International Conference on Modeling and Simulation
Coimbatore, 27-29 August 2007

DESIGN AND MODELING OF DTN BUNDLING PROTOCOL

Deepak N A[a]*, Unnikrishnan E[b], Dr. A.G Ananth[a], P.Lakshminarasimhan[b]

[a] *Department of Telecommunication & PG Studies, RV College of Engineering, Bangalore – 560059, India;* e-mail: deepnadig@gmail.com

[b] *DCS, Telecommand Division, Digital Systems Group, ISRO Satellite Centre, Bangalore – 560017, India*

Abstract. Communication protocols today being extremely complex contain exhaustive definition and specification sets which need to be incorporated in the software implementations. As the information is difficult to comprehend, a visual model of the specifications provide a greater amount of clarity, thus leading to more efficient and bug-free implementations. A model driven approach to communication protocol design reduces significant development time and cost through automatic construction of the software from the visual design model. This paper presents a model driven architecture paradigm for design of Delay-Tolerant Network (DTN) Bundling Protocol. The design model formalism provides a well-defined structure which supports refactoring, event based modeling. The visual model also provides formalism for validation and verification of the design model. Models are designed using Object Management Group's (OMG) model driven architecture core, Unified Modeling Language (UML). Furthermore, the protocol can be modeled using any language which supports state-transition semantics. The design can be modified to incorporate any of the four architectures suggested in the Bundle Protocol Specifications [6]. Since UML is based on the object oriented methodology, translation of the state machines to the language of implementation is easily automated. The protocol interactions are modeled through the use of sequence and state diagrams This paper presents a methodology for designing and modeling the Delay Tolerant Network Bundling protocol. The methodology provides a common framework for use of the design with various suggested system architectural implementations. The automation in the implementation language also provides flexibility regarding the choice of the language itself.

Key words: Protocol Modeling; Bundle Protocol; Delay-Tolerant Networks;

1. INTRODUCTION

The Model Driven Architecture (MDA) approach to communication protocol design reduces significant development time and cost through automatic construction of software implementations from the visual design model. To realize this, we need a graphical programming environment which possesses the ability to program directly in the modeling language. Unified Modeling Language (UML) is used here to incorporate a MDA based approach. The architectural modeling capability of UML is based on mature languages such as Specification and Description Language (SDL) which is a formal language providing useful abstractions for protocol engineering. UML can also be used to incorporate higher level abstractions which are implementation platform/target language independent. Also, UML can

be used in a non-standard approach by merging UML specifications with a suitable formal language, for example, ITU-T Z.109 Recommendation [4], merges UML and SDL.

2. BUNDLE PROTOCOL DESIGN FRAMEWORK

The Bundle protocol design framework is a top-level structure representing the communication services, interfaces, entities, interactions, bundle formats, addressing schemas as defined in the Bundle protocol specification [6]. The framework provides a high level abstraction model consisting of different specifications which serves as a vehicle for implementations and also for protocol testing and analysis. The Figure.1 shows the protocol design framework used.

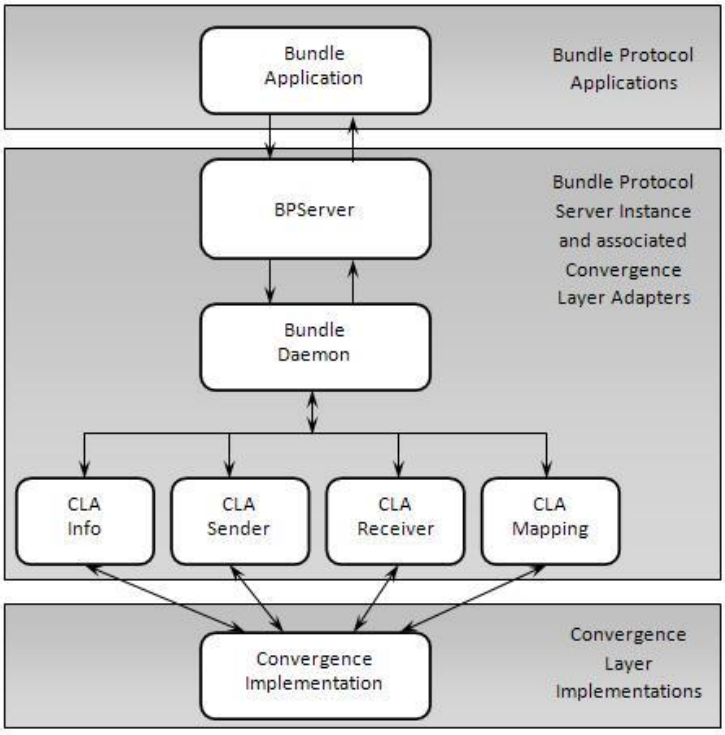

Fig. 1: Bundle Protocol Design Framework.

The Bundle Application entity is the abstraction layer that implements the Bundle Protocol Service API interfaces. It has applications like Bundle Sender, Bundle Receiver, Ping Applications, Batch Sender and Multiple Bundle Senders with configurable bundle size, and priority levels. The Bundle Protocol Server provides functionalities of the application specific agent of the application agent. It basically provides an interface for different protocol services in the form of defined operations which can be invoked by various Bundle Applications. The BPServer runs as a thread which is instantiated on every run.

The Bundle Daemon acts to provide all the functionalities of the Bundle Protocol Agent. It is associated with the creation of Bundles and Bundle Payloads, Status Report generation, Fragment management, Consumption of received Bundles and also the Forwarding of Bundles.The Bundle Daemon is associated with one or many Convergence Layer Adapters (CLAs) which provide interface to specific Convergence Layers. The CLAs map the data to the specific Convergence Layer formats thereby routing the data through the lower layer (transport) protocol structures. The CLA interacts directly with a Convergence Layer

Implementation through various entities like the CLA Sender and Receiver entities for routing the Bundles through them.

The following section describes the top level package design used in the modelling of DTN Bundling Protocol. The Figure 2 describes the design framework. It consists of the following packages:

- Bundle Protocol Server Package
- Bundle Protocol Agent Package
- Application Agent Package
- Convergence Package
- Contacts Package
- Registration Package
- Routing Package
- Naming Package
- Events Package
- Convergence Layer Package
- Utilities Package

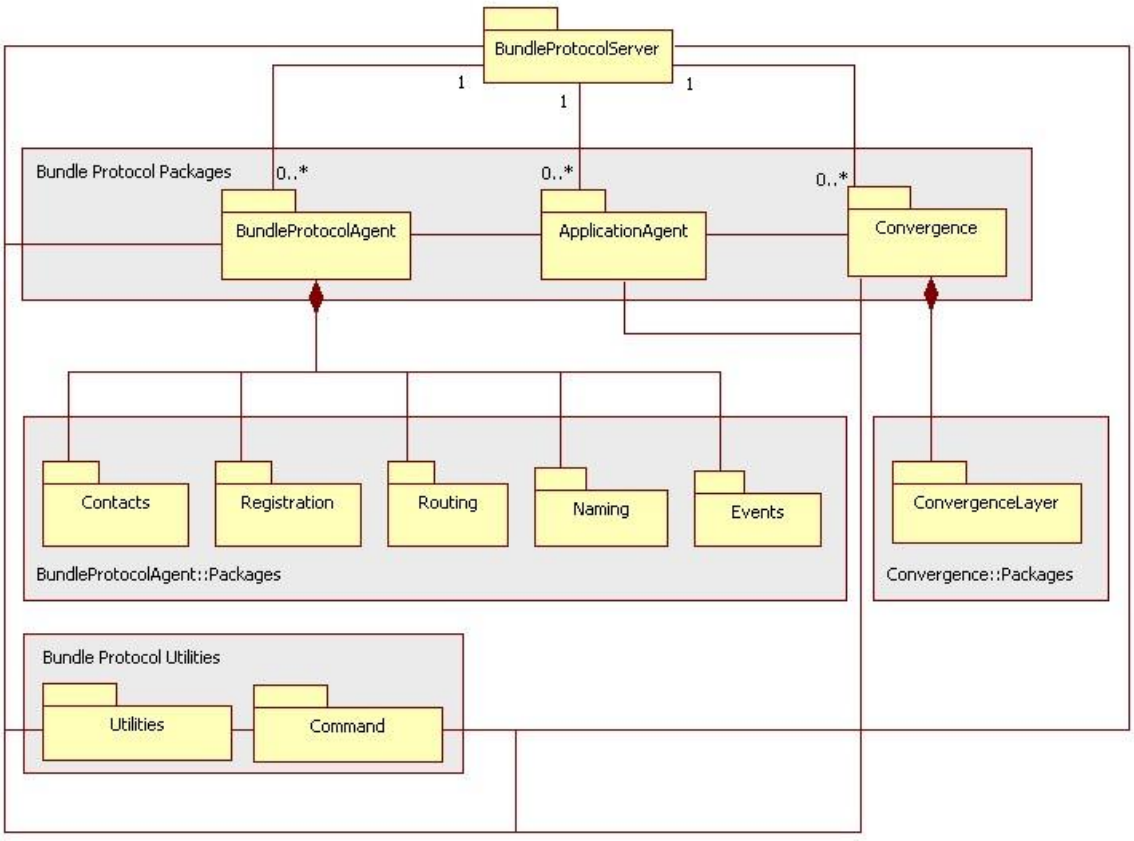

Fig. 2: Top Level Package Design Framework.

This Top-Level Design consists of 12 main packages, each of which forms a composite of multiple classes, operations, attributes and relationships. The Bundle Protocol Design consists of over 130 classes, 350+ operations and 500+ attributes.

3. BUNDLE PROTOCOL INTERACTION INSTANCES

Protocol Interaction Instances provide a case specific presentation of the bundle protocol usage. A sequence diagram illustrates the sequence of events that are to occur to use a particular protocol service. A sequence diagram illustrating bundling transmission is shown below:

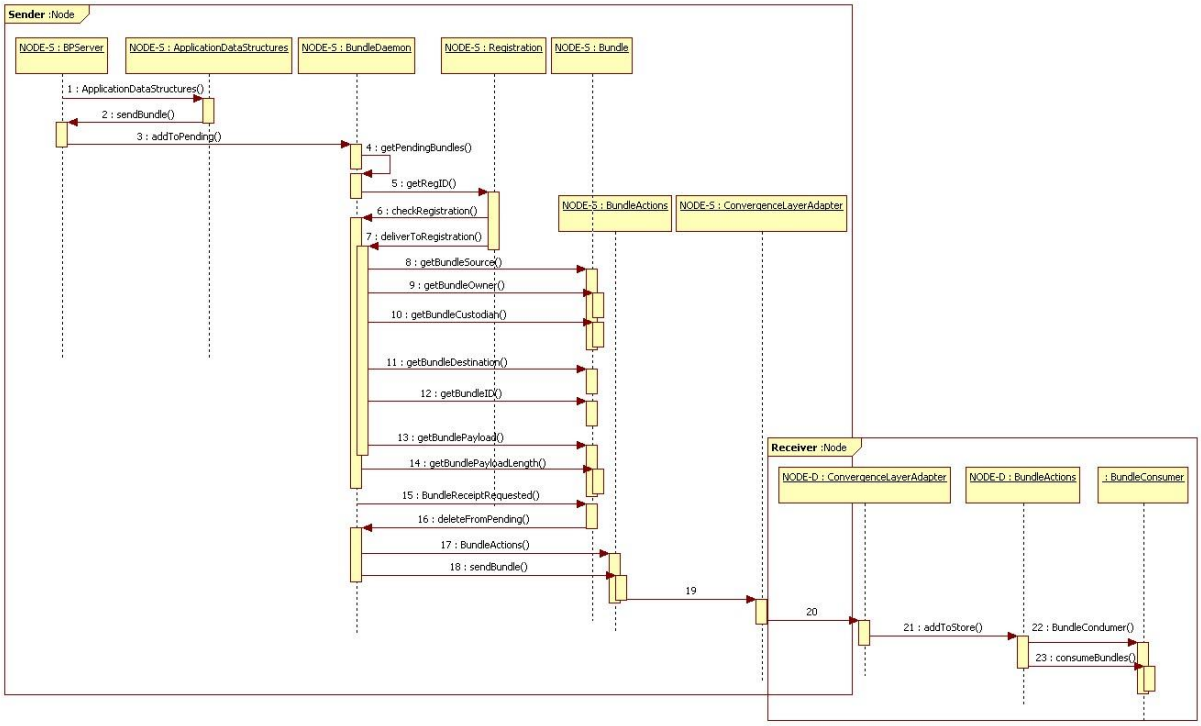

Fig. 3: Bundle Protocol Transmission Sequence.

The design models discuss the various relationships that exist between different protocol entities. The protocol interaction instances are also shown and there are designed to follow the bundle protocol specifications with a high degree of accuracy. The design although fairly comprehensive, does not include the modeling of the custody transfer mechanisms and operations. The custody transfer entities should be modeled as a separate package and an extra interaction instance has to be augmented to each protocol interaction so accommodate custody transfer.

3. BUNDLE PROTOCOL DESIGN VERIFICATION/ VALIDATION

The design and verification of communication protocols being extremely complex make issues regarding correctness of protocol structure, robustness and performance of the communication protocols important and critical. The protocol verification techniques deal mainly with correctness of the design structure, liveness and safety properties by comparison of the design with the protocol specification.

3.1 Design Abstraction

The communication protocols designed exhibit complex behaviors. Since they have to interoperate with the application as also the low level communication medium, they are to be designed to be robust to failures of the associated entities. Since the failure of the protocol

dependent entities imply the induction of an asynchronous behavior into the protocol operation, designing protocols with the ability to cope with the associated failures is extremely important. The level of design abstraction used here, makes the protocol structuring less complex.

3.2 Bundle Protocol Design Validation

Implemented protocols suffer design failures due to serious design errors and ambiguous protocol requirement specifications. Protocol validation techniques must be used to ensure that the protocol interactions are according to the requirement specifications and satisfy properties and conditions as deemed necessary by the protocol specifications. The protocol design validation techniques can be used to check for design errors like deadlock errors, livelock errors and unspecified receptions. Some of the design errors include:

(i). Non-executable interactions
(ii). Unboundedness
(iii). State ambiguities
(iv). Lack of adaptation

3.2 Bundle Protocol Reachability Analysis

Reachability analysis is based on global state generation where each protocol process is represented as a state machine. The technique deals with the generation of all possible state transitions between all possible processes. The resulting sequence can be generated and represented as a reachability tree. This method can be used to diagnose errors like deadlocks and unspecified receptions. The state diagram for reachability analysis is shown below.

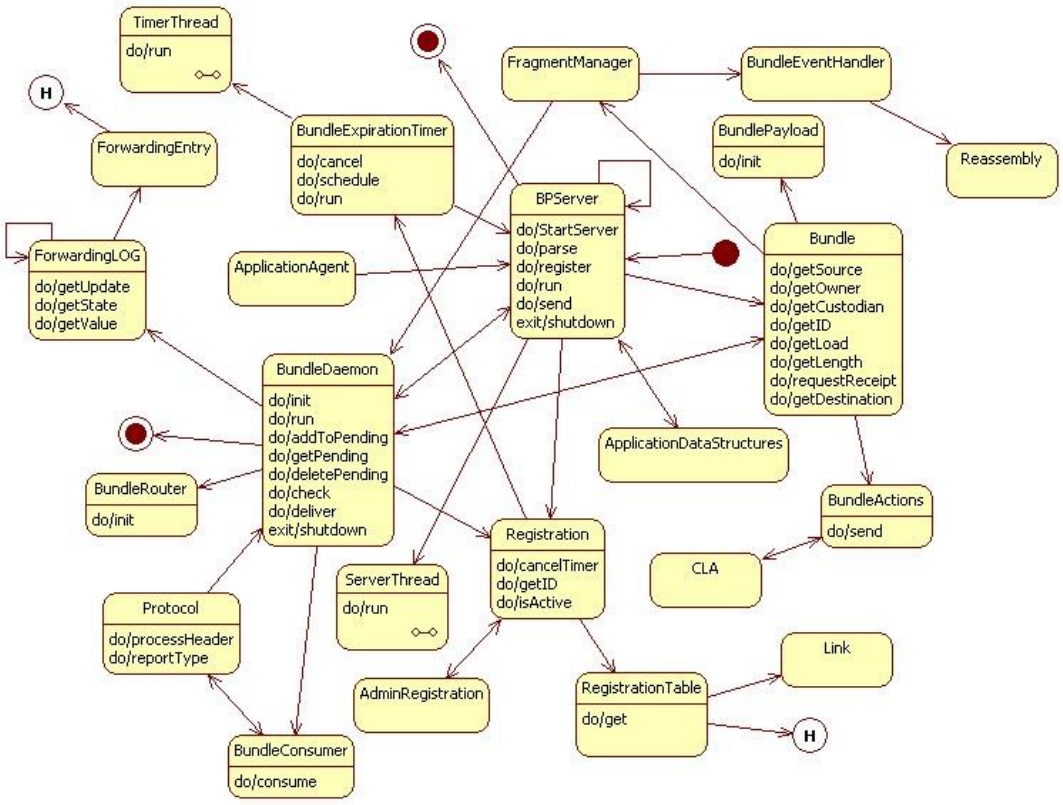

Fig. 4: Bundle Protocol State Diagram for Reachability Analysis.

CONCLUSIONS

This paper presents a methodology for the design and modeling of Delay-Tolerant Networking Bundle Protocol. A Model Driven Approach to protocol design is presented with a development framework for use with the DTN Bundle Protocol. The protocol interaction instances are also modelled through the use of sequence diagrams. The methodology provides a common framework for use of the design with various system architectural implementations incorporating DTN Bundle Protocol.

REFERENCES

[1]. Delay-Tolerant Networking Research Group (DTNRG), http://dtnrg.org/

[2]. Forrest Warthman, "Delay-tolerant networks (DTNs): A tutorial", v1.1, Mar 2003.

[3]. Henrik Eriksson & Patrik Jonsson, "Implementation and Analysis of the bundling Protocol for Delay-Tolerant Network Architectures", *Master's Thesis*, Lulea University of Technology.

[4]. ITU-T Recommendation Z.109, " SDL Combined with UML," 1999.

[5]. K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," *Proc. ACM SIGCOMM Conf.*, ACM Press, 2003, pp. 145-158.

[6]. K. Scott & S. Burleigh, "Bundle Protocol Specification," *Internet Draft*, Work in Progress.

[7]. M. Demmer *et al*., Implementing Delay-Tolerant Networking, tech. report IRBTR-04-020, Intel Research Berkeley, 2004.

[8]. RFC4838, V. Cerf *et al*., "Delay-Tolerant Networking Architecture," April 2007.

[9]. S. Burleigh *et al*., "Delay-Tolerant Networking: An Approach to Interplanetary Internet" *IEEE Commun. Mag.*, vol. 41. no. 6. June 2003, pp. 128-36.

[10]. S. Symington *et al*., "Bundle Security Protocol Specification," Internet Draft, Work in Progress.

[11]. V. Cerf *et al*., "Interplanetary Internet (IPN): Architectural Definition." Internet draft, draft-irtf-ipn- arch-00.txt. May 2001