Distributed Interplanetary Delay/Disruption Tolerant Network (DTN) Monitor and Control System

Shin-Ywan Wang
Jet Propulsion Laboratory
4800 Oak Grove Dr.
Pasadena, CA 91011
818-354-4832
shin-ywan.wang@jpl.nasa.gov

***Abstract*—** The main purpose of Distributed interplanetary Delay Tolerant Network Monitor and Control System as a DTN system network management implementation in JPL is defined to provide methods and tools that can monitor the DTN operation status, detect and resolve DTN operation failures in some automated style while either space network or some heterogeneous network is infused with DTN capability. In this paper, "DTN Monitor and Control system in Deep Space Network (DSN)" exemplifies a case how DTN Monitor and Control system can be adapted into a space network as it is DTN enabled. Also a "DTN Status Diagnose and Treatment Creator" tool is devised to provide a platform for network operator to compose failure treatment methods according to the received DTN BSR (bundle status report), DTN implementation specific log message and the network specifically produced monitor and control data. This innovative interplanetary DTN Monitor and Control system was created and installed in the mission administrative node as a multi-tiered internet-based web application to support DTN system by monitoring the data delivery and analyzing the delivery statistics from ION (Interplanetary Overlay Network), JPL originally implemented DTN or other DTN implementation. This monitor and control system includes modules related to the DTN bundles status report receiving, health monitoring, DTN operation failure diagnosing, failure treatment composing and failure treatment to GUI conversion. Moreover, a GUI system is designed to provide visual interfaces to display the operation status and DTN topology lively, query the DTN status stored in the database, and guide the designated user to fix the DTN failure.

TABLE OF CONTENTS

1. OVERVIEW OF DELAY TOLERANT NETWORK PROTOCOL (DTN) AND INTERPLANETARY OVERLAY NETWORK (ION)

A delay-tolerant network[1][2] is a network architecture designed to operate effectively as an overlay on top of regional networks, or as an interplanetary internet. The interaction among regional network are characterized as long delay, intermittent connectivity and asymmetric data rate and high error rate due to the long distance and extreme environments as in deep space communications. DTN provides the solution to overcome the stated problems by using store-and-forward service to move the data from the source to the destination. Bundle [3] is the data unit to be transferred in the DTN system and the devices which move the bundles are called DTN nodes. Bundle status report is a bundle which, instead of containing the payload data, contains DTN operation administrative information reporting the DTN operation status.

Interplanetary Overlay Network (ION) [4] is JPL originally developed DTN. In order to raise the technical readiness of ION software, JPL Deep Impact Network Experiment (DINET) had utilized Deep Impact spacecraft to perform the first DTN experiment in deep space to validate the usefulness of DTN protocol in November, 2008 [5][6]. In the DINET topology, the Deep Impact spacecraft acted as a router to pass payload data from one simulated space region (Mars and Phobos) to another (Earth). For the DINET experiment, a DTN Monitor and Control system was created and installed in the experiment operation center (EOC) administrative node located in JPL's Protocol Technology Laboratory(PTL) for live status monitoring.

2. DTN MONITOR AND CONTROL SYSTEM

General Speaking, the purpose of network management means activities, methods, procedures and tools that pertain to the operation, administration, maintenance, and provision of networked system. The main purpose of DTN monitor and control system as DTN network management implementation is defined to provide methods and tools that can monitor status and resolve the failures that occurred in DTN operation in some automated style.

The entire DTN Monitor and Control system is designed to function in a distributed working style. The central part of the monitor and control system collects DTN operational

978-1-4577-0557-1/12/$26.00 ©2012 IEEE

status from remote DTN nodes, detects the operation failures and helps seek the failure treatment. The failure treatment may be provided by some expert application which may reside in somewhere away from the center of the monitor and control system. The overall structure of DTN monitor and control system is depicted in Figure 1. The core part of the Monitor and Control System, as shown in the figure, is installed in mission's administrative node which receives the message and bundle status report issued from all DTN nodes. The major modules in the DTN Monitor and Control system are message receiver which receives and parses status messages, health and status monitor unit which detects the failures from the received messages and dispatches classified error messages to the proper failure diagnosis system which resides locally or

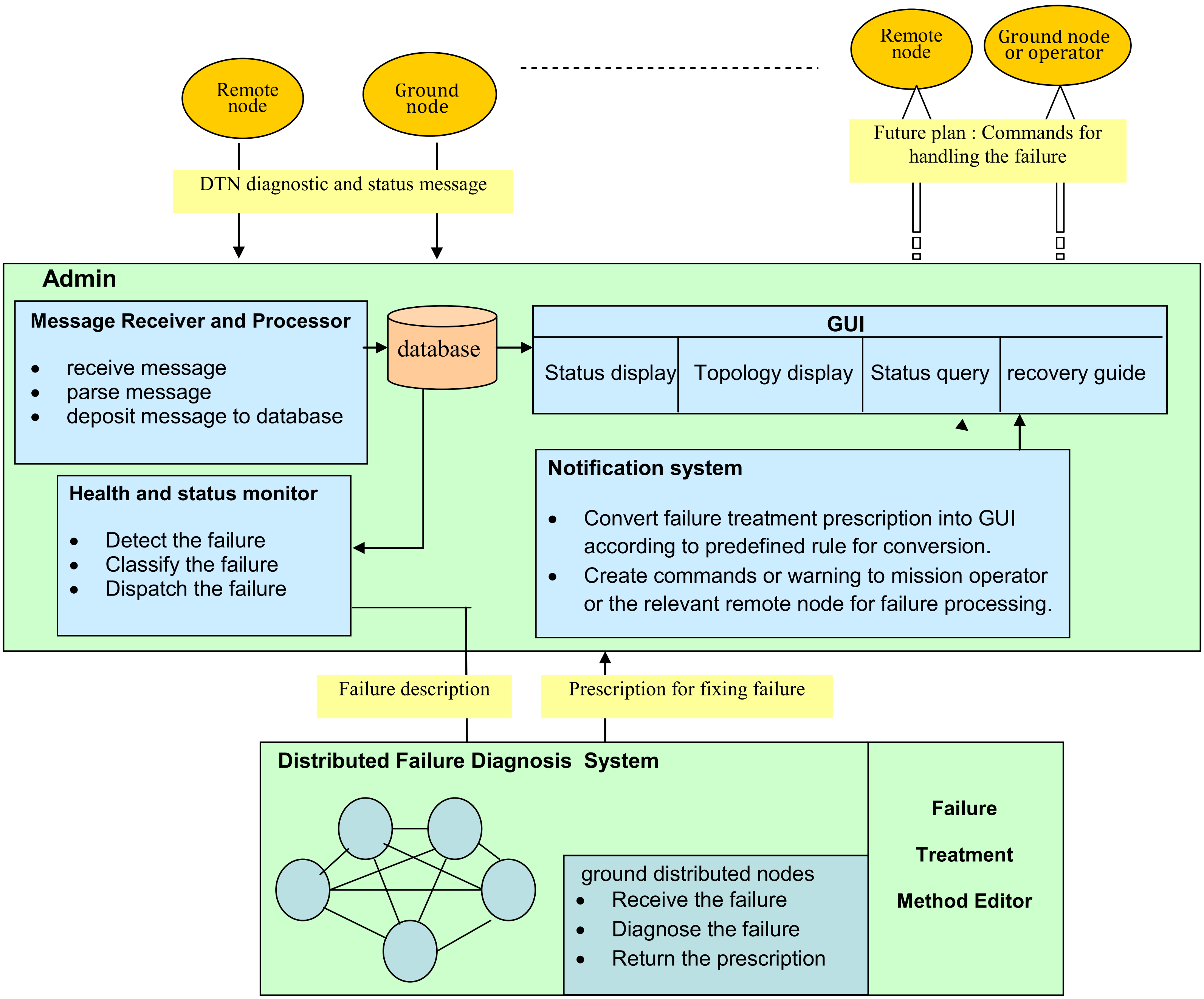


Figure 1: structure of DTN Monitor and Control System

remotely or the Failure Treatment Method Editor, a tool for composing the treatment for DTN operation failure, and the notification module which converts the failure treatment composition sent from the diagnostic system or Failure Treatment Method Editor into Web GUI display. The system user interactively communicates with the Monitor and Control system via the Web GUI pages to track the operation status and the network topology lively, send the status query to the database, and resolve the network failure according to the solving instructions given by the GUI page. The main modules are described in more details as follows.

- ***Message Receiver and Processor***.

The DTN Monitor and Control System intends to work with the DTN system which uses either JPL original DTN implementation, ION, or other DTN implementation, such as DTN2[9]. Other DTN implementation, like DTN2, mainly sends the bundle status report to inform the bundle transmission errors. As for ION, it has the implementation to convert the content of the bundle status report and custody signal to some readable text message which can be further redirected to the administrative node. Besides the bundle administrative messages, ION software itself also produces some other types of text messages, such as diagnostic and warning messages which can be also redirected to the administrative node to notify some failure or problem to be.

The message receiver and processor in the DTN Monitor and Control system receives the status messages, parsers the messages, and deposits the message to the database. Mainly, the message receiver and processor receives two types of messages from network simulation, bundle status reports via DTN from Non-JPL DTN implementation, and log message (converted from bundle status report, BSR) via TCP/IP from JPL originally impleneted ION. As shown in Figure 2, "Bundle receiver" is used to receive the bundle status reports from other DTN implementation, extract and time-stamp the message content, and further the message items are parsed by "Bundle Status Report Parser". On the other side, "Message receiver" is used to receive log message from ION via TCP socket, time-stamp the message and classify the message into different categories (warning message, informational message, status message from BSR, diagnostic message, communication statistics message and other message). Further the message content is parsed by "log message parser" which extracts message items for each defined message type. The bundle content extracted from the bundle receiver can also be converted into ION formatted text message first and sent to the "Message Receiver" for further processing. All parsed bundle status reports or log messages are deposited into the database. A "database management library" which manages the database creation and deletion plus data content deposit and removal is implemented to bridge the database system and message parser/bundle parser.

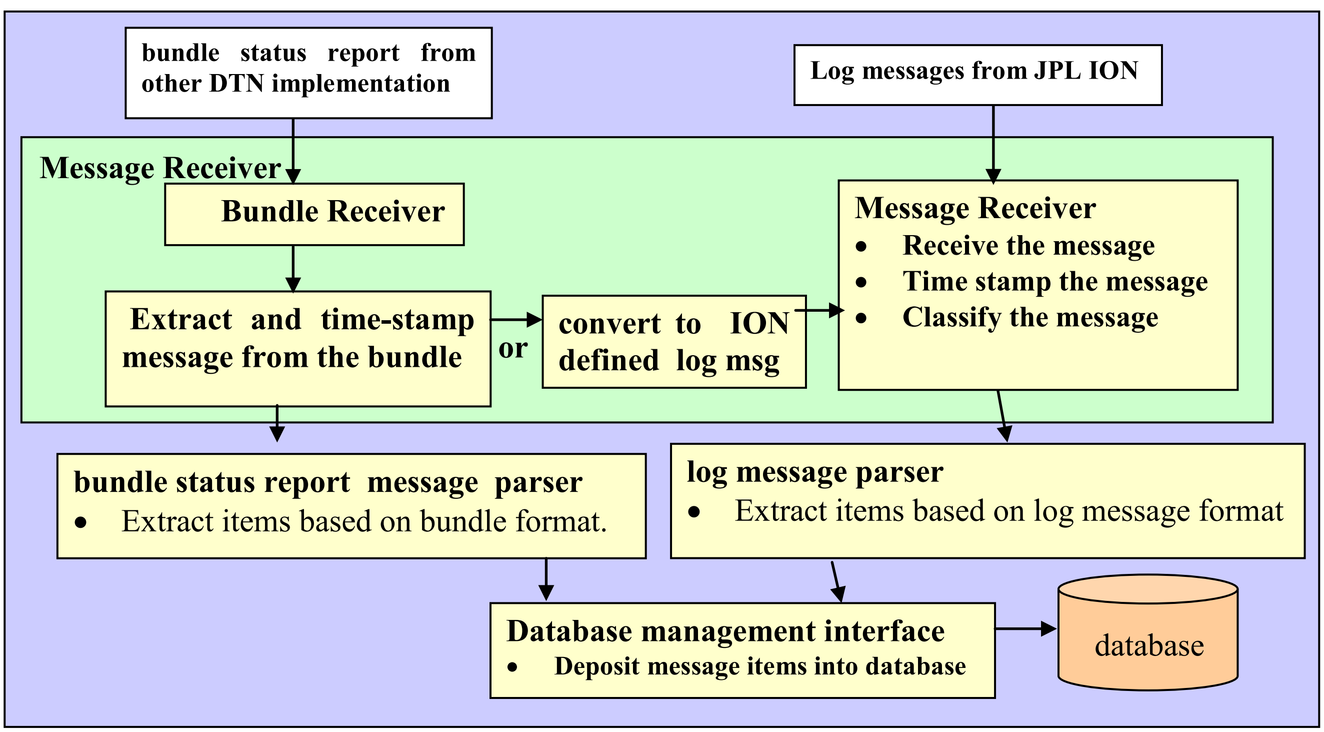


Figure 2 Message Receiver and Processor

▪ ***Health and Status Monitor***

The health and status monitor is started to receive the notification from the Message Receiver and Processor unit which informs the new deposit of the log message into the database. As the Health and Status Monitor unit is signaled, it retrieves the new message from the database and detects its health. The message which is detected to be problematic will be further classified and dispatched to some specific failure diagnostic application or the Failure Treatment Method Editor for further examination and treatment. The problematic messages which the Health and Status Monitor sends to the diagnostic system include one key message which contains the major 'symptom' of the failure and some relevant messages which reveal more 'symptoms' probably related to the failure. The communication between the "Health and Status Monitor" unit and the "Failure Diagnosis" unit can be implemented by using TCP/IP or some message subscribe/publish system, such as Asynchronous Message System (AMS) protocol implementation [7]. Figure 3 is a example of failure description expressed in XML format in which key symptom and related symptoms of the failure are included.

▪ ***Failure Diagnosis System***

Failure Diagnosis system could be either procedures bundled with the Health and Status Monitor unit, or some application located in local or remote machine to diagnose the network failure. If the diagnostic unit is a procedure, then some structure object which contains the key message and relevant messages is sent from the Health and Status Monitor to the diagnostic unit. If the diagnostic unit is a standalone executable piece, the key message and the relevant messages which express the failure is sent via TCP/IP or some subscribe/publish system.

Each Failure Diagnosis system diagnoses the failure and provides the solution to the failure based on the key message, the relevant messages and network status data. It may also communicate with some other application to seek more specialized solution.

The Failure Diagnosis system results in the failure treatment method which contains a linked list of solution steps as shown in Figure 4. The order of the solution steps stands for the sequence to solve the failure. Three types of solution steps are defined to describe treatment methods They are topology step, notification step, and suggestion step. Figure 5 shows an example of treatment description of XML format in terms of the three solution steps in which the required information for each solution step type is included. Topology step reports the connectivity and availability of the network according to the failure messages. In the XML description, the network connectivity is expressed by a list of nodes and their neighbors. The network operation administrator can further reconfigure the topology at the front end according to the network availability reported by the topology step. Notification steps contains the text expression which could be a warning message, informational message or some instruction used to guide the user of Monitor and Control system to fix the network error. The suggestion step proposes resetting some network operation or configuration parameters in order to resume the network operation. Each parameter listed in the suggestion step will be presented to the user and let the user reset the value of that parameter. In the XML description, parameters suggested to be reset are categorized into different group and the valid value range or possible values of each parameter might be also provided. As shown in Figure 5, each solution step may have some application name associated with. Those application names hint applications to be executed to fulfill solution steps at the server site of the Monitor and Control system.

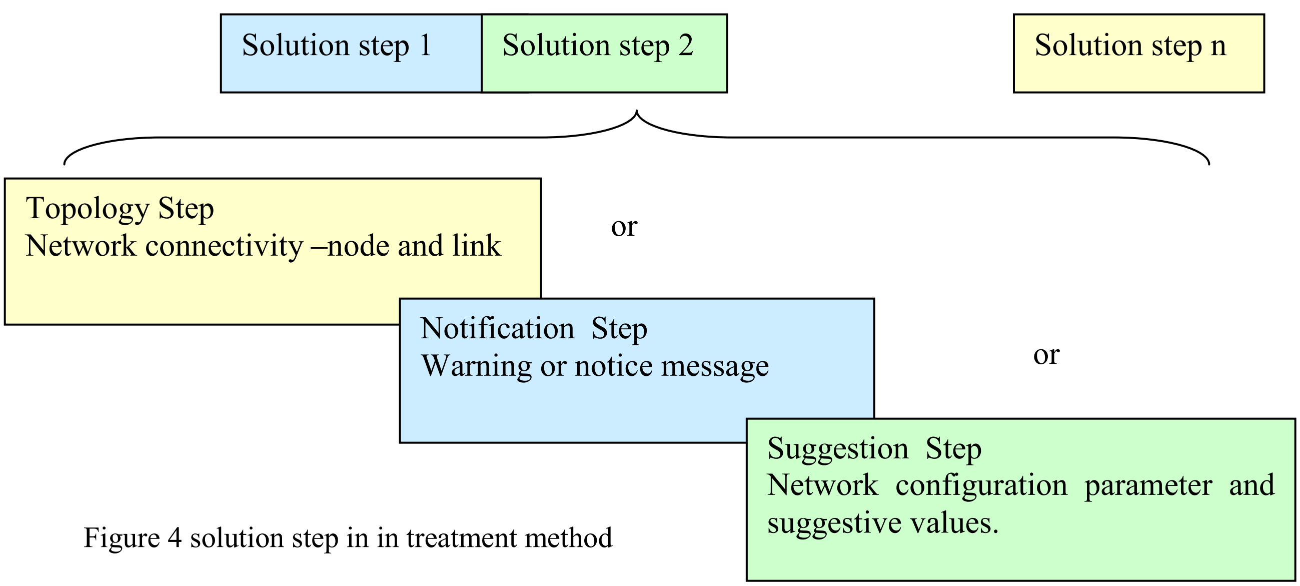


Figure 4 solution step in in treatment method

```
<networkFailure>
<totalMessageNumber>num</totalMessageNumber>
<databaseName>dname</databaseName>
<keyMessage>
<databaseTableName>tname</databaseTableName>
<IDInTable> id </IDInTable>
</keyMessage>
<relevantMessage>
<databaseTableName>tname</databaseTableName>
<IDIntable> id </IDIntable>
</relevantMessage>
  :
</networkFailure>
```

Figure 3 XML format of failure description

▪ *Notification System and GUI*

Notification System unit receives the failure treatment methods from the Failure Diagnosis application in either structure object format or XML description format via TCP/IP or some publish/subscribe system. As shown in Figure 7, the Notification System resides at the server site of the Monitor and Control system, interpreting solution steps of the treatment method received either locally or remotely, converting the treatment method into web GUI pages which interactively guide the user to fix DTN operation failures. The conversion from each type of the solution step into the GUI web is respectively illustrated in Figure 6 (a) (b) and (c). Figure 7 also shows the interaction between the Notification System and the Web GUI user. From the GUI, the user may see some notification, warning or instruction message, reconfigure network topology and reset some parameter values while the guide proceeds. The reconfigured topology and reset parameter values will be sent back to the notification system for further processing in order to fix the failure.

▪ *Diagnosis Treatment Method Creator*

In the DTN Monitor and Control system, how to effectively diagnose and treat various DTN failures is the main challenge. It relies on the quantity and quality of the failure diagnosis units connected to the Monitor and Control system. Diagnosis Treatment Method Creator is a tool which provides an environment for DTN expert to compose the treatment method according to the failure messages and DTN network configuration information in case no existing failure diagnosis unit can be found to fix the occurred network operation failure. This tool is a Java GUI application, It mainly provides editing interfaces for the user to express the sequence of the solution steps and fill in solution content for each type of solution step, and eventually converts the composition into the treatment method description to be sent back to the Notification System.

```
<treatmentMethod>
  <solutionStep type = topology >
  <topologyStep>                           // topology  type
     <oneNode>
       <name>nodeName</name>
       <neighborsNode> name </neighborsNode>
           :                          // more <neighborsNode>
     </oneNode>
     <oneNode>
       <name> nodeName </name>
       <neighborNode> name </neighborsNode>
         :
     </oneNode>
    <application> applicationName <application>
  </topologyStep>
</solutionStep>
<solutionStep type = notification>
  <notificationStep>                        //notification type
    <message type = warning|notification> text </message>
    <application> applicationName </application>
  </notificationStep>
</solutionStep>
<solutionStep type=suggestion>
  <suggestionStep>                         //  suggestion type
    <instruction> text </instruction>
    <application> applicationName </application>
    <oneEntrySet>                  // categorized parameter set
       <oneEntry>                         // one parameter
         <label> text </label>
         <possibleEntryValues>
           <value> text | num </entry>
                       :
         </possibleEntryValues>
       </oneEntry>
       <oneEntry>                         // more parameter
            :
       </oneEntry>
           :
    </oneEntrySet>
    <oneEntrySet>
           :
    </oneEntrySet>
  <suggestionStep>
</solutionStep>
     :                                     // more solutionStep
</treatMethod>
```

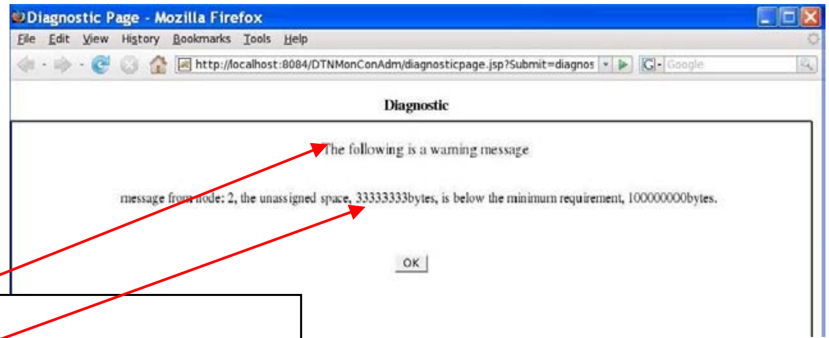
Figure 5 XML description of treatment method

Figure 6

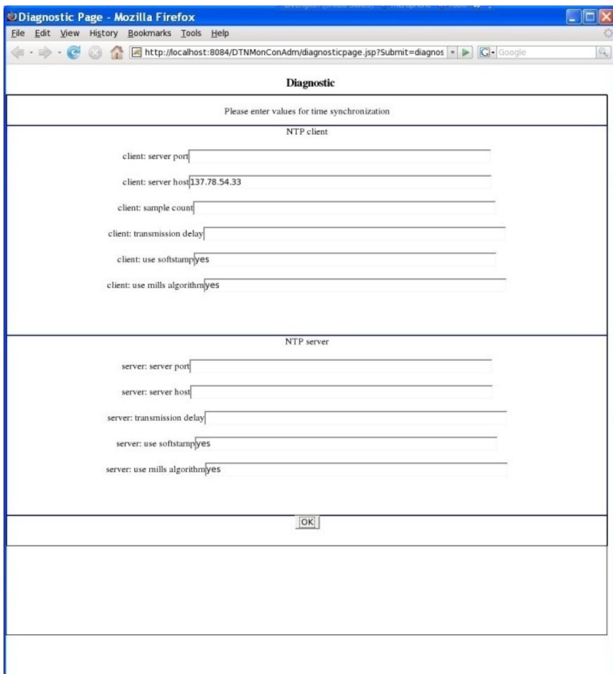
(a) from notification step to GUI

(b) from suggestion step to GUI

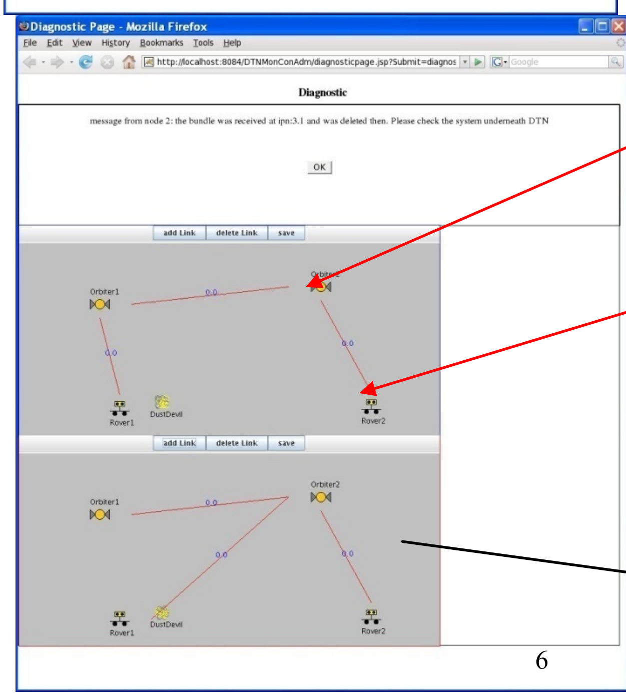
(c) from topology step to GUI




```
<notificationStep>
   <message type = warning|notification> text </message>
   <application> applicationName </application>
 </notificationStep>
```




```
<suggestionStep>
   <instruction> text </instruction>
   <application> applicationName    </application>
   <oneEntrySet>
      <oneEntry>
         <label> text </label>
         <possibleEntryValues>
            <value> text | num </entry>
                        :
          </possibleEntryValues>
      </oneEntry>
               :
   </oneEntrySet>
   <oneEntrySet>
             :
   </oneEntrySet>
 <suggestionStep>
```




```
<topologyStep>
   <oneNode>
      <name>nodeName</name>
      <neighborsNode> name </neighborsNode>
          :
    </oneNode>
    <oneNode>
      <name> nodeName </name>
      <neighborNode> name </neighborsNode>
       :
    </oneNode>
   <application> applicationName <application>
 </topologyStep>
```

User composing area

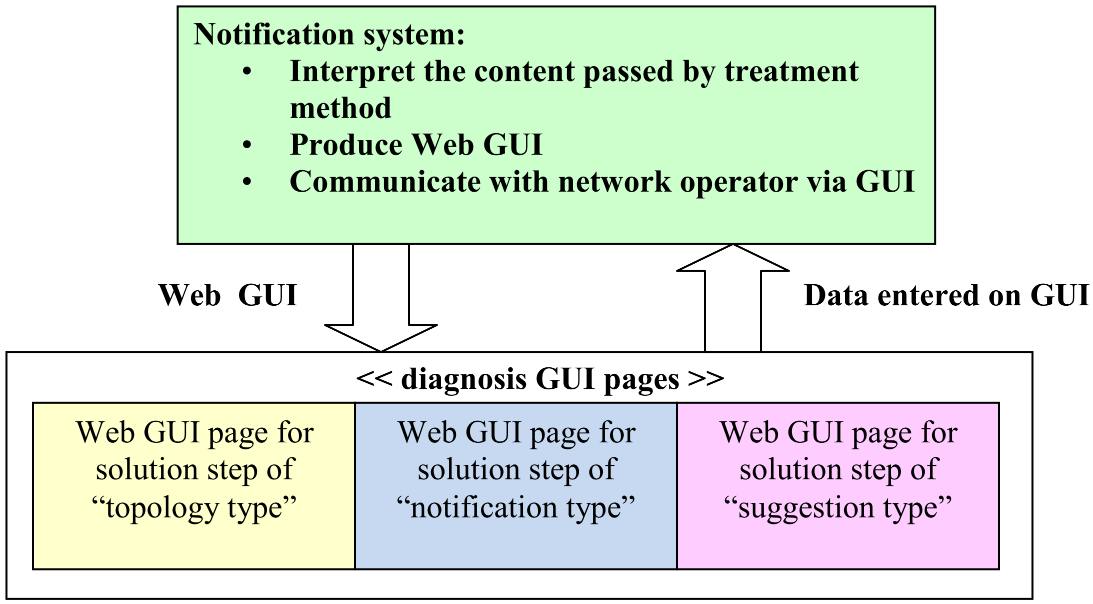


Figure 7 Notification system and GUI system

3. DTN Monitor and Control System in Deep Space Network (DSN) System

DTN is an approach to heterogeneous network that may lack continuous network connectivity. Such network could be planned space network or terrestrial network in mobile or wireless environments. For the example of space network, future missions will make more use of multiple spacecraft and relays. DTN provides the store-and-forward features to resolve the data delay and data loss problem due to the extreme space environment. With DTN, the data sent from spacecraft can be automatically routed through other spacecraft, relay and DSN antenna to the ground data center. Currently some investigation and research on infusing DTN capability into DSN system is ongoing. The DTN Monitor and Control development intends to extend the network management capabilities by exploring some technology to interface DTN Monitor and Control system with DSN system in order to detect and diagnose the network operation failure of DSN system when DSN is infused with DTN service,

DTN Monitor and Control System in DSN system is expected to detect and fix DTN operation failure according to the received DTN bundle status report, log message, and DSN ground subsystem monitor data. The access of DSN subsystem monitor data affects the approach how the DTN Monitor and Control system interfaces with the DSN system. DTN Monitor and Control system is designed to report the operation status lively and detects the operation failure instantly. Therefore how to interface with DSN system to get DSN status data in real time format is the optimal goal while building the DTN Monitor and Control system in the DSN system. However, due to the complexity of accessing live DSN status data, how to interface DTN Monitor and Control system with DSN subsystem modules to access static, instead of dynamic, DSN status data is first investigated. This investigation also includes exploring and selecting DSN status data that can benefit to fix the DTN operation failure.

The DSN status data ranges from DSN antenna configuration to link budget performance. In order to get values of certain DSN status data effectively, some agreement was made between the DTN Monitor and Control developer and the DSN subsystem engineers regarding the selection of DSN status data and the format of the request sent from DTN Monitor and Control system to DSN web-service. With the return information, the DTN Monitor and Control system user can further detect the health of the data transmission for the link between the DSN ground system and the relevant relay or orbiter.

Figure 8 displays the main GUI of the DTN Monitor and Control system, The blue circled area in Figure 8 shows the DSN status data between MRO and DSN at some time interval extracted from the response of some request sent from DTN Monitor and Control to remote DSN web-service.

The experience obtained from interfacing DTN Monitor and Control System with DSN system is not only limited to be applied to space application. It can be also adapted into terrestrial wireless or mobile networks (such as cellular systems) with DTN service infused and the key is to seek the interfacing method and get the network status properly.

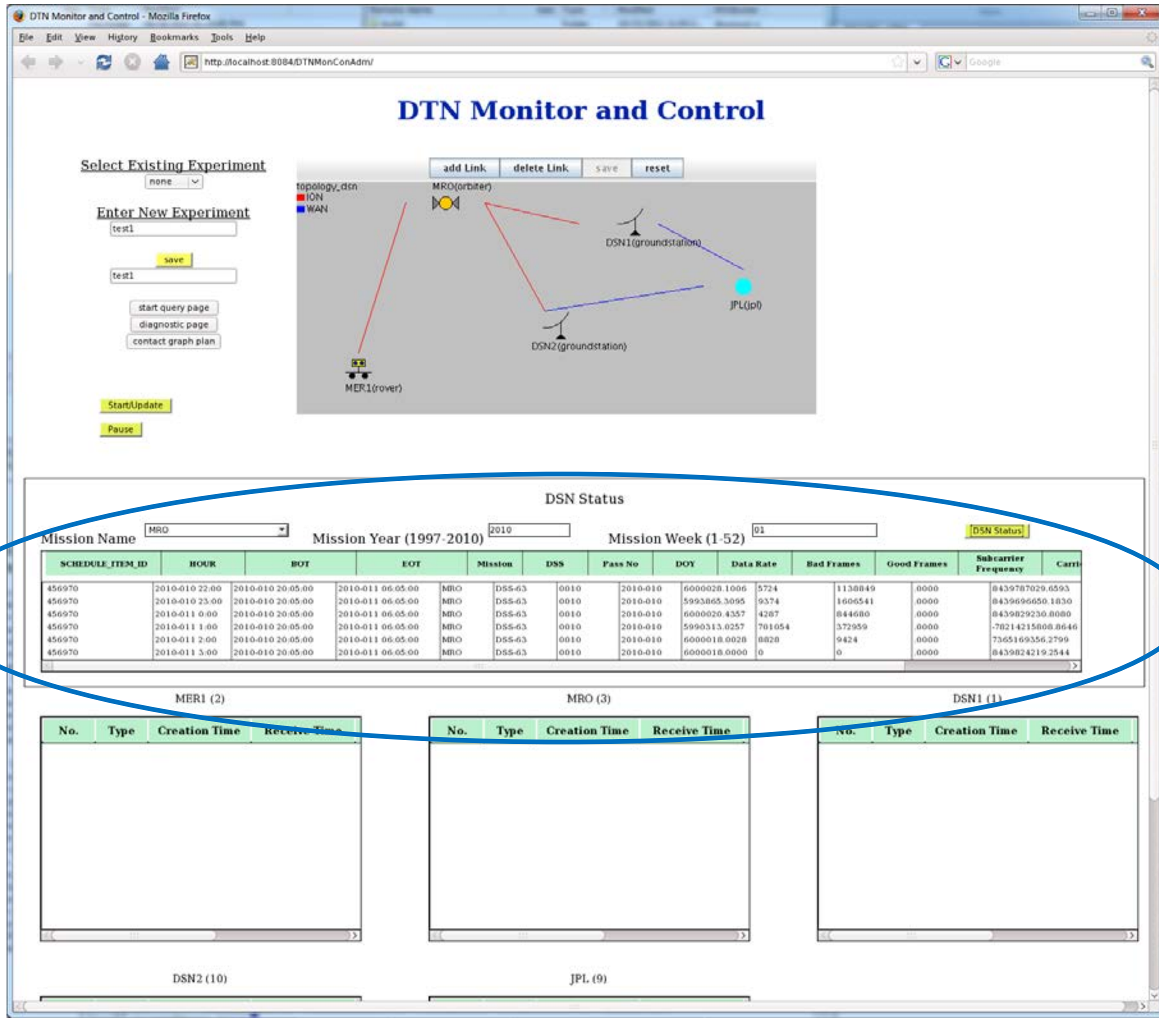


Figure 8 The main GUI page of Interplanetary Monitor and Control System which is interfaced with DSN subsystem.

4. Conclusion and Future Works

The paper has first described that the DTN based Monitor and Control System is an web based software application which intends to monitor the status of DTN system's operation, detect the health of the operation, and provide the solution to fix operational failure. With this monitor and control system, the ION software specific diagnostic messages and DTN diagnostic messages (bundle status report) issued by each network node are collected, parsed and deposited into a database in real time. All received diagnostic messages are detected and the messages which indicates any network failure or abnormality will be further dispatched to one of existing DTN failure diagnosis systems or the innovative DTN Failure Treatment Method Editor. Each DTN failure diagnosis system is mainly built to diagnose certain DTN operation failure and provide the possible treatment method to fix that failure. How to broadly and accurately diagnose and resolve DTN operation failures is challenging in the entire DTN Monitor and Control system. The DTN Failure Treatment Method Editor is some affiliated software implementation of the DTN Monitor and Control system which provides an editing environment for the network operator to compose the treatment method for the detected failure. In current DTN Monitor and Control system's implementation, three types of treatment methods are proposed as the protocol used to describe the solution to fix the DTN operation failure. As either the failure diagnose system or the failure treatment editor sends the treatment method back to the center of the DTN Monitor and Control, a notification module is built to receive the treatment method and convert it into a series of web GUIs which may notify the network user the occurred failure, guide the user to reset network configuration parameters or reconfigure the network topology. This conversion from the treatment method description to the GUI pages is based on some predefined rules.

The DTN Monitor and Control system mainly provides functions to monitor the status of DTN operation and propose solutions to the operation failure. However the collection of operation status may affect the quality of problem solving. The current development of DTN Monitor and Control system detects the failure based on the received DTN diagnostic messages and the specific status data from the network it is attached, such as DSN system. In the future, the DTN Monitor and Control system can be expanded to be interfaced with various mission operation center or design center to obtain information related to DTN operational status and DTN node/link configuration for either the space or terrestrial cases. For example, interfacing with JPL's Mars Relay Operations Service (MaROS) [8] can obtain viewing period between the orbiter and the rover, and that will be useful for scheduling the connectivity of DTN nodes.

Failure diagnosis capability is the core technology of DTN Monitor and Control system. All diagnostic units form a 'database' which provides approaches to diagnose potential DTN operation failures. How to enrich this diagnosis "database" is one of the most essential technology needed to be implemented for the monitor and control system.

The DTN Monitor and Control System is constructed to interface with some mission operation center, interact with the user via the built-in GUIs, and communicate with failure diagnostic systems or the failure treatment editor to get the solution to the failure. Some research and development can be conducted to build the interfacing protocol between the monitor and control center and each relevant entity. Besides, since failure treatment method involves the update of existing DTN configuration, how to distribute the configuration update to space objects or some designated user is also some topic to be explored.

Besides currently developed features in DTN Monitor and Control system, more network management functions can be further added into current implementation in a systematic style due to its multi-tire client-server design.

5. Acknowledgement

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology and the National Aeronautics and Space Administration.

REFERENCES

[1] V. Cerf. et al., Delay-Tolerant Network Architecture, IETF RFC 4838, informational, April 2007.

[2] S. Burleigh, "Delay-Tolerant Networking for Space Flight Operations: Design and Development," AIAA-2008-3221, SpaceOps, 12-16 May 2008, Heidelberg, Germany

[3] K. Scott, S. Burleigh, Bundle Protocol Specification, IETF RFC 5050, experimental, November, 2007.

[4] S. Burleigh, ION.pdf, with ION open source.

[5] J. Leigh Torgerson, Loren Clare, S. Y. Wang, Joshua Schoolcraft, "The Deep Impact Network Experiment Operations Center", IEEE Aerospace Conference, Big Sky, Montana, March 2009.

[6] Shin-Ywan (Cindy) Wang, J. Leigh Torgerson, Joshua Schoolcraft, Yan Brenman, "The Deep Impact Network Experiment Operations Center Monitor and Control System,", SMT-IT, Pasadena, July, 2009.

[7] Asynchronous Message Service, CCSDS 735.1-B-1. Blue Book, September, 2011.

[8] Roy. Gladden, Mars Relay Operations Service (MaROS):Rationale and Approach, SpaceOps 2010.

[9] DTN2: http://www.dtnrg.org/wiki/Code.

BIOGRAPHY

Shin-Ywan (Cindy) Wang *is a senior member of the Communications Networks Group at JPL. She currently work on tasks related to telecommunication link budget analysis and delay tolerant network (DTN) protocol which includes the DTN Monitor and Control System and video streaming over DTN configuration.*