Delay-tolerant Networking:

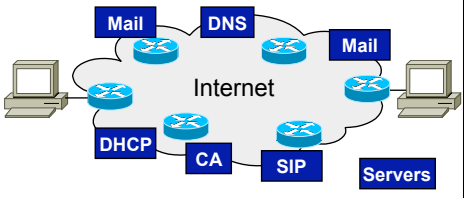
DTNRG Architecture

RFC 4838

S-38.3151: Delay-tolerant Networking

Motivation (1): Traditional Networks

- Typical networking assumptions
  - E2E path exists, changes rarely
  - Reasonable path characteristics
  - Short RTT (max O(1s))
  - Accessible infrastructure (servers, …)



- Traditional Protocol Operation
  - End-to-end operation at the IP layer and above
  - Protocol operations can be confirmed “instantaneously”
  - Mutual reachability (return path) can be validated
  - Information can be obtained (e.g., DNS) and validated (e.g., certificates)
- Exceptions
  - Application protocols with intermediaries (mail servers, caches, proxies)
  - Redefining the “ends”

Motivation (1): Challenged Networks

- Deviating from traditional networking environments

A. Challenges arising from the communication links
  - Very long delay (deep space: moon: 3s, Mars: 2min, Pluto: 5h)
  - Very low speed links (e.g., acoustic underwater modems: 1 bit/s–few kbit/s)
  - High bit error rate (wireless, underwater, satellite, stellar)
  - **Interactive communication may not be possible/efficient or reliable**

B. Node reachability and density
  - Predictable: Planetary dynamics, scheduled vehicles, message ferries
  - Semi-predictable: Sparse sensor networks, data mules, vehicular
  - Unpredictable motion (animals, vehicles, etc.)
  - **End-to-end path may not exist**

- Human mobility—will return to this aspect later

Example: Sparse Sensor Networks

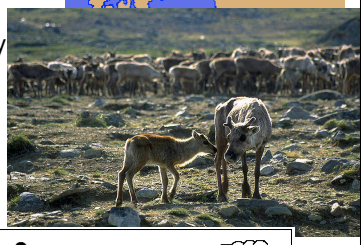
- Sensor networks without end-to-end path
  - Traditional ad-hoc routing not applicable
  - Collect and store data, forward opportunistically
  - Offload to fixed or mobile access gateways
  - Limited infrastructure support
    - Mobile sensors + stationary sinks
    - Stationary sensors + mobile sinks / forwarders (e.g., message ferries, data mules)
    - Stationary storage / forwarding stations
- Zebranet
  - Monitoring a wild-life habitat with networked computers
  - Ad-Hoc Networks, computers on Zebra exchange information dynamically
- Applications in Oceanic studies
  - Measurements using sensors on seals, whales, etc.
  - Also: fixed underwater measurement equipment
- Seismic and fire monitoring in remote areas

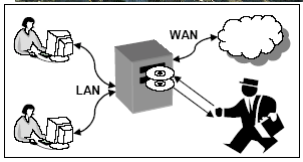
Example: Carrier Pigeons

- RFC 1149, RFC 2549
- Implemented by Bergen Linux users group
  - Printed datagrams on paper
- Further experiments in Israel (Wi-Fly)
  - Used tiny memory of 1.3 GB per pigeon
- Characteristics
  - High delay
  - Don't fly at night (your favorite surfing time)
- Up to 1.5 Mbit/s data rate, faster than simple ADSL

Example: Internet Access in Remote Areas

- Sámi Network Connectivity
  - Provide Internet Connectivity for Sámi population of Reindeer Herders
  - Nomadic users, no reliable communication facilities
  - Mix of fixed and mobile gateways
  - Routing based on probabilistic patterns of connectivity
  - E-Mail, Web cache prefill, file transfer
- DakNet
  - Internet access for remote villages in India and Cambodia
- Wizzy Digital Courier service
  - Using motorcycles to carry message to/from villages
- ZebraNet
  - Sensor network for habitat monitoring in Africa
- Postmanet





Store and Forward Communications

- It's hard to get a similar data rate compared to a container load of:
- DVDs: 4.7 GB
  - 2 DVDs in a jewel case: 190.0 x 142.2 x 6.9 mm
  - 1 device per hour = 10.4 Mbit/s
  - R/W via 802.11g: ~30min
- 2.5" HDD: 160 GB
  - 9.5x69.85x100.2mm
  - 1 device per hour = 355 Mbit/s
  - R/W via 802.11g: ~17 hours
- 4 GB SD card
  - 18 mm x 24 mm x 1.4 mm, 1.5g
  - 1 device / hour = 8.9 Mbit/s
  - R/W via 802.11g: ~27 min
- Filling a shipping container: 5.89 x 2.33 x 2.38 m
  - Ferry across the Baltic Sea: 20 hours
  - DVD: 178 Gbit/s, HDD: 8.5 Tbit/s, RS-MMC: 480 Tbit/s

Motivation (2): (Human) Mobility

- Mobility means (potential) disconnection

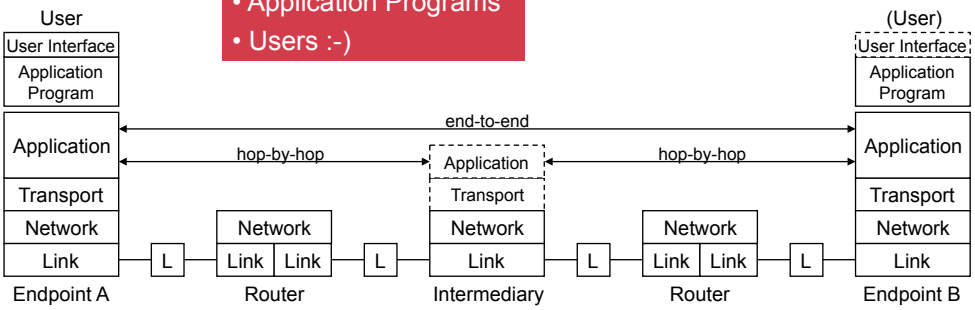
A. Challenges arising from the communication links
  - Connectivity is usually not ubiquitous (particularly when moving)
  - Even if available, permanent connectivity may be expensive
  - Further limitations: battery power, legal and social aspects
  - **Interactive communication may not be possible or cost-effective**

B. Node reachability and density
  - Ad-hoc and peer-to-peer networking may help overcoming the issues above to some extent, but:
  - There may not be enough people around
  - There may not be enough people with compatible devices
  - There may not be enough people willing to cooperate
  - **End-to-end path may not exist**

Motivation (2): Dealing with Mobility

- Many lower layer mechanisms available today
  - Mobile IP, HIP, various transport and session layer approaches
  - End-to-end as well as using intermediaries
- Disconnections and delays make all of them fail
- Timeouts!

• Application Protocol
• Application Programs
• Users :-)



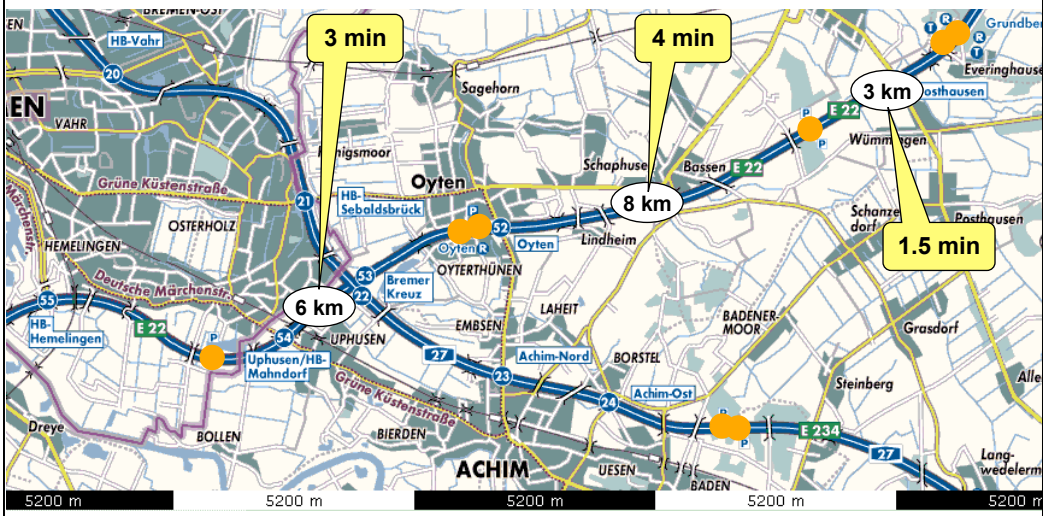
Extreme Mobility: Drive-thru Internet

- Opportunistic use of access networks
  - Unpredictable and potentially (well: likely!) short connectivity periods
  - Disconnections for arbitrary durations
  - Changing IP addresses, access links (characteristics, L2 technology, ISP)
  - May have perfect first hop connectivity
    - But potential bottlenecks in the access links and server/peer performance
  - Disconnection tolerance required for applications
- More general: ad-hoc communication without end-to-end path
  - Use other vehicles as data carriage while not connected
    - E.g., pausing in a parking lot without Internet access
    - E.g., two users in different cars not connected at the same time
  - Generalization: mobile Internet access without end-to-end connectivity
  - Asynchronous communication needed

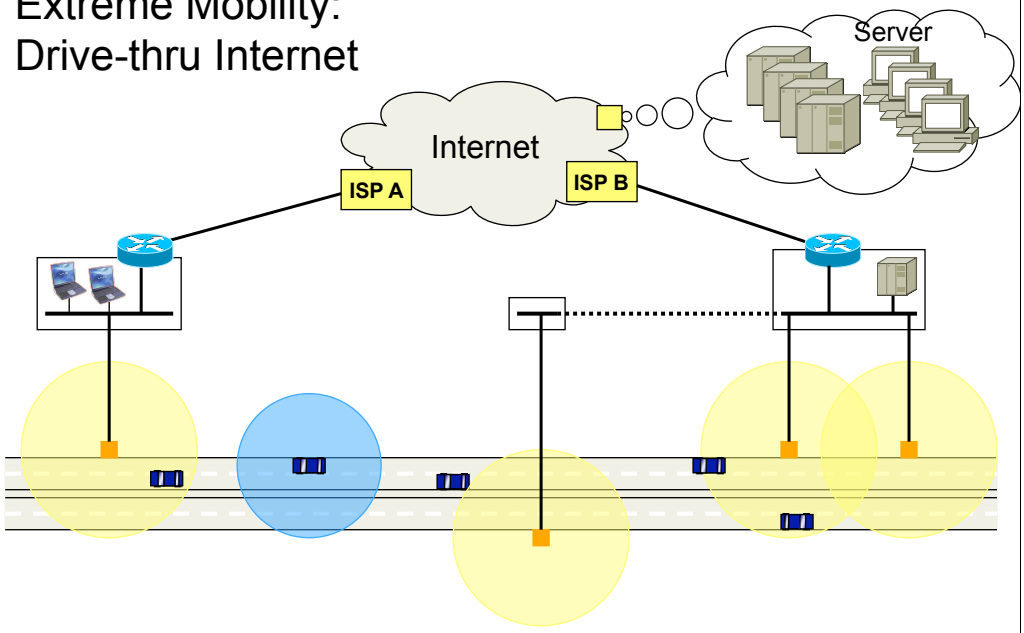
Extreme Target Environment: Autobahn

- 12,174 km in 2005
- ~700 service areas
  - Every 18 km on average
  - Usually: every 40–60 km
  - Denser in urban areas
- Up to 190,000 vehicles/d
- Lots of variation in
  - Speed, car density, …
- Applicable to highways, city traffic, countryside, too

Extreme Target Environment: Autobahn

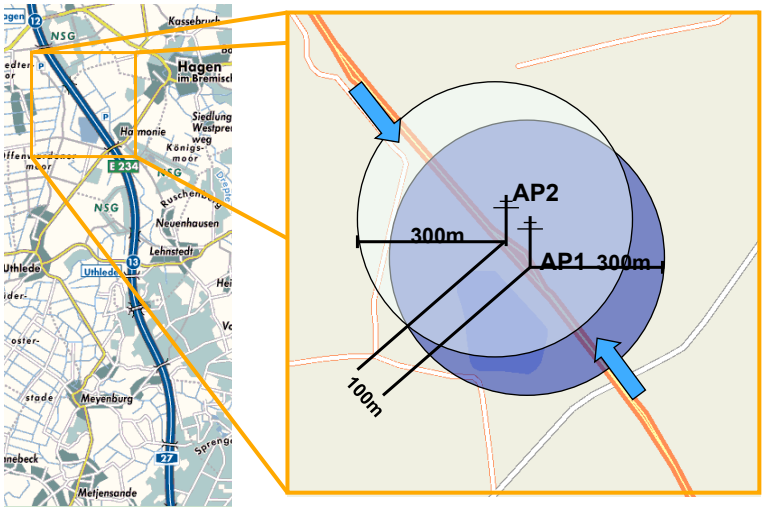


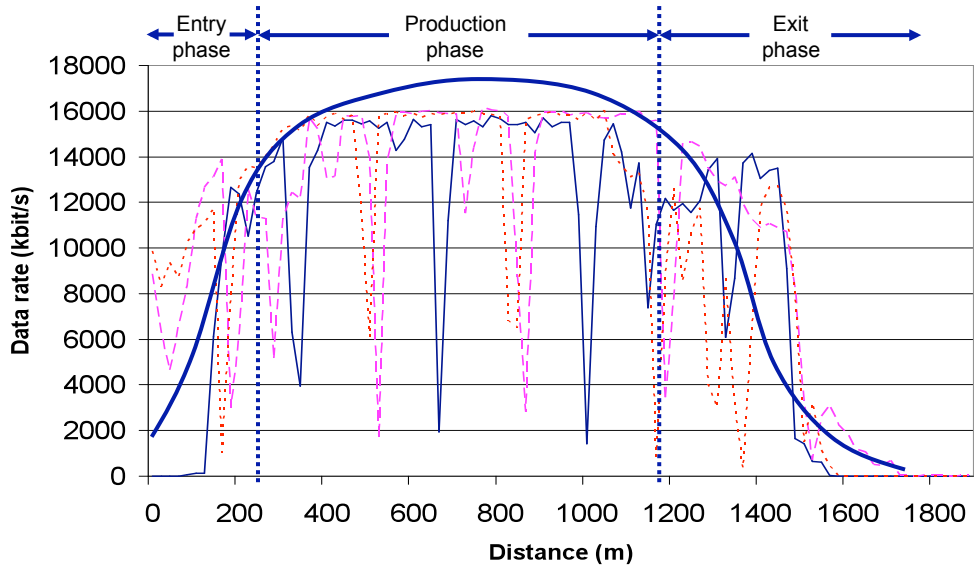
Extreme Mobility: Drive-thru Internet



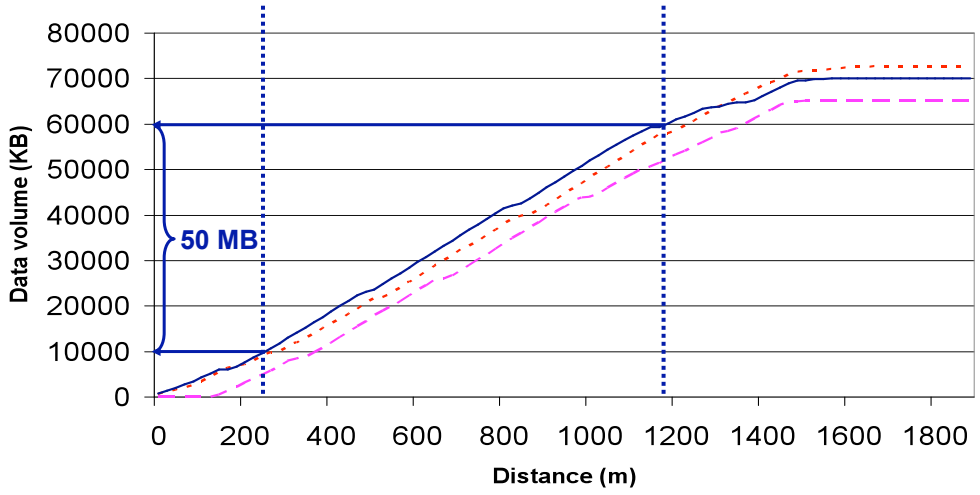
Measurements: Autobahn

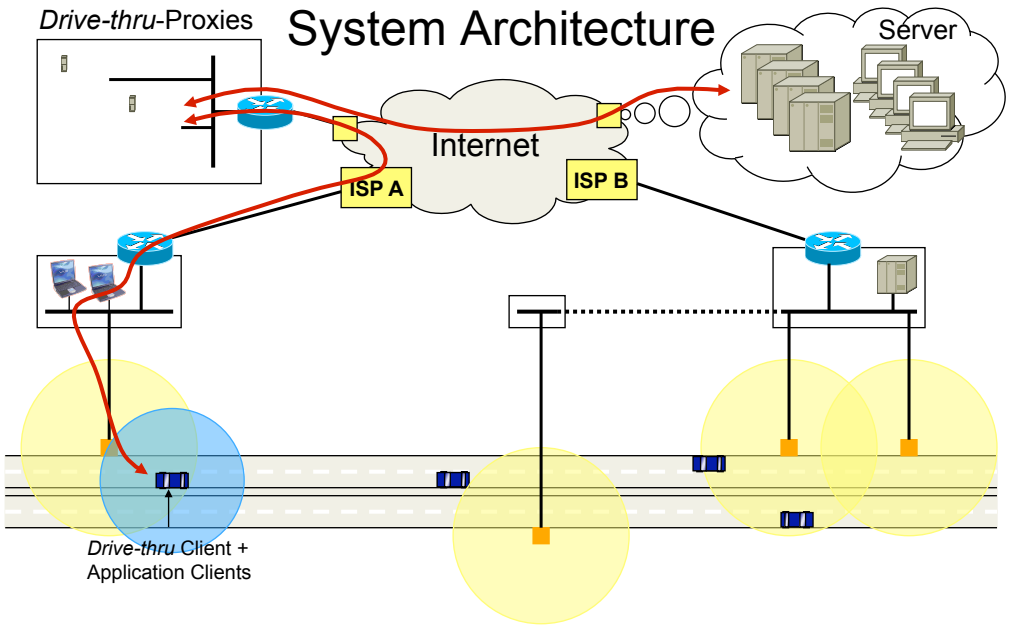
N 53° 20.484'
E 8° 36.425'



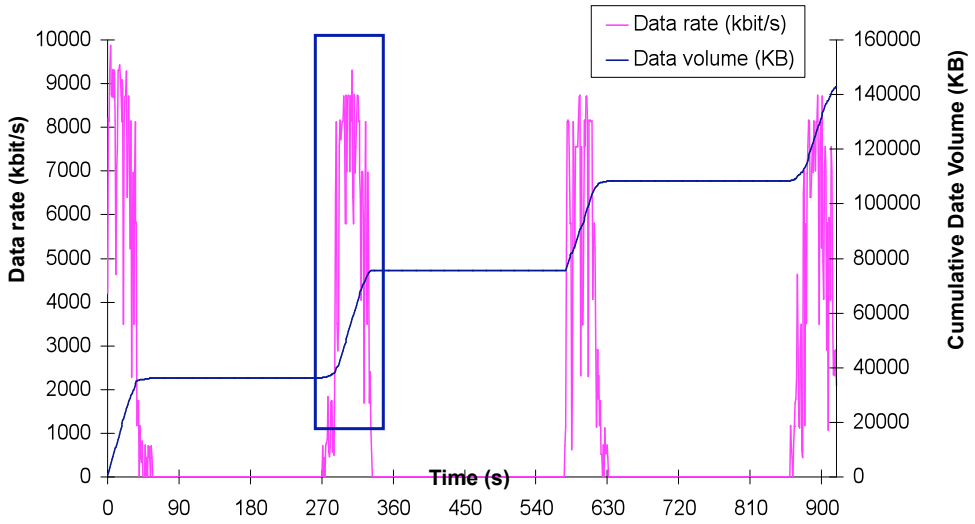


802.11g Data Volume (120km/h)

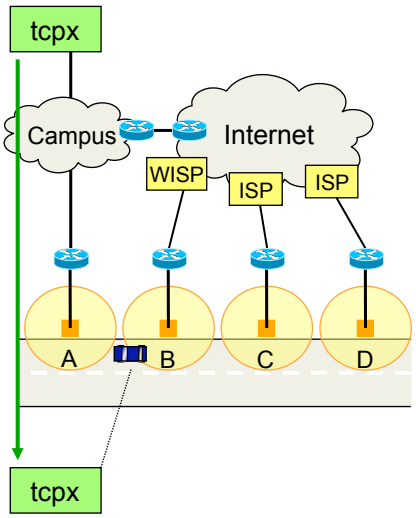




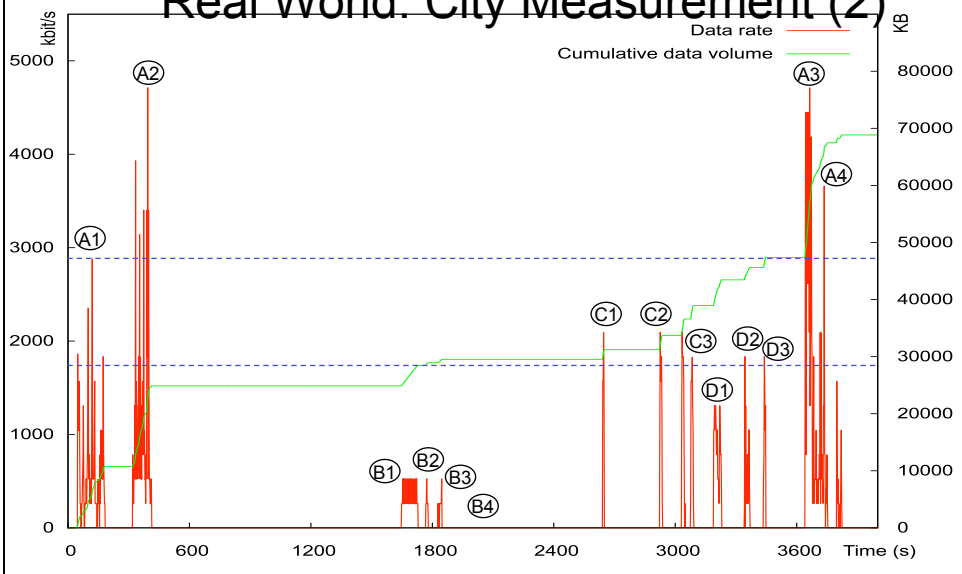
Preserving Communication across Hot-Spots



Real World: City Measurement



Real World: City Measurement (2)



Transport and Applications

Subset of applications in principle usable

- Asynchronous (mail), transaction-based (web), distributed "objects" (files)

provided that…

- Transport connections persist
  - Since application interactions may not complete in a single hot-spot
- Application-specific support is available ("ALGs" or endpoints)
  - Deal with application timeouts and allow for disconnected operation
- Strong authentication is provided
  - As resources may be allocated at components in the fixed network
- Efficient operation is possible
  - Minimize round-trips and overhead (and allow for L2 triggers)

Summary of Challenges

- Intermittent, unpredictable connectivity periods and blackouts
  - Unpredictable, possibly short-lived connectivity
  - Frequent network partitions
  - Non-existent end-to-end paths

Reachability

- Transmission characteristics
  - Potentially: Low data rate, high error rate, asymmetry
  - High propagation delay
    - Due to link latency (in space, under water), intermittent connectivity

Interactivity

- Node and environmental constraints
  - Lifetime, availability, density, processing capabilities
  - Non-availability of infrastructure

Capability

- Change communication semantics, application paradigms

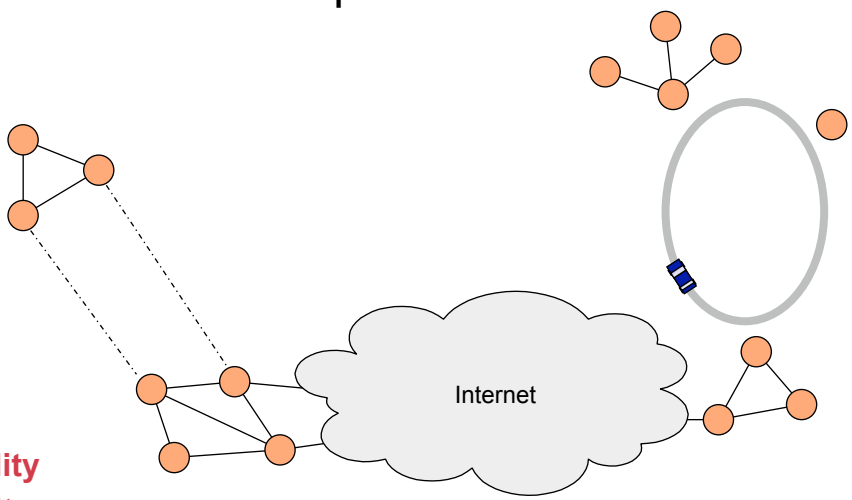
One Approach:

Delay-tolerant Networking (DTN)

The Architecture Developed by the DTN Research Group (DTNRG) in the Internet Research Task Force (IRTF)

http://www.dtnrg.org/

Some Sample Scenario



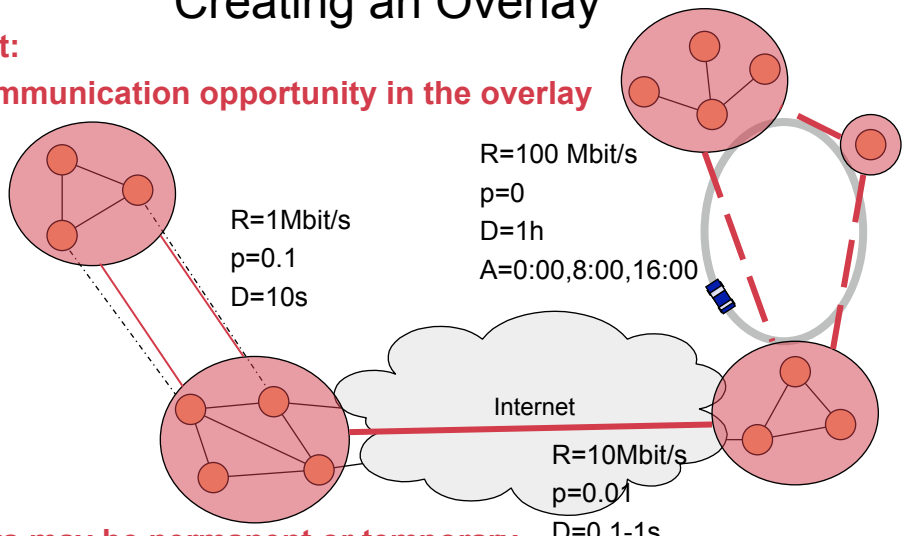
Reachability
Interactivity
Capability

Avoid (the Need for) Synchronous Communications

- Delays may be too long for interactive protocols
  - We have seen that RTTs in the order of seconds are already bad
  - How about RTTs of minutes or hours or even days?
- An end-to-end path to a peer may never exist
  - At least not at the order of time IP routers and end systems operate
- Delay tolerance implies disruption tolerance
  - If a peer, a link, or a path is currently not available, just wait until it comes back
  - Store the “packets” in the meantime
  - Or hand the data to someone else who may have better chances of delivery

Creating an Overlay

Contact:
any communication opportunity in the overlay



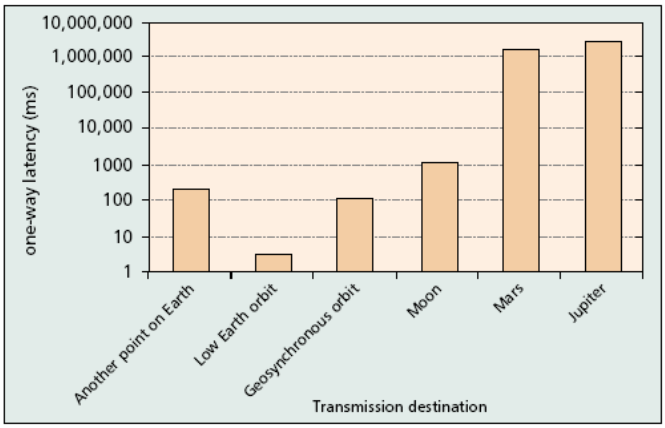
Contacts may be permanent or temporary,
Long or short-lived, scheduled or opportunistic, …

Revisiting Communication Paradigms

- Use only asynchronous communications
  - Simply modeled after email
  - Store and forward: wait for the next suitable opportunity to send
  - Store, carry, and forward: add physical data carriage as communication option
- Decouple sender from receiver as much as possible
  - Realize end-to-end semantics where it belongs: at the application layer
  - Requires dedicated (delay-tolerant) protocols, applications, and users

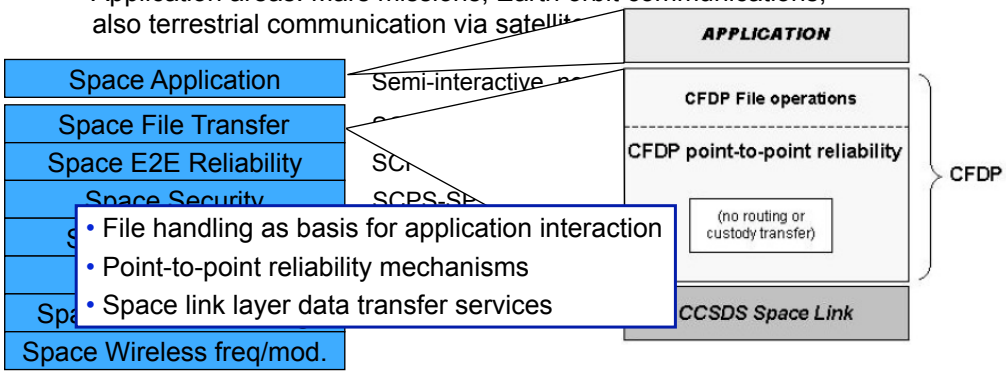
Origin: Space Communications

- ping moon.earth.sol
  - ~2,200ms
- ping mars.sol
  - ~2,200,000ms
- ping pluto
  - Distance 4.28E9 km
  - > 14,270 s
  - = 4 hours

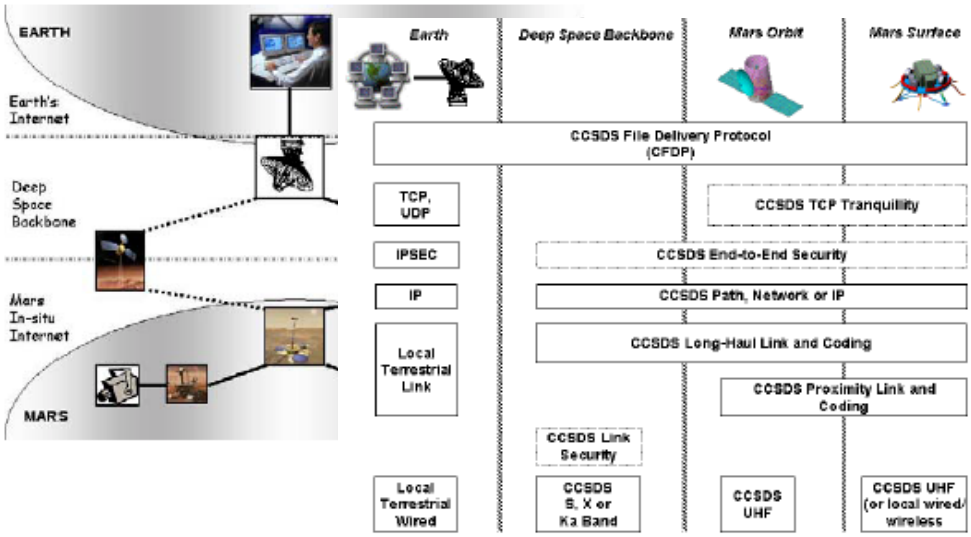


Origin: Space Communications

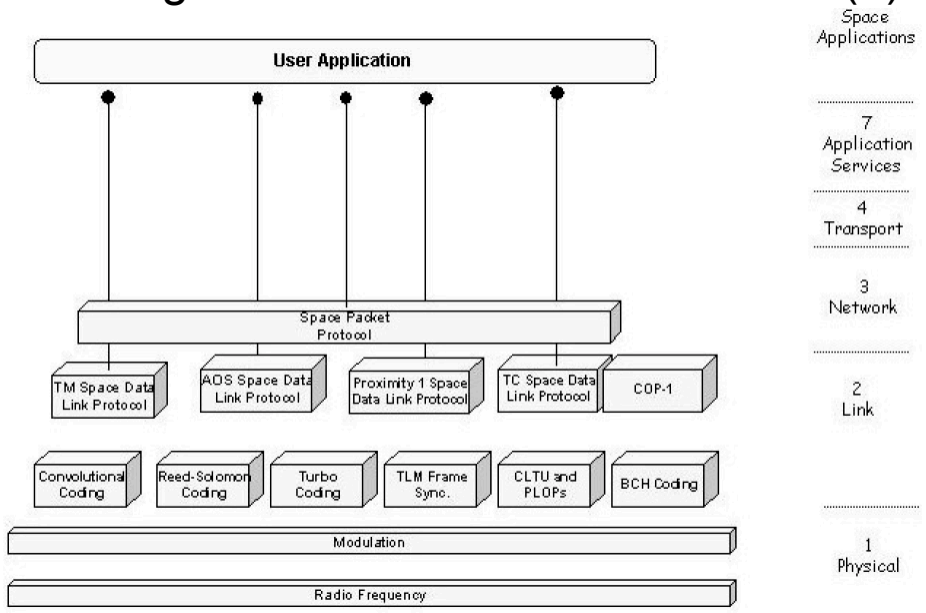
- Consultative Committee for Space Data Systems (CCSDS)
  - Defined specific protocol suites for space communications
  - Highly tailored towards long delay and error-prone transmissions
  - Later versions leverage Internet technologies
  - Application areas: Mars missions, Earth orbit communications, also terrestrial communication via satell



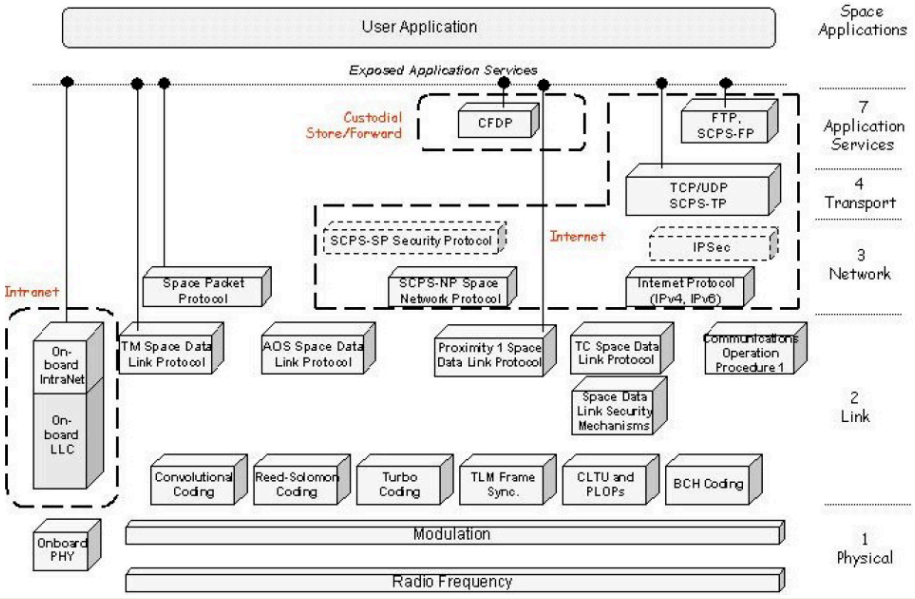
Example: Mars Mission



Evolving CCSDS Protocol Architecture (1)



Evolving CCSDS Protocol Architecture (2)



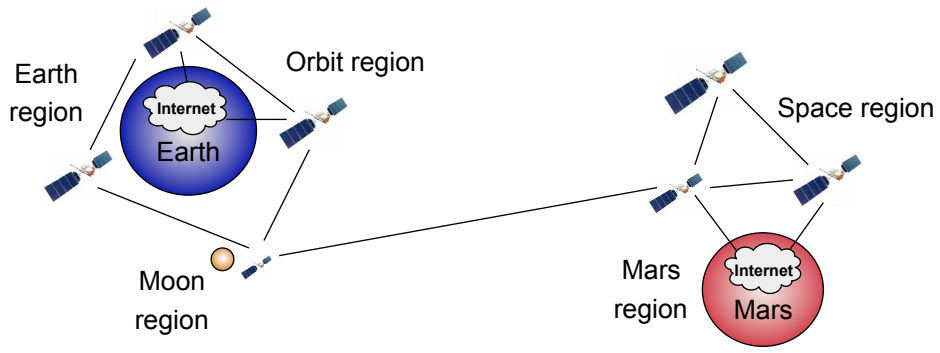
Towards the Interplanetary Internet

- Interplanetary Internet (IPI)
  - Development since late 1990s
  - Expanding internetworking to interplanetary scale
  - Motivation: Allow some degree of interoperability between different missions (countries, vehicles, applications, etc.)

- Improvements over CCSDS
  - Support for more flexible applications beyond just file transfer
  - Improve modularity of the system design
  - Improve on reliability (custody transfer)
- In essence: generalize towards an evolvable architecture
  - Combining local terrestrial (or Marsian, …) communications
  - With interplanetary communications
  - Provide suitable application support

IPI Architecture

- Network of regional internets



- Message-based communications
- Basic concepts for most of the DTN architecture in place

http://www.ipnsig.org/reports/draft-irtf-ipnrg-arch-01.txt

Generalizing IPI…

- Not all communications in a local environment will be able to use IP
- Moving from stiff region to more flexible structures
- Maintain the basic message-based communication properties

1. From Packets to Messages

- Reminder: IP packets are self-contained wrt routing
  - Multiplexing, independent routing decisions, drop granularity (best-effort)
- Need often many IP packets for an application exchange
  - Transport protocols (TCP), Application Data Units
- Asynchronous communications requires self-contained messages
  - Limited end-to-end interactivity (RTT!): cannot have handshakes
  - Self-contained messages may but need not be large
    - A few bytes for a meter reading
    - A gigapixel image from the solar system or a planet or a DVD
  - Still semantic fragmentation at the application layer useful (cf. RTP)
    - Lower layer fragmentation may be needed due to contact times
- Store-and-forward granularity
  - Useful for buffer management in intermediaries
  - Cannot easily repair loss of individual packets (again: RTT)
  - Need to keep contents together

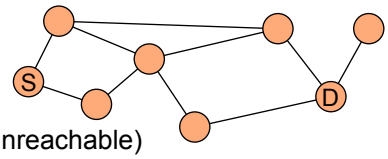
Think email!

2. Store, Carry, and Forward

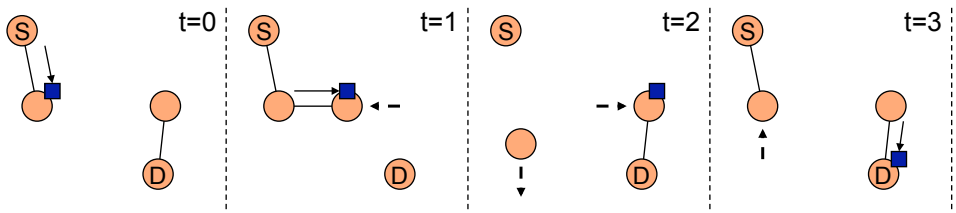
- Hop-by-hop message relaying (cf. email)
  - Receive a message (completely)
  - Store it (in memory or on persistent storage)
  - Perform a routing table lookup
  - Forward the message to the next hop
- Storing may need to be done for an extended period of time
  - If there is no next hop known
  - If the link to the known next hop is not yet available
  - Buffer management becomes important
  - Congestion control becomes really tricky
- Nodes may move while storing a message
  - Physical message carriage added to forwarding
  - May move large amounts of data over arbitrary distances
  - Even short distances may be essential for ultimate forwarding success

3. Routing in the Time-Space Domain

- Traditional routing uses instantly existing paths only
  - Run link-state or distance vector protocol
    - Metrics and weights define preferred path
    - Few optimizations: reachability is key
    - Well: load balancing, traffic engineering as admistrative way to steer traffic
  - If there is no path: drop the packet (ICMP unreachable)



- Delay-tolerant routing must consider future paths
  - Store messages until the next hop becomes available
  - Links may come up and down for many reasons (incl. motion)



4. Naming and Late Binding

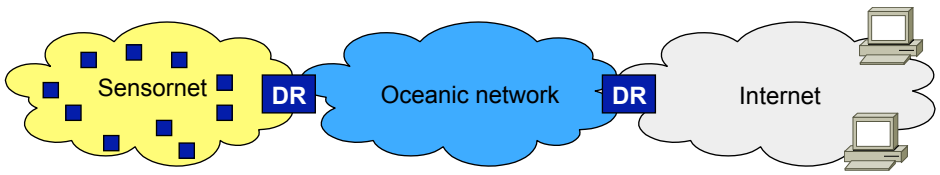
- Name resolution is impractical
  - Cannot always wait until a resolution server (e.g., DNS) becomes available
- Defer the resolution to the node as long as possible
  - Route based upon the name
  - Defer routing decisions to other areas of the networks
    - If no information is available locally
    - Default routing potentially at larger scale
  - Perform the mapping
- Side effect: if names are long, large messages preferred over small packets to reduce the overhead
- Allow multiple naming schemes to co-exist
  - Do not enforce a particular naming scheme on applications
  - Support diversity

5. Layer-Agnostic Internetworking

- Obvious need to interconnect different networks
- Widely varying capabilities
  - May or may not be able to run IP
  - May just run L2 protocols
  - May run a vertically integrated protocol stack (sensor network)
- Provide a common messaging abstraction to communications
  - Define mapping to different lower layers
  - Entirely different protocol stacks may be used on individual hops
  - Only the DTN message structure is preserved (like an email message)
    - You can transfer email using SMTP, POP, IMAP, NNTP, FTP, HTTP,
    - Remember UUNet?
      - Hop-by-hop transmission using UUCP over serial lines and modems
      - X.25-based hops

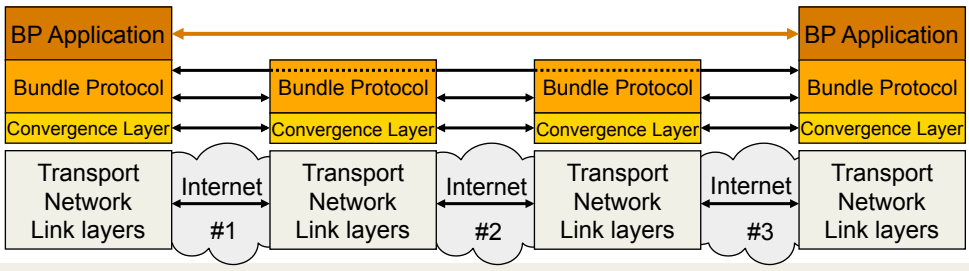
DTN RG Architecture (1)

- Purpose: asynchronously interconnecting different internetworks
  - Which may be based upon arbitrary underlying technologies
  - Which may encompass just a link layer technology or a complete protocol suite
  - Which may cross different administrative boundaries
  - Which may be used for different (presently unforeseen) applications with diverse requirements
  - Which cannot necessarily rely on an always accessible infrastructure
- Example

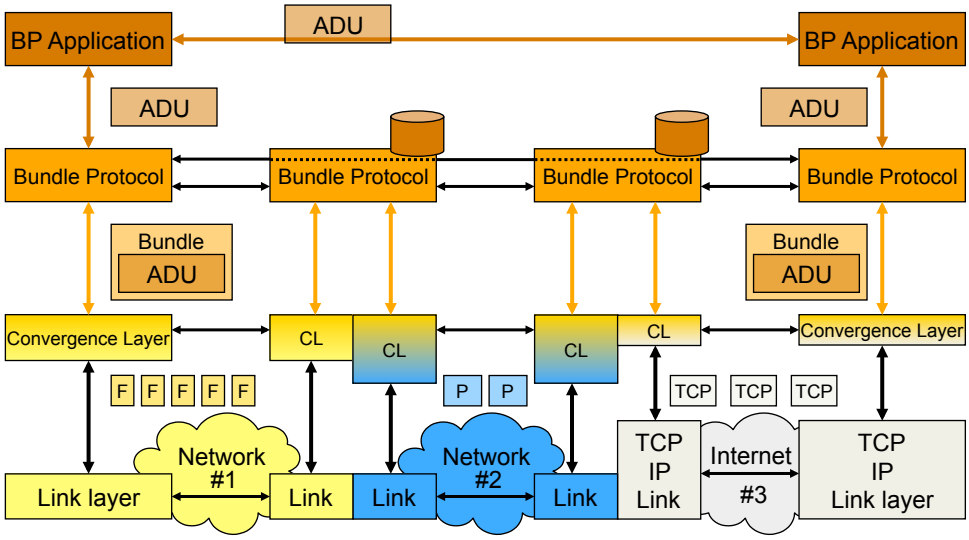


DTN RG Architecture (2)

- Applications exchange Application Data Units (ADUs)
  - Semantically meaningful pieces of information (=messages)
- **Bundle** as communication unit encapsulate ADUs
  - **Bundle layer** on top of underlying networks using **Bundle Protocol (BP)**
  - Above the transport layer in the Internet (and similar architectures)
  - Or above the link layer
- Mapping to lower layers defined by “convergence layer”

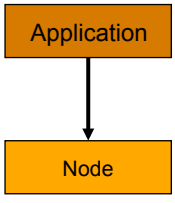


DTN RG Architecture (3)

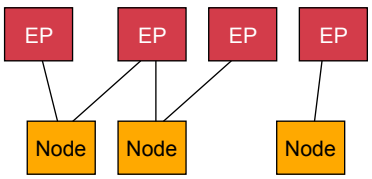


Nodes and Endpoints

- DTN node (short: node)
  - An entity implementing the bundle protocol
    - Sometimes also referred to as Bundle Protocol Agents (BPAs)
    - Similar to IP nodes (=hosts and routers)
  - Applications use DTN nodes to send and receive ADUs



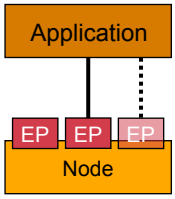
- DTN endpoint: set of one or more DTN nodes
  - Minimal reception group (MRG): subset of a DTN endpoint
  - Defines communication semantics
    - One node: unicasting
    - One node of a group: anycast
    - Multiple nodes of a group: multicast, broadcast



- Endpoint identifier (EID)

Naming and Addressing

- Endpoint Identifier (EID)
  - The “name” of an endpoint to be used for routing and addressing
  - Singleton: one unique EID of a node (or an application instance)
    - Endpoint comprising exactly one DTN node
    - Each node has at least one singleton EID
  - Other EIDs may be shared: multicasting, broadcasting, anycasting
    - Larger endpoint groups, different semantics
- EID: a Uniform Resource Identifier (URI)



- Currently: name = address
  - No separation of identifier and locator defined
  - Routing takes place based upon the EID
  - Different interpretations conceivable depending on the URI scheme

URI Schemes

- EIDs may use arbitrary URI schemes
- Example: DTN scheme
  - dtn://none
  - dtn://<some opaque string>
  - dtn://host.domain/some-further-id
  - http://www.netlab.tkk.fi/
  - mailto:jo@netlab.tkk.fi
- Semantics and interpretation still open
  - No naming conventions defined yet how to identify applications, application instances, higher layer protocols, target network domain
  - Address space divided into schemes which have to define their usage
- Late binding
  - No address mapping or resolution needed
  - Routing takes place based upon complete URI: sender “just sends”

Time

- DTN nodes require a rough notion of time
  - Modestly synchronized clocks
- Bundles contain the creation timestamp
- Bundles have TTLs
  - Expressed in absolute time, related to a reference clock
  - Used for bundle expiration
- Traditional time synchronization mechanisms not applicable in a general fashion
  - NTP synchronization is in the order of RTT (which may be huge)
  - Peerwise synchronization during contacts covers subsets only with partially connected networks
    - "Who is right?" if two nodes disagree
  - …

Bundle Services: Endpoint Registration

- Application Registration (EID)
  - Local bind()ing to a specified EID at a DTN node
- Unicast, multicast, and anycast support
  - Uniqueness of names not enforced
  - An application may not know whether or not its EID is unique

Bundle Services: Bundle Transfer

- Bundle transmission
  - Bundles may in theory be of arbitrary size (few bytes to many terabytes)
  - Default transfer is best effort
    - Losses, re-ordering, duplication
  - Storage for an extended period of time (if necessary)
- Transmission priorities
  - Define relative forwarding priority at each node
    - Coarse prioritization
  - Bulk < Normal < Expedited
  - Chosen by the application
    - Per-source node classification
    - No common policies defined across multiple nodes
- Time-to-Live (TTL)

Bundle Services: Transfer & Reporting

- “Postal-style” (email) delivery options
- Reporting
  - Bundle delivered to the destination node (“return receipt”)
  - Bundle acknowledges by the target application
  - Bundle reception, forwarding, delivery, deletion
  - Application end-to-end acknowledgement
  - Diagnostic reporting
    - Bundle received at an (intermediate) node
    - Bundle forwarded an (intermediate) node
    - Bundle deleted (queue full, TTL expired)
  - Reports sent to the source or an explicitly specified EID
    - Reporting limited for multicasting/broadcasting
- Security-related options
  - Confidentiality, authentication required
  - Error detection

Bundle Services: Custody Transfer

- Custody transfer
  - Motivation: create hop-by-hop reliability
  - A node may decide to accept custody (= responsibility) for a bundle
    - Bundle will be stored on persistent storage (and thus survive a reboot)
    - Bundle will not be deleted until a node further down the path has accepted custody
  - Custody nodes may be multiple hops apart
- Application control
  - Custody requested
  - Source node custody requested
- Application reporting
  - Custody acceptance
  - Custody transfer

DTN Applications

- Applications should minimize the number of round-trip exchanges.
- Applications should cope with restarts after failure while network transactions remain pending.
- Applications should inform the network of the useful life and relative importance of data to be delivered.

DTN Applications

- Well… nothing standardized defined yet
  - Applications are mostly used in closed systems
  - Focus on common infrastructure
- Application data simply placed in bundles
  - Example: file transfer application
- Implicit identification of application by means of EIDs
  - No hierarchical demultiplexing
  - No explicit content indication
  - Must all be handled by the application
- One (inefficient) option for moving forward
  - RFC 2822 headers
  - MIME for content identification, encoding, handling, etc.
  - S/MIME for end-to-end security
- Yet, conventions needed (working on it)