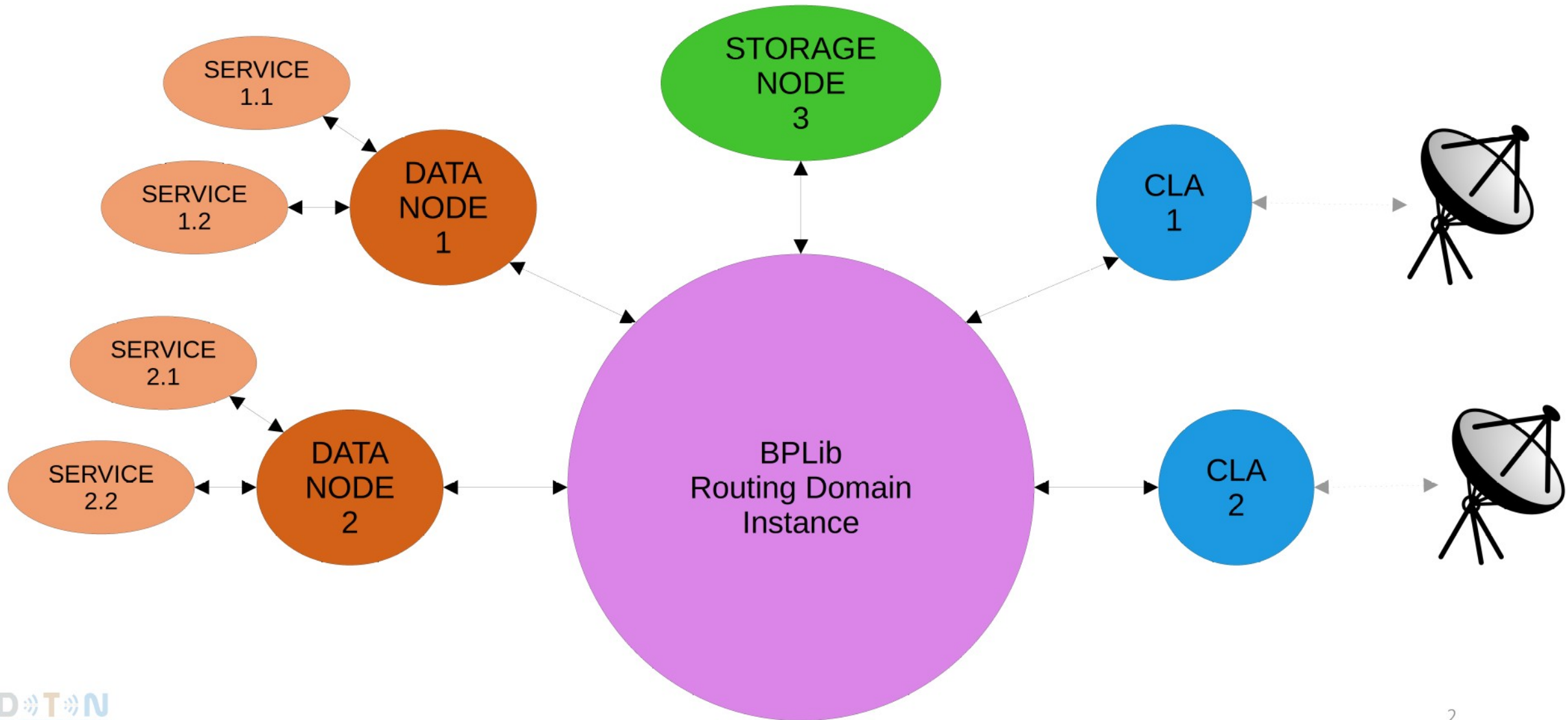




DELAY/DISRUPTION TOLERANT NETWORKING

BPv7 Prototype Architecture

Software Elements



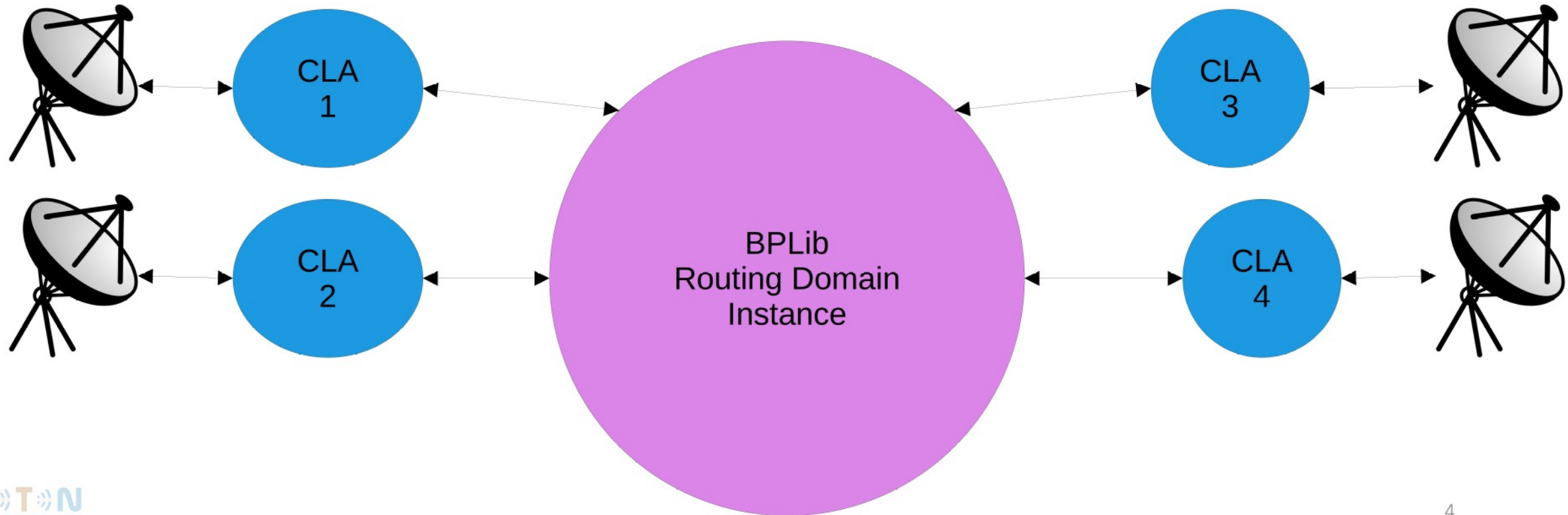
- Each routing domain has a pool of memory/RAM associated with it
 - Provides temporary storage as bundles/blocks are generated, decoded, or passed around internally
 - Establishes a cap for the amount of RAM this instance will ever use
 - Pool divided into fixed-size blocks
 - Fixed size avoids fragmentation problems
 - Blocks can be chained together for larger objects
- All bundle I/O done by way of logical “interfaces”
 - Interfaces (Data, Storage, CLA) are created by SW separately and independently
 - Routes are then created based on destination IPN (analogous to IP address)
 - Units passed between interfaces are bundles (Primary block + canonical blocks “in tow”)
 - All interfaces have the notion of “Admin” and “Operational” states
 - Admin state of the interface is based on current policy
 - Operational state of the interface indicates whether it is actually available for data flow (Link UP, carrier present, etc)

Storage-Less Relay Node



Storage-Less entities such as this only forward bundles

- Cannot accept “custody” themselves – but can pass thru to another node that has storage
- Note, still has RAM-based temp storage for queued bundles as they pass thru, awaiting contacts

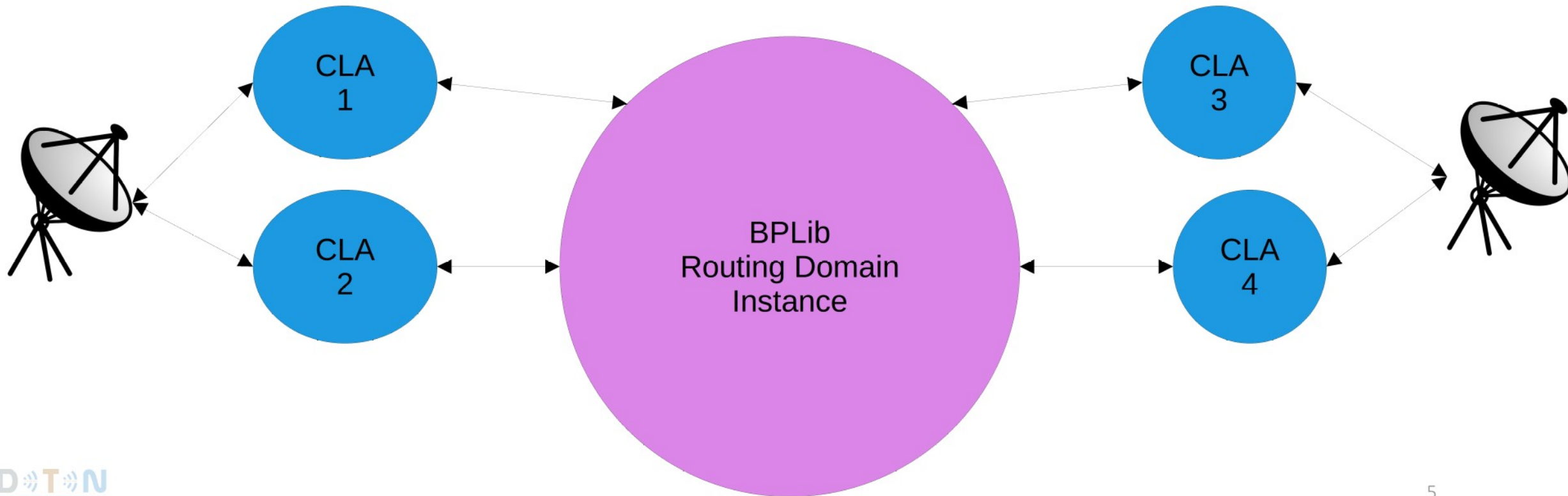


Multiple CLA on single radio



Routing is done on a CLA “interface” level

- Each CLA intf has its own set of reachable node numbers
- But multiple CLAs can feed back to a single radio

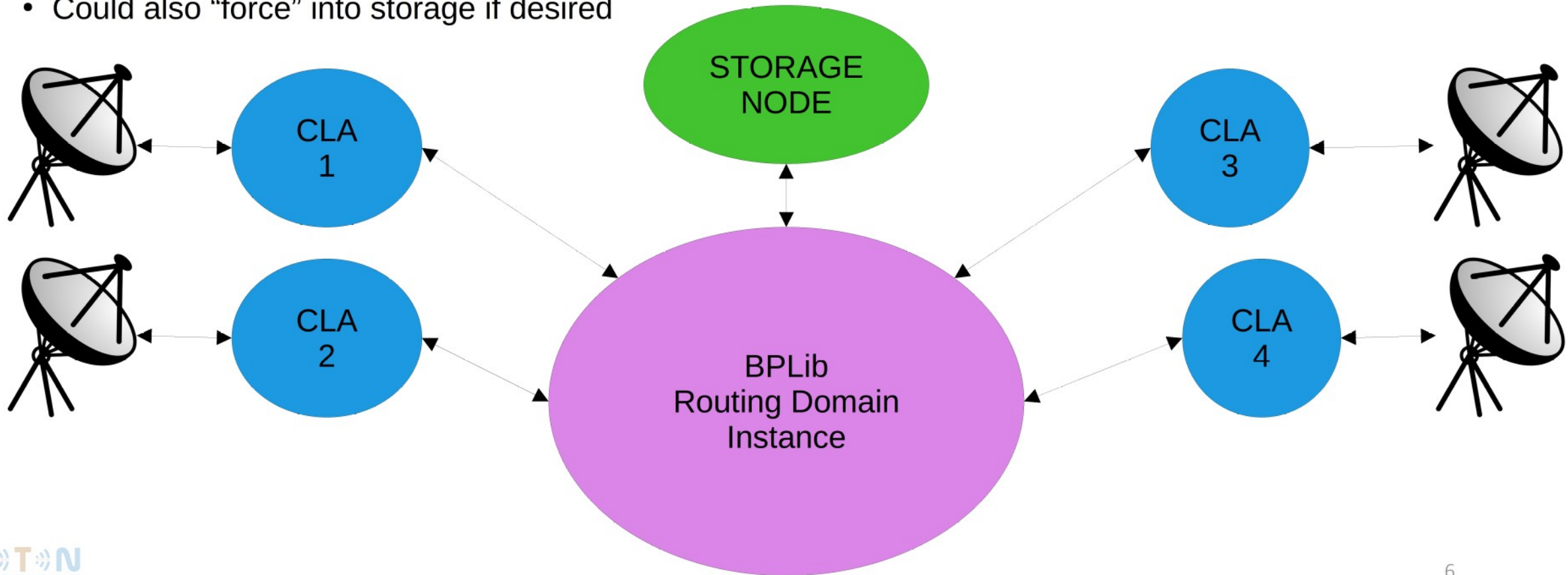


Relay Node with Storage



Entities such as this may forward bundles or accept custody

- Based on current operational states of CLA, bundles may be forwarded from CLA to CLA or Storage
- If bundle arrives at storage, custody signal may be sent back to sender
- However if the next hop is “up” bundle may go direct to another CLA (bypassing storage/custody)
- Could also “force” into storage if desired



Software Concepts: Storage Service



- Storage service is conceptually an interface like the others
 - Route to storage are configured like any route
 - Bundles are delivered to the storage service based on currently configured routes through BPLib
 - Would typically be analogous to a “default route” in IP, so bundles will be sent to storage if no other egress interface is available
- Bundles are stored in RAM when delivered to/from storage service
 - Same as all other interfaces (bundles in transit)
 - Storage Service maintenance task can offload bundles from RAM to more persistent memory (disk, flash, etc)
 - This frees up RAM blocks for other use
 - Custody should only be claimed once offloaded to persistent memory

- Services are implemented as “sub-interfaces” on top of data interfaces
- On receive, This further routes the data based on other fields aside from node number
 - All bundles with a destination node ID matching the node number are first delivered to the data interface
 - Then the next item is used to route to a sub-interface, in this case service number
- On transmit, information from the service interface is used to assemble the bundle
 - Info such as source/destination EID, lifetime, etc.
 - When creating the flow/service the application would have to supply this info
- Other sub-interface types can be created based on any “route-able” information in the bundle.

