

# Architecture of Bundle Protocol

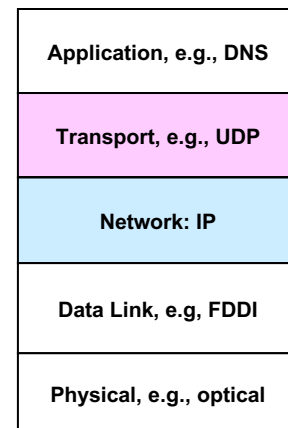
Scott Burleigh, IPNSIG

# It's Not Really New

- The DTN architecture is much like the architecture of the Internet, except that it is one layer higher in the ISO protocol “stack”.
- The Bundle Protocol (BP) is designed to function as an “overlay” network protocol that interconnects “internets” – including both Internet-structured networks and also data paths that utilize only space communication links – in much the same way that IP interconnects “subnets” such as those built on Ethernet, SONET, etc.
- Just to review this terminology a little....

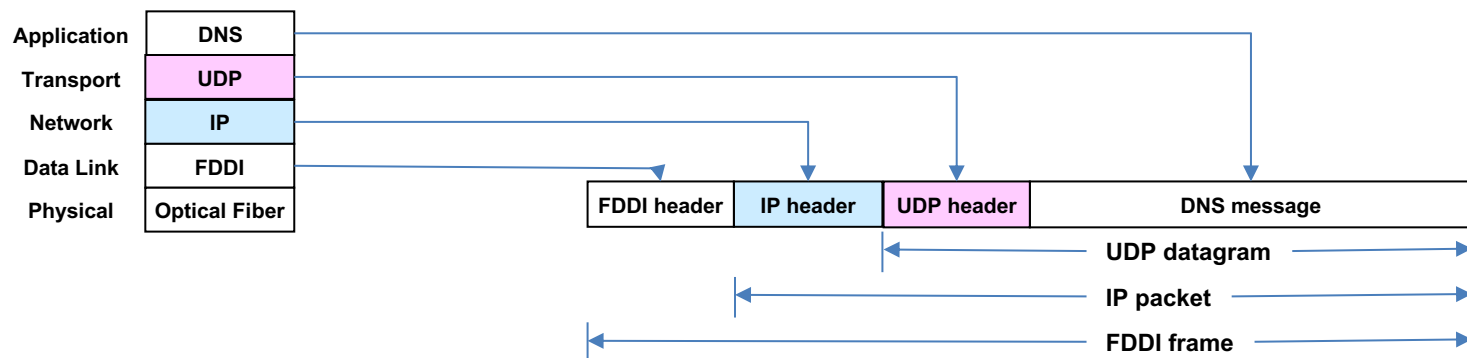
# Protocol Stacks (1 of 2)

- In network design, we allocate different functions to different protocols.
- Those different protocols work together in a nested fashion, with each one using the services of another and providing service to yet another.
- We represent these service relationships graphically in a *stack*, where each layer of the stack corresponds to a function.
- In a protocol stack diagram, each layer of the diagram identifies the protocol that performs the function at that layer:
  - It relies on the services of the layer below it.
  - It provides service to the layer above it.
- The Internet protocol stack looks like this:



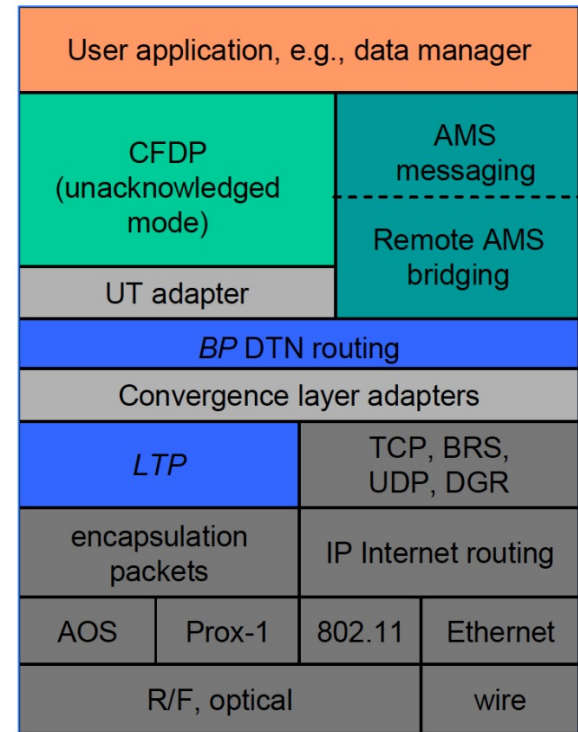
# Protocol Stacks (2 of 2)

- The stacking of protocols is reflected in the structure of the *protocol data units (PDUs)* that are transmitted over the network:
  - A data unit of protocol N is usually formed by prepending a protocol-N header to a data unit of protocol N + 1, which it is said to *encapsulate*.
  - The protocols headers in a PDU echo the layers of the stack, except rotated 90°.



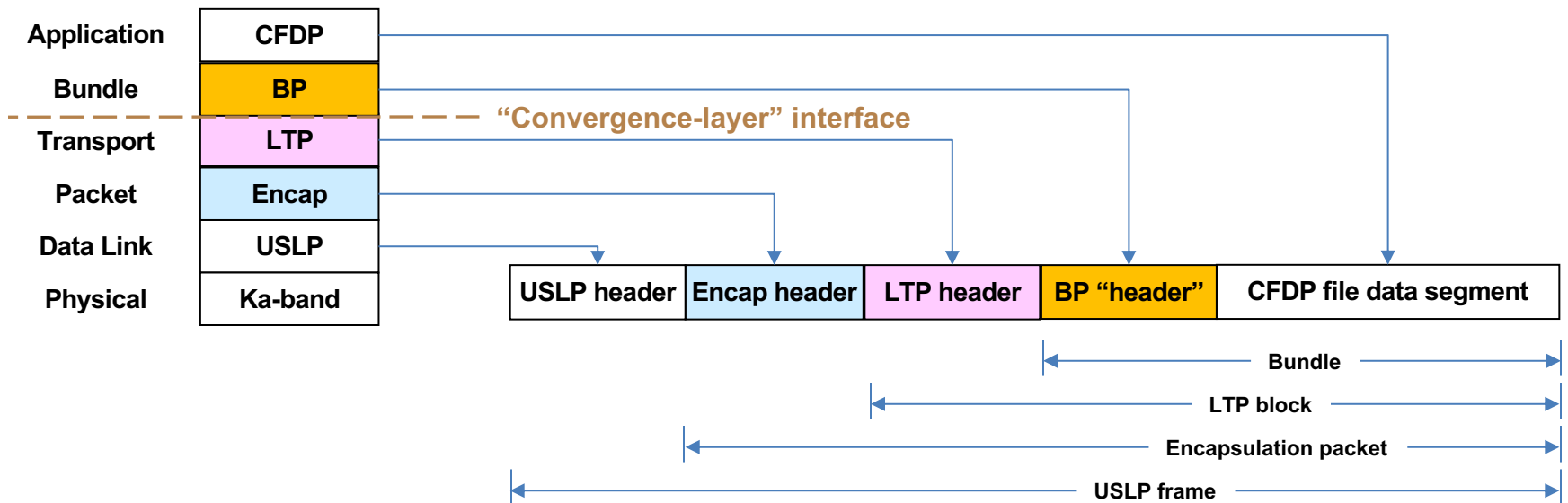
# DTN Works the Same Way

- Data traversing a DTN are conveyed in DTN *bundles* – which are functionally analogous to IP packets – between BP *endpoints* which are functionally analogous to sockets.
- A single BP *node* may communicate via multiple BP endpoints, just as a single IP node (host or router) may communicate via multiple sockets.
- The DTN protocol stack looks like this:

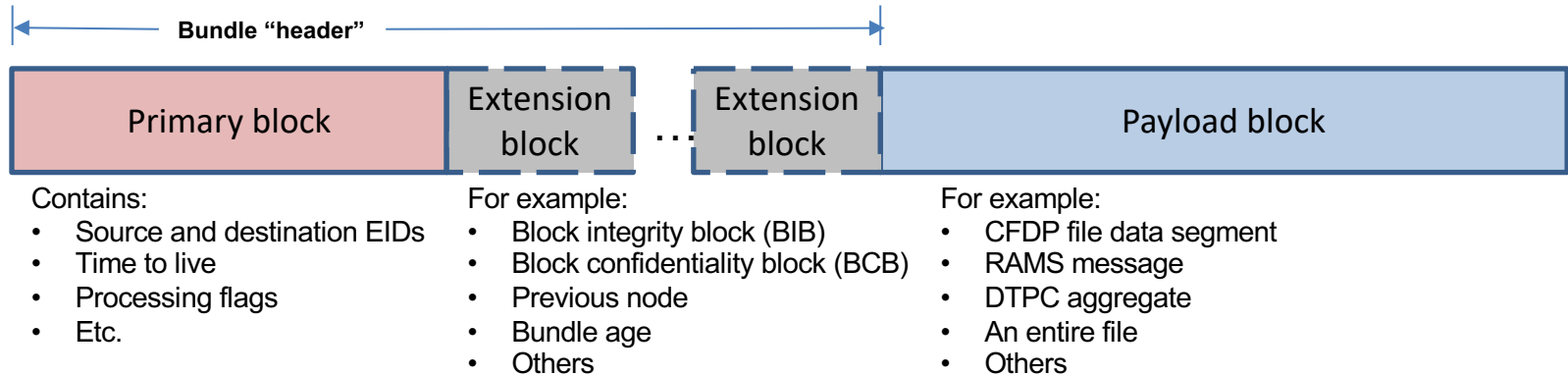


# Same Structure

- Just as in the Internet, the stacking of DTN protocols is reflected in the structure of the protocol data units that are transmitted over the network.

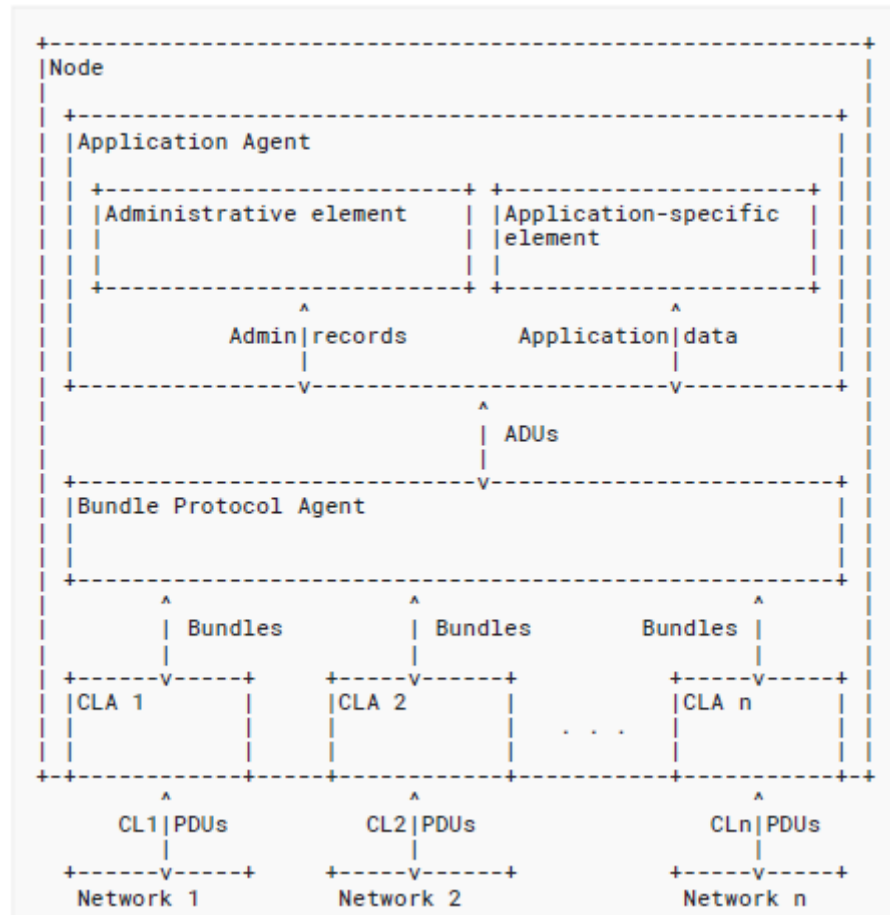


# A Bundle



- Every bundle is a sequence of blocks:
  - Primary block
  - Zero or more extension blocks
  - Payload block
- BP is designed for bandwidth efficiency where possible, expansion as necessary. Most fields are variable-length and are encoded in CBOR.

# Architecture of a BP Node



# Application Agent

- The application-specific element of the application agent comprises all of the user applications that are going to use bundle protocol for the transmission and reception of application data units (ADUs) encapsulated as the payloads of bundles. Each such application uses the BP application programming interface to:
  - Pass application data units to the bundle protocol agent (BPA) for transmission and
  - Receive application data units extracted from the bundles that the BPA has received.
- The administrative element of the application agent acts in exactly the same way as any other BP application except that the application data units it issues and receives are all BP *administrative records* – items of information that are used to monitor and control the operation of the BP node itself.

# Bundle Protocol Agent

- The BPA simply performs all of the protocol procedures mandated by the BP protocol specification, IETF RFC 9171.
- RFC 9171 does not mandate the implementation approach by which the BPA is realized, and a wide variety of approaches have been taken to date. For example, the BPA may take the form of:
  - A single monolithic operating system process.
  - A set of cooperating system processes that are responsible for the various procedures.
  - A library of functions that are invoked by applications and convergence-layer adapters as application data units and bundles are delivered via application programming interfaces.

# Convergence Layer

- The BPA doesn't physically radiate bundles; it invokes the services of other underlying protocols that are appropriate for the communication regime in which the bundles must be transmitted.
- Those underlying protocols form a subordinate stack of their own, as they perform communication functions of their own that ultimately rely on the electromagnetic radiation performed at the “physical” (lowest) stack layer.
- The topmost protocol in this subordinate stack – the protocol that directly interoperates with the BPA – is termed a *convergence-layer protocol*.
- In order to invoke the service of the convergence-layer protocol (directly) and the other protocols in that subordinate stack (indirectly), additional software must be provided that accesses BPA functionality in a standard way and also utilizes the application programming interface of the convergence-layer protocol. This software is termed a *convergence-layer adapter*.

# Endpoint IDs (1 of 2)

- The sources and destinations of bundles are endpoints, identified by endpoint IDs that are functionally similar to IP addresses.
- But BP endpoint IDs are names, not addresses; they have no topological significance and contain no hints about the route the bundle should take.
- Instead they are Uniform Record Identifiers (URIs), i.e., ASCII text strings of the general form:
  - `scheme_name:scheme_specific_part`
- For example: `dtm://topquark.caltech.edu/mail` conforms to the “dtm” endpoint ID scheme.

# Endpoint IDs (2 of 2)

- To reduce transmission overhead, endpoints may instead be named in conformance to the alternative “ipn” scheme:

ipn:allocator\_number.node\_number.service\_number

- Endpoint IDs formed in the “ipn” scheme can be abbreviated to an array of unsigned binary integers by CBOR.
- ipn-scheme BP endpoint IDs (EIDs) look a little bit like Internet addresses (for example, “ipn:11.3982.64”), but they are still just names (written mostly in numbers rather than letters).
- Node numbers identify the flight or ground data system computers on which network software executes, and service numbers are roughly analogous to TCP and UDP port numbers, but node numbers are not topologically aggregated to indicate routes.