
reserved	src	dest	parm	corr	res

Confidentiality Details

- Typically, security result contains a random bundle encryption key (BEK)
- Payload encrypted "in-place"
  - Payload block cleartext → cyphertext
  - Fragmentation and custody ACKs ok
- Some crypto algorithms support this
  - Counter-mode encryption generally
  - GCM (CTR+Galois authentication) [NIST] specifically

Mandatory Ciphersuites

- BAB-HMAC uses HMAC-SHA1 [RFC2104]
- PIB-RSA-SHA256 uses sha256WithRSAEncryptionPKCSv1.5 [RFC4055]
- PCB-RSA-AES128-PAYLOAD-PIB-PCB
  - Encrypts PIBs, PCBs and payload block
  - AES in GCM mode [RFC5084] – 128bits
  - GCM counter limits bundle to ~ ½ TByte

Security Policy Minimums

- Under what conditions recvd bundle is
  - Forwarded, reqd to have valid BAB/PIB/PCB, given a BAB/PIB/PCB,
  - (e.g. dropped) if policy violated
- Information adequacy
  - Is information included in the BAB/PIB considered adequate to authenticate?

IRTF Documents & IANA Allocations

Published RFCs

- V. Cerf et al, Delay Tolerant Networking Architecture", RFC 4838, Apr 2007
- K. Scott, S. Burleigh, "Bundle Protocol Specification", RFC 5050, Nov 2007
- S. Farrell et al, "Licklider Transmission Protocol – Security Extensions," RFC 5327, Sep 2008
- M. Ramadas et al, "Licklider Transmission Protocol – Specification," RFC 5326, Sep 2008
- S. Burleigh et al, "Licklider Transmission Protocol – Motivation," RFC 5325, Sep 2008
- M. Blanchet, "Delay-Tolerant Networking Bundle Protocol IANA Registries," RFC 6255, May 2011
- W. Eddy, E. Davies, "Using Self-Delimiting Numeric Values in Protocols," RFC 6256, May 2011
- S. Symington, S. Farrell, H. Weiss, P. Lovell, "Bundle Security Protocol Specification," RFC 6257, May 2011
- S. Symington, "Delay-Tolerant Networking Metadata Extension Block," RFC 6258, May 2011
- S. Symington, "Delay-Tolerant Networking Previous-Hop Insertion Block," RFC 6259, May 2011
- S. Burleigh, "Compressed Bundle Header Encoding (CBHE)," RFC 6260, May 2011

IANA Allocations

- "dtn:" scheme
- TCP / UDP Internet Convergence Layers (CLs) - Port 4556
- not to be confused with Port 2445 ("DTN1")

IRTF Drafts

- Drafts (alive)
  - draft-blanchet-dtnrg-bp-application-framework
  - draft-dtnrg-ltp-cbhe-registries
  - draft-sims-dtnrg-bpmib
  - draft-softgear-dtnrg-eprophet
- Drafts (dead, but might come back)
  - draft-irtf-dtnrg-ltpcl
  - draft-irtf-dtnrg-udpcl
  - draft-eddy-dtnrg-checksum
  - draft-eddy-dtnrg-eid
  - draft-fall-dtnrg-schl
  - draft-farrell-dtnrg-alt-time
  - draft-farrell-dtnrg-bpq
  - draft-irtf-dtnrg-dtn-uri-scheme
  - draft-irtf-dtnrg-ipnd
  - draft-irtf-dtnrg-prophet
  - draft-irtf-dtnrg-sec-overview
  - draft-irtf-dtnrg-tcp-clayer
  - draft-mcmahon-dtnrg-dtn-edp ↵
  - see https://datatracker.ietf.org for others

Availability

- All code is open source and freely available
  - http://www.dtnrg.org/wiki/Code
  - DTN2, ION, POSTELLATION, IBR-DTN, DASM
  - Mercurial repository
    - hg clone http://www.dtnrg.org/hg/oasys
    - hg clone http://www.dtnrg.org/hg/DTN2
- DTN mailing lists
  - http://irtf.org/mailman/listinfo/dtn-interest
  - http://irtf.org/mailman/listinfo/dtn-users

Outline

- *Introduction: The Internet and Challenged Networks*
- *The DTN Architecture*
- ***DTN People & Projects***
- *Discussion*

DTN People & Projects

- DTNRG (IRTF) – various folks
- Trinity College Dublin (Ireland) – Stephen Farrell & Alex McMahon
- Aalto University (Finland) – Jörg Ott
- NASA JPL, GRC (USA) – Scott Burleigh, Will Ivantik
- MITRE (USA) – Bob Durst & Keith Scott
- Google (USA) – Vint Cerf
- DARPA WNaN Program (USA) – see DARPA web site
- TU Braunschweig (Germany)
- Viagenie (Canada) – Marc Blanchet
- BBN/Raytheon
- Ohio University (USA) – Hans Cruse

Relevant Links

- DTNRG:
  - http://www.dtnrg.org
- DARPA WNaN Program:
  - http://www.darpa.mil/Our_Work/STO/Programs/Wireless_Network_after_Next_%28WNAN%29.aspx
- U Mass Diverse Outdoor Mobile Environment
  - http://prisms.cs.umass.edu/dome
- Tetherless Computing:
  - http://blizzard.cs.uwaterloo.ca
- EDIFY Research Group:
  - http://edify.cse.lehigh.edu/
- Technology and Infrastructure for Emerging Regions:
  - http://tier.cs.berkeley.edu/
- DTN Group @ TKK Netlab:
  - http://www.netlab.hut.fi/~jo/dtn/index.html
- N4C:
  - http://www.n4c.eu

Outline

- *Introduction: The Internet and Challenged Networks*
- *The DTN Architecture*
- *DTN Reference Implementation*
- *DTN People & Projects*
- ***Discussion***

Thanks

http://www.dtnrg.org

kfall@qualcomm.com