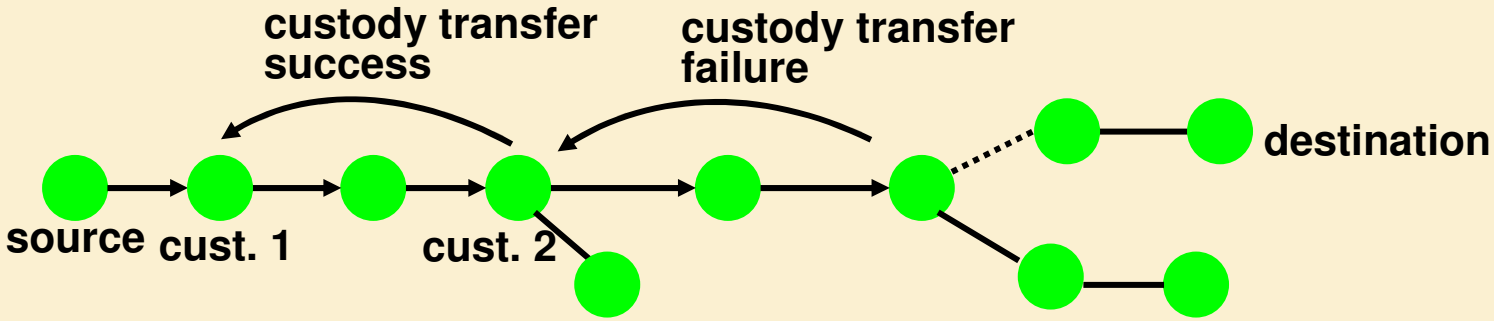
# Custodial Multicast in Delay Tolerant Networks

Susan Symington
Robert Durst
Keith Scott
susan@mitre.org, durst@mitre.org, kscott@mitre.org

# Support for Multicast in DTN

- **Basic DTN message delivery service is unacknowledged and is not guaranteed**
- **The Bundle Protocol supports unacknowledged delivery to singleton and multicast endpoints.**
- **The Bundle Protocol supports custodial delivery of bundles sent to singleton endpoints.**
- **Support for custodial delivery of bundles sent to multicast endpoints is not defined**
- **Internet Draft** (Symington, Durst, Scott)
  - Defines extensions to the Bundle Protocol to support custodial transfer of bundles sent to multicast endpoints.
  - www.ietf.org/internet-drafts/draft-symington-dtnrg-bundle-multicast-custodial-01.txt

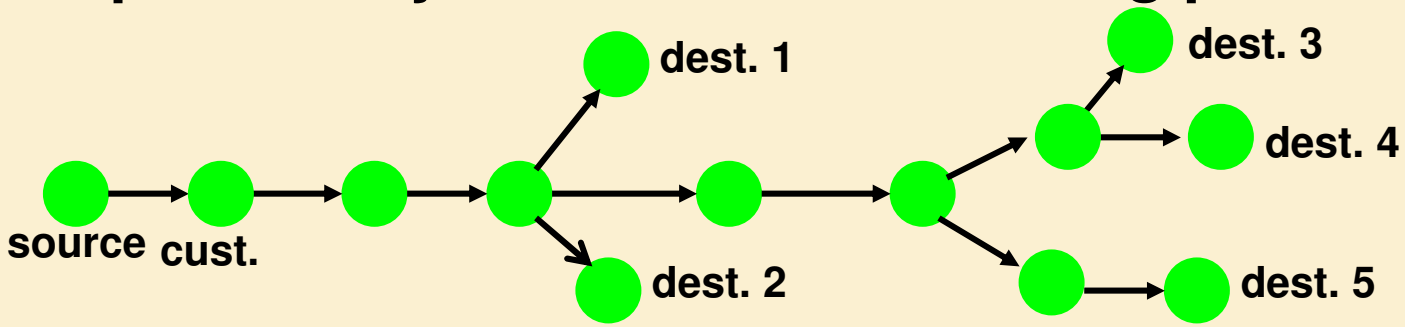
# How Custodial Unicast Works




- Bundles may have a "custody transfer requested" bit set.
- A DTN node that receives a bundle with this bit set may:
  - Take custody of the bundle by sending a custody transfer success signal to the previous custodian and starting a retransmission timer
  - Forward the bundle without taking custody, or
  - Send a custody transfer failure signal to the previous custodian, indicating that the bundle can neither be forwarded nor stored
- A custodian should store the bundle in persistent storage until custody is transferred or the bundle is delivered
- A custodian may retransmit the bundle in response to a failed custody signal or retransmission timer timeout
- When bundles are unicast, there is only one copy of each bundle for a custodian to keep track of

# How Multicast Complicates Custodial Delivery

- Any node (custodial or non-custodial) may be a branching point for the bundle
- Custodians are not aware of downstream branching points or how many times a bundle is copied at any downstream branching point




- The custodian of a bundle is responsible not only for the copy(ies) of the bundle that it forwards, but for all copies of that bundle that may be created at other nodes downstream, until a new custodian takes over or the copy is delivered.
- Need a mechanism to enable a custodian to determine when all downstream copies have either been delivered or have been taken custody of.

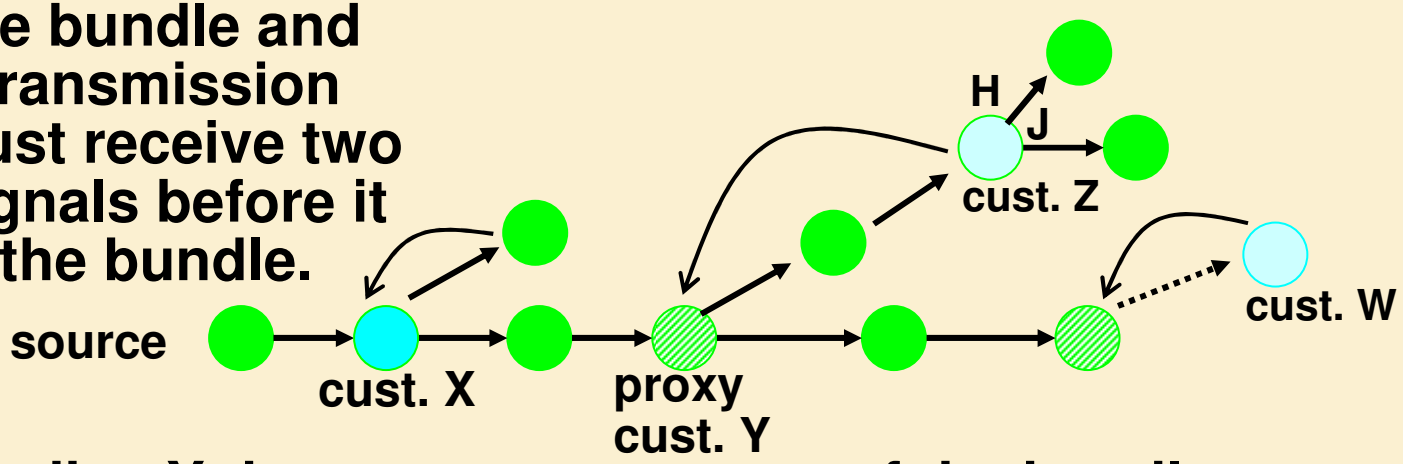
# Custodial Multicast Objectives

- **Increase the likelihood that the bundle will be delivered to its destination endpoint before expiration**
  - **to as many nodes as possible in its destination endpoint**
- **Reduce the cost of bundle retransmissions**
  - **Want the cost (in terms of the routing metric in use) of re-forwarding from the custodian to be less than the cost of re-forwarding from the source and (ideally) previous custodians**
- **Enable delivery to late-registering nodes***
  - **Custodians can deliver bundles to nodes whose registration request hadn't yet propagated sufficiently to be grafted onto the distribution tree when the bundle was originally sent**

***This objective is not addressed by the mechanisms presented**

# Bundle Protocol Extensions for Supporting Custodial Multicast (Overview)

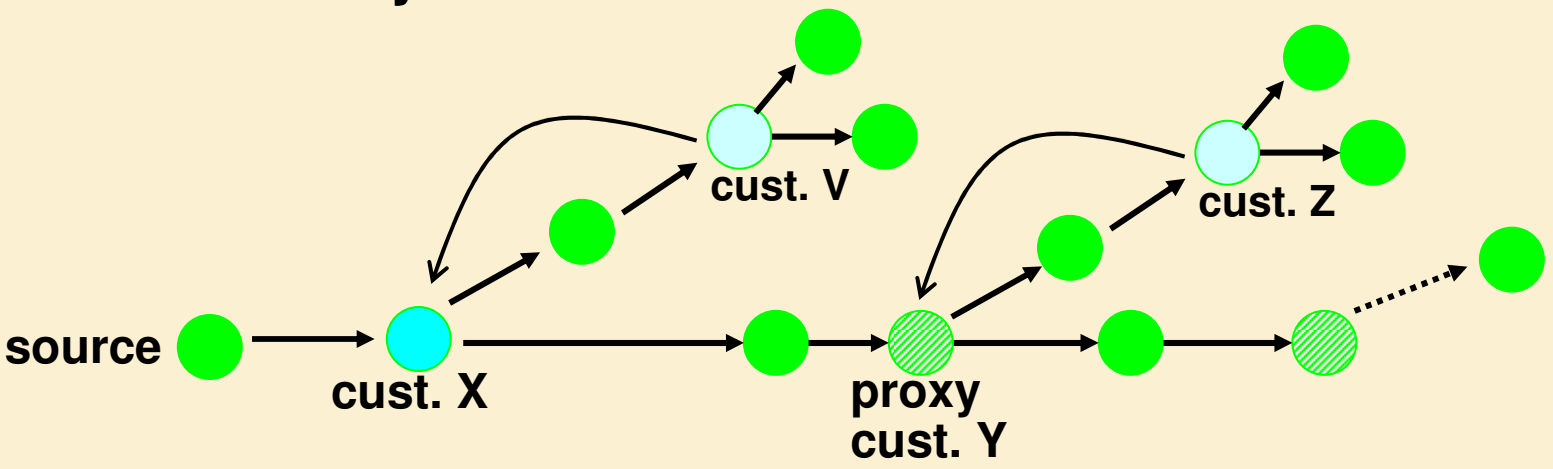
- X stores the bundle and starts a retransmission timer. X must receive two custody signals before it can delete the bundle.




- Proxy custodian Y does not store a copy of the bundle, but inserts its own EID in the bundle's custodian field. It expects to receive two custody signals.
- If Y receives two Success signals, it sends a single custody success signal to X.
- If Y receives one or more custody Failure signals, it sends these (possibly aggregated) to X.
- If X's timer times out or X receives a custody Failure signal, X retransmits the saved bundle.
- Y should retransmit on only those branches for which it did not receive a Success signal (copy IDs are useful)

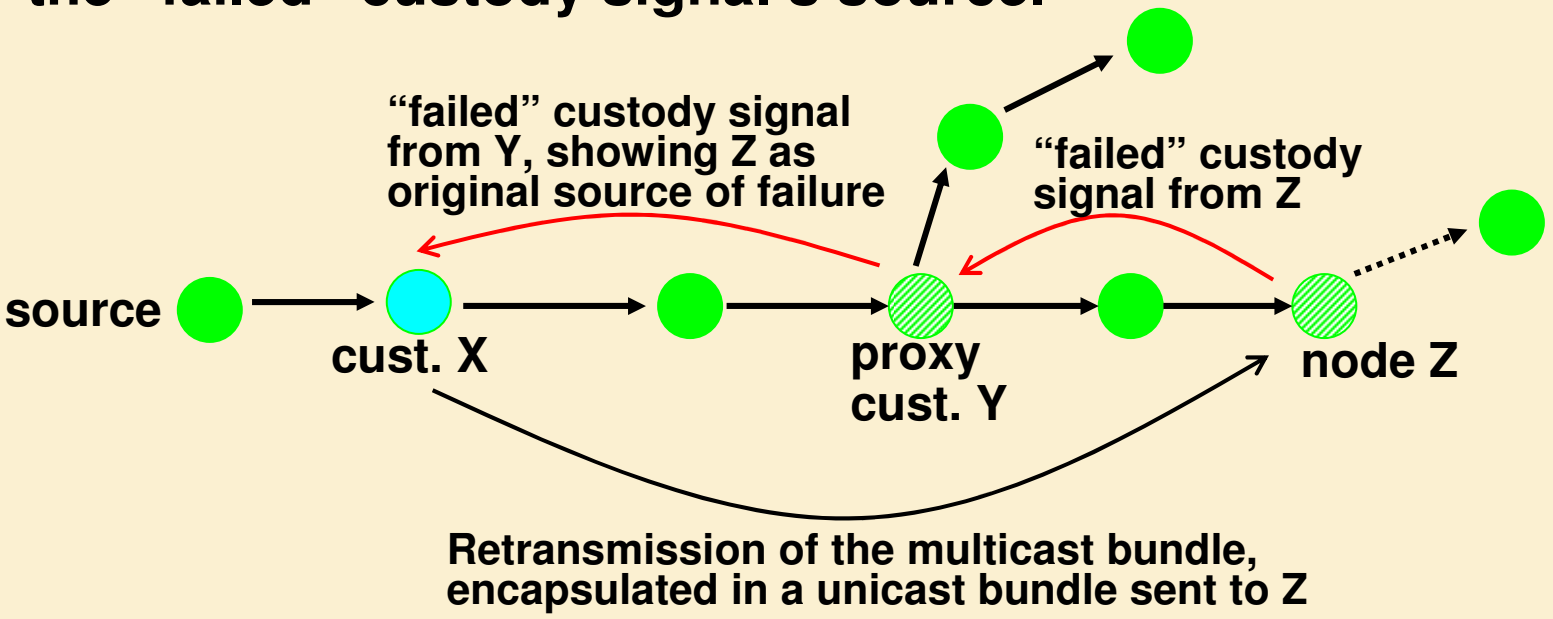
# How custodians know when they can delete a bundle

- When a custodian receives a “succeeded” custody signal for a copy of the bundle that it forwards, this indicates that all copies downstream of that copy have been either delivered or taken custody of.
- When a custodian receives a “succeeded” custody signal for every copy of the bundle that it branched, it may delete the bundle from storage.
- Until a custodian receives a “succeeded” custody signal for every copy it branched, it must assume that there is at least one downstream copy that has neither been delivered nor taken custody of.



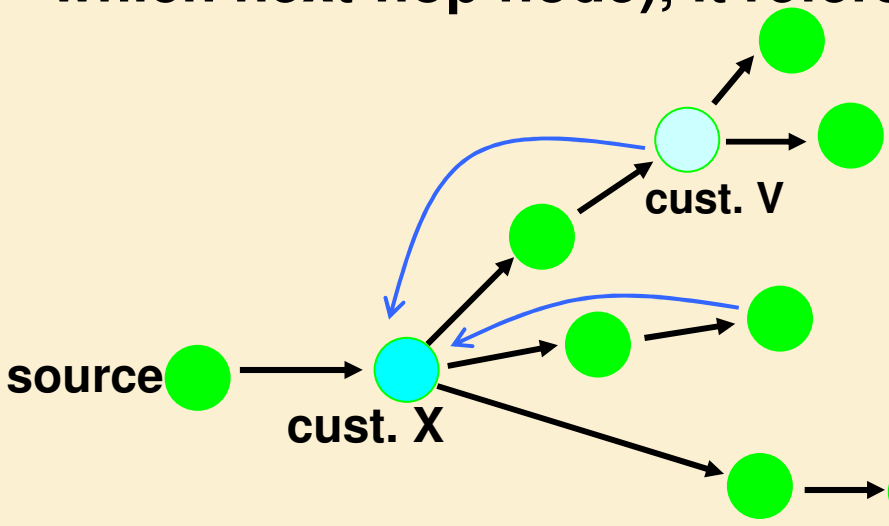
# Efficient Re-forwarding

- When a proxy custodian receives a "failed" custody signal, it must generate a replacement "failed" custody signal and insert a Proxy EID extension block into this bundle (if there is not one in there already) that identifies the source of the original "failed" custody signal.
- When a custodian receives a "failed" custody signal, it retransmits a copy of the bundle referred to by the "failed" signal, perhaps encapsulated in a unicast bundle sent to the "failed" custody signal's source.



# Efficient Re-forwarding (continued)

- When a custodian's retransmission timer expires, it retransmits a copy of the bundle associated with that timer to the next-hop node(s) associated with that timer.
- Ideally, it retransmits only to those next-hop nodes from which it has not received a successful custody signal
- If a custodian marks the copies of a given bundle that it delivers or forwards individually, then when a custody signal is received, the receiving custodian can determine not only which bundle, but which copy of the bundle (e.g. which next-hop node), it refers to.




- The EID that a custodian inserts into the bundle's custodian field can differ for each copy forwarded so as to act as a copy discriminator.

# Custodial Multicast Extensions are Optional

- Not all nodes in the delivery tree must implement the optional custodial multicast extensions
- Non-custodial-multicast-capable nodes can deliver custodial multicast bundles
- Non-custodial-multicast-capable nodes can forward custodial multicast bundles to a single next hop
- Only custodial multicast capable nodes may be custodians and branching points
- The multicast routing protocols must enforce the restriction that only custodial-multicast-capable nodes may be branching points

# DTN Multicast Open Issues

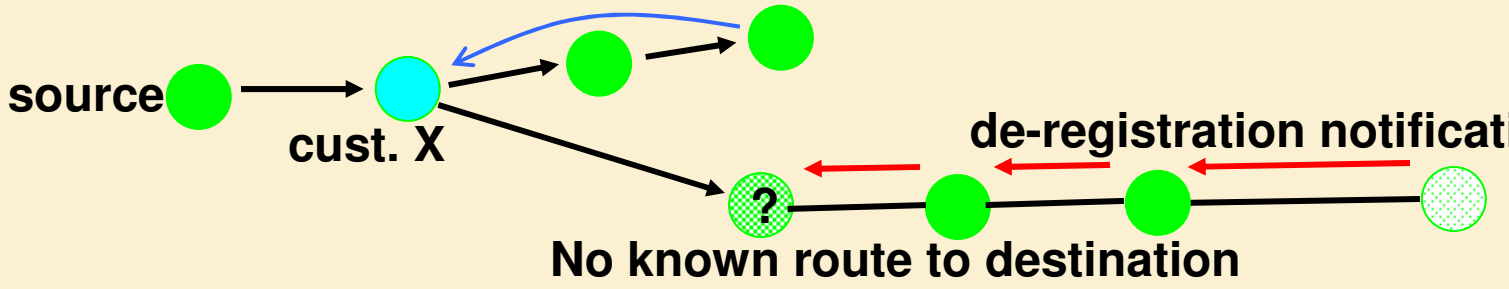
- **Accommodating nodes that register late or whose registration requests are delayed** (W. Zhao, M. Ammar, E. Zegura)
- **Multicast routing protocols are not yet defined for DTN**
  - Multicast routing protocols for supporting custodial multicast must meet certain requirements
    - Must operate over trees rather than meshes
    - Only custodial-multicast-capable nodes may be branching points
    - If convergence-layer multicast is used, the forwarding node must know the number of next-hop receivers
- **Security protection for multicast bundles**

# Backup Slides

## Backup slides

# Forwarding Failures

- As mentioned earlier, there may be a delay between when a node registers and when that registration propagates to graft that node onto the distribution tree.
  - As a result, custodians may need to re-forward bundles to "late-joining" nodes.
- Analogously, there may be a delay between when a node de-registers and when that de-registration propagates to prune that node from the distribution tree.
  - A custodian could forward a multicast bundle that arrives at a downstream node with "no known route to destination".
  - This should generate a "success" custody signal because there are no remaining downstream copies of the bundle that need to be either delivered or taken custody of.



# Assumptions and Design Principles

1. **Branching point nodes are not required to be custodians**
2. **Branching point nodes are required to maintain state for each custodially-transferred bundle that they branch (even if they are not custodians)**
   - **Must keep track of the delivery/custody status of each of the copies that it creates and report this to the nearest upstream custodian or branching point node**
3. **DTN multicast routing protocols must enforce certain restrictions when supporting custodial delivery**
   - **Only custodial-multicast-capable (CMC) nodes may be branching points**
   - **Routing protocols must operate over trees rather than meshes**

# Assumptions and Design Principles (continued)

4. **Convergence Layer Multicast may be used, with some restrictions**
   - The node that is forwarding using convergence layer multicast must be CMC (because this node is a branching point), and
   - The forwarding node must know the number of next-hop nodes the bundle is expected to reach via that convergence layer
5. **Bandwidth conservation is given priority over robustness of delivery by default, but local policy may override this**
   - Custodians and branching nodes do not forward bundles to next-hop nodes from which successful custody signals for those bundle have already been received