2010 Second International Conference on Advances in Satellite and Space Communications

Operational Performance of the Context-Aware Broker (CAB): A Communication and Management System for Delay-Tolerant Networks (DTNs)

Cathryn Peoples, Gerard Parr, Bryan Scotney, and Adrian Moore
Faculty of Computing and Engineering, School of Computing and Information Engineering,
University of Ulster,
Coleraine, Northern Ireland.
{c.peoples; gp.parr; bw.scotney; aa.moore}@ulster.ac.uk

***Abstract*—The Context-Aware Broker is a policy-based management system developed by the authors to achieve autonomic communication in delay-tolerant networks. This is in recognition of environment challenges when operating in remote regions, and time, human, and financial resource costs incurred during mission-specific configuration. The Context-Aware Broker seeks to limit cost overheads through achieving a standardised transmitting approach, and operating autonomically to optimise reliability and sustainability levels achieved. In achieving its network management function, a cost-benefit impact is the consequence. Performance results from the Context-Aware Broker's deployment in *ns-2.30* are presented and evaluated in this paper.**

Keywords-autonomy; context-awareness; delay-tolerant network; ns-2.3; policy-based management; quality of service.

I. Introduction

One of the National Aeronautics and Space Administration's (NASA) objectives is to achieve standardisation of hardware and software mission designs to limit development costs. The *Juno* mission to Jupiter, for example, is anticipated to launch in August 2011 and arrive at its destination approximately $968.1\text{x}10^6$ kilometres (km) [1] from Earth in July 2016. Its mission objectives include observing Jupiter to discover more about the planet and its history, much like recent exploration of Mars. In terms of hardware costs and the degree of standardisation achieved, the *Juno* spacecraft uses a spinning solar-powered spacecraft [2] constructed for earlier missions, demonstrating evidence of reusability and a standardised component. Communication software, on the other hand, will be developed on a mission-specific basis. Details on the *Juno* mission are unavailable as yet, but the NASA Mars *Phoenix* mission can be reviewed as an example. Upon arrival at Mars, transmission was suspended to concentrate resources on set-up [3] and initial images were returned in black and white to allow viewing within the shortest delay. Co-operation with the NASA *Mars Reconnaissance Orbiter* (*MRO*) and the European Space Agency (ESA) *Mars Express Orbiter* (*MEO*) missions allowed image captures of *Phoenix* landing on Mars and data return via an alternative route. These characteristics of the communication strategy are mission-specific, resulting from reactions to scenario characteristics (i.e., distance from Earth, imagery data return, and additional *MRO* and *MEO* network resources). A standardised communication strategy has not been a feature of previous deep space missions, and transmission cost in terms of human time and their financial expense to configure mission operation is a limiting factor of future deep space exploration. Achieving autonomic operation and communication in challenged networks therefore continues to be a research gap.

In achieving a standardised and autonomic communication strategy, mission-specific configuration is not required. Contextual data regarding the environment can be collected and used to optimise the cost-effectiveness of the communication approach. Furthermore, autonomy allows reaction to unexpected environment conditions to optimise science return and operational performance. In the case of the NASA *Spirit* rover on Mars, for example, attempts were made to release it from a sand trap over a period of several months after it became immobile in 2009, which involved experiments with a prototype rover in a sandbox on Earth, analysis, and reviews. After ten months, experimental results of the strategy to free *Spirit* are promising, but the Martian winter begins in May 2010 and power levels which will invoke movement are insufficient as of February 2010. The priority is therefore to use remaining power levels to tilt the rover to improve the communication angle before winter arrives. Otherwise, line-of-sight (LOS) operation with Earth will be impossible during the period [4]. This is an example of a scenario where autonomic operation could have improved the rate of decision execution. In being able to self-diagnose and inform mission control on Earth that movement has become restricted, power consumption efficiency can be improved. Such capability is important in the future: the *Juno* spacecraft will be approximately $566.8 \text{ x } 10^6$ km further from Earth than the *Spirit* rover on Mars, resulting in significantly longer round-trip communication times and longer periods of waiting for responses from Earth during which finite node power resources are consumed. Autonomic operation may reduce the number of communication transactions with Earth.

We therefore propose a Context-Aware Broker (CAB) algorithm [5] as an attempt to achieve an autonomic and

978-0-7695-4067-2/10 $26.00 © 2010 IEEE
DOI 10.1109/SPACOMM.2010.28

standardised approach to communicating in Delay-Tolerant Networks (DTNs) to remove the need to perform mission-specific configuration. It incorporates contextual data on application requirements and network environment, and configures transmissions autonomically in response to environment constraints. Its overall objectives are specific to each application scenario, but focus on maximising transmission reliability and sustainability in challenged networks.

The objective of this paper is to present and evaluate a series of CAB performance results in scenarios which experience long-distance and mobility issues common to DTNs. Results exhibit performance improvements introduced by the CAB and measure this against additional configuration, latency, and/or resource costs.

The remainder of this paper is structured as follows: in Section II, the CAB concept is presented, including a high-level representation of its algorithm to allow an understanding of key phases, contextual attributes used, and the way in which policy-based decision-making occurs. In Section III, a range of experiments demonstrating the cost-benefit overhead of the CAB are shown to appreciate ways in which performance improvements and overhead costs occur. The paper concludes and discusses future work in Section IV.

II. CONTEXT-AWARE BROKER (CAB)

The Context-Aware Broker provides an extension to network management protocols to achieve a communication function which is specific to DTN challenges. The CAB employs policy-based decision-making, with execution driven by collected contextual data. Its core functionalities are represented in a summarised high-level algorithm in Fig. 1, which includes examples of the ways that key attributes are evaluated to achieve its Quality of Service (QoS) objective. A discussion of its operation follows: after gaining an appreciation of the communicating environment's context, the CAB steps through a series of phased evaluations, first assessing ability to achieve any level of transmission within environment constraints (`Phase1`), before evaluating ability to achieve individual application QoS requirements at a lower level of detail using all collected and inferred information (`Phase2`). (Also incorporated within these phases is the `Validation` of collected data and `Inferring` of additional context.) The final stage of the evaluation process involves determining requirements of a selected protocol (`Phase3`), including ability to support store-and-forward operation or unreliable mode of transport, for example. This information fuels the protocol selection and its configuration in response to environment constraints. Once `Transmission` begins, the CAB functionality `Monitors` performance achieved and `Predicts` future performance based on past trend information retained within a Historical MIB. In the event that a required performance level is breeched, `Alarms` incite that recovery `Action` should occur. In worst-case scenarios, a transmission may be sent to the Sleep state when the application can cope with this additional latency, or may continue on a Best-Effort (BE) basis when it cannot.

```
Collect context data from Application and DTN
Environment MIBs, then execute CAB algorithm:
Phase1 {
if {(e.g., propagation path can be traversed
within application latency) then
  Phase2 else
  sleep; }} // either suspend for undefined period
// or sleep for predefined period and re-check
Phase2 {
if {(e.g., bandwidth insufficient in relation to
transmission vol. given application latency) then
  adapt transmission vol. in relation to
  bandwidth else
  action not required;}}
Phase3 {;//Protocol capabilities required:
  Store-and-forward mechanism (on/off);
  Unreliable mode of transport (on/off);
  Retransmissions (on/off);}
ProtocolSelection {
  Evaluate protocol ability to support
  requirements identified during Phase3;}
ProtocolConfiguration {
  SCTP: Max. retransmissions (count);
  SCTP: Heartbeat interval (seconds);
  TCP: Max. retransmission timeout (seconds);
  UDP: Packet size (bytes);
  XCP: Timestamps (on/off);}
Transmission {
  send;}
Monitoring {;//Monitor performance attributes:
  Packet drop rate (packets/sec);
  Propagation distance between nodes (seconds);
  Retransmission count (packets/sec);}
Prediction {;//Predict performance attributes:
  Packet drop rate (packet/sec);
  Propagation distance between nodes (seconds);
  Retransmission count (packets/sec);}
Alarms {
  Packets not reaching transport layer dest. node;
  Bandwidth below required transmission rate;
  Queuing latency exceeds app. latency;}
Action {
  Suspend transmission temporarily;
  Send transmission to Sleep state;
  Protocol handover;
  Continue transmission on BE basis;}
main {
  Phase1;
  Phase2;
  Phase3;
  ProtocolSelection;
  ProtocolConfiguration;
  Transmission;
  Monitoring;
  Prediction;
  if (required performance levels breeched) {
    Alarms;
    Action;}}
```

Figure 1. High-Level Representation of CAB Algorithm

Applications are classified according to a set of QoS attributes, which indicate mission-criticality (true/false), interactivity (true/false), acceptable latency (seconds), and acceptable bit error rate (BER), and decisions made by the CAB are driven by these attribute values. A mission-critical application, for example, may not be sent to the Sleep state, but should always progress on a BE basis. Such a decision will be required when, for example, environment context data indicates that available bandwidth is insufficient to achieve required application transmission latency. This demonstrates that it is therefore through careful analysis of application and environment contextual attributes in relation to each other that the CAB communication and network management function is achieved.

III. EXPERIMENTS AND RESULTS

In Section III, a range of experiment results from a deployment of the CAB in *ns-2.30* [6] are presented. The simulation environment responsible for the results obtained will be explained: experiment results demonstrate that the range of transport layer protocols which the CAB utilises are not DTN-specific (e.g., [7]-[9]). This is a consequence of the simulation environments available that allow DTN and associated protocol modelling: a simulation environment does not exist within which DTN operations at the range of stack layers can occur. The Opportunistic Network Environment (*ONE*) simulator [10], for example, has been developed specifically for DTN operating challenges and protocol improvement, but incorporates protocol specification at the network layer only. Another simulation environment, *ns-2.30*, developed for terrestrial wired and wireless network communications, and which allows extensibility to DTN environment modelling but does not incorporate DTN protocol standards, has been chosen. We justify this selection by appreciating that it allows integration and operation of context Management Information Bases (MIBs) and policy-based decision-making specific to the delay-tolerant domain. While DTN-specific protocol operation cannot be observed, an understanding of the CAB's operational costs and performance benefits can be gained.

The cost-benefit influence of the CAB is evident in a range of test scenarios presented in the remainder of this section: In *Test Scenario 1*, a 50-second Voice over Internet Protocol (VoIP) transmission is sent between two nodes which are 1,000 milliseconds apart (or 299,792 km when the link is propagated at the speed of light). Operational performance is measured in terms of the time to transmit between source and destination nodes, with VoIP having interactive requirements (latency between 0.1 and 0.4 seconds to receive once sent is acceptable). The influence of the CAB in this scenario is that it restricts application transmission given distance between nodes and real-time requirements. This action is taken due to the CAB's attempt to restrict resource consumption when latency QoS will not be achieved, an acceptable action as the VoIP transfer is not mission-critical (i.e., failure to receive application traffic will not cause the mission to fail and it can be re-sent at a time when network conditions are more conducive to application requirements).

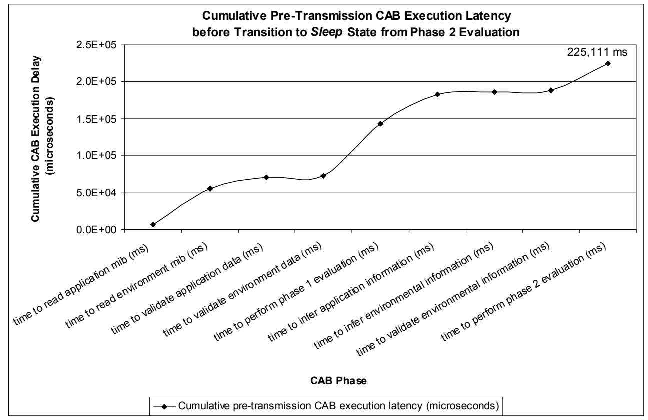

Figure 2. Cumulative CAB Pre-Transmission Execution Latencies prior to Suspension from Phase 2 Evaluation

CAB overhead costs can be observed in this scenario (Fig. 2), incurred when progressing through several stages before reaching the suspend decision during Phase 2 Evaluation. Maximum delay (70,765.7 microseconds (ms)) in this scenario is incurred during Phase 1 Evaluation. Time to read the Environment MIB (48,555.2 ms) and Infer Application Information (39,697 ms) represent the next highest latencies. The total time taken by the CAB to reach this decision, when measured as an average of ten runs, is 225,111 microseconds. Execution latencies consumed during Environment Validation and Environment Data Inferring are minimal in relation to the other latencies. The performance improvement achieved by the CAB in this scenario is that it restricts the non mission-critical transmission because of the one-way propagation delay between nodes, limiting resource consumption when latency QoS will not be achieved. If allowed to progress, it is unlikely that users will be satisfied with the service achieved and resource consumption in restricted environments may be considered wasteful.

Experiments in *Test Scenario 2* explore the CAB's ability to perform intelligent protocol configuration in the dynamic environment prior to transmission beginning. In this example, the Stream Control Transmission Protocol (SCTP) heartbeat (HB), the time interval between heartbeat (control) packet sending, is configured in relation to propagation distance. The network topology simulates a dynamic short- (150 milliseconds) to long-distance (1,650 milliseconds) environment, where 4 megabytes of non mission-critical File Transfer Protocol (FTP) traffic is sent between nodes. A network error rate of 0.4 packets lost per second is applied to ensure that protocol flow and error control mechanisms are invoked.

Transmission latencies incurred when the HB has been intelligently configured in relation to when the default of 30 seconds is used are presented in Fig. 3. (The intelligent configuration is fuelled by previous experiments with SCTP and identification that performance can be optimised as a function of propagation distance, BER and transmission volume.) As a result of the intelligent configuration in this scenario, the HB is lower with less distance between nodes, and increases towards the default as distance increases towards 1,500 milliseconds. For distances beyond 1,500 milliseconds, further benefits are not achieved from intelligent HB configurations. Results in Fig. 3 confirm the configuration suitability: over the range of scenarios tested, there is an average latency reduction by 591.43 seconds when the HB is intelligently configured in relation to when the default is used. The greatest performance improvement is evident in the network distributed over 900 milliseconds, with an average latency reduction by 1,328.92 seconds. Transmission latency is reduced when intelligent configuration occurs because the HB interval allows errors and/or losses to be identified and corrected within a shorter period of time, enabling transmission to complete more quickly. As propagation distance increases, delay savings decrease before the HB becomes more closely matched with the default, which is more suitable given the extending distance between nodes. Errors and losses are identified within more similar periods of time, and there is little difference in overall latency incurred.

Costs and benefits of performing intelligent SCTP configuration in the 1,350 millisecond scenario are explored in Fig. 4 and Fig. 5. In this scenario, these results conform to the expected and ideal impact of the CAB on performance: (1) Running latency is reduced when intelligent configuration is applied; (2) Time of last packet reception is reduced when intelligent configuration is applied; and (3) Pre-transmission latency is increased when intelligent configuration is applied. The FTP application requires full reliability and does not have real-time or interactive latency requirements: QoS is therefore achieved in scenarios both when the CAB is and is not applied. Once the reliable protocol selection has occurred, additional latencies to configure the protocol may therefore be questioned when QoS is achieved in both instances. This additional latency can be justified, however, by additional positive impacts of the intelligent configuration, which include reduced pressure on network resources. By more accurately configuring the HB interval, transmission completes more quickly, CAB monitoring occurs for a shorter period, less bandwidth is consumed during transmission, and fewer node resources are used. In intelligently configuring the protocol, the CAB's objective of maximising QoS within the environment and minimising network resource consumption is achieved.

A core CAB functionality includes monitoring transmission progression and it takes intermediary action if achieved performance declines due to environment dynamics. This may be in response to varying distance between nodes, network bit errors, or changes to LOS connectivity in the delay-tolerant environment. In *Test*

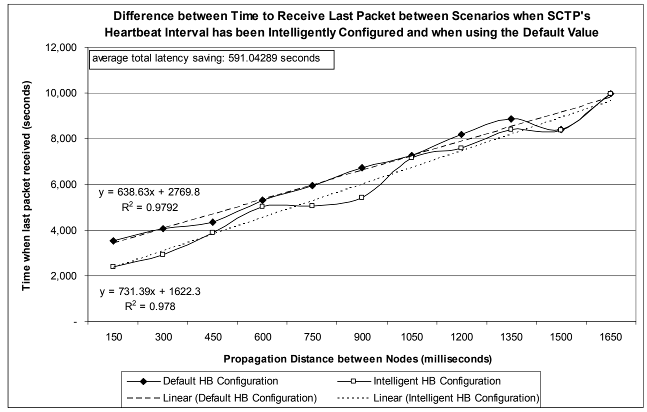

Figure 3. Transmission Latencies when SCTP Heartbeat Intelligently Configured and Default Used

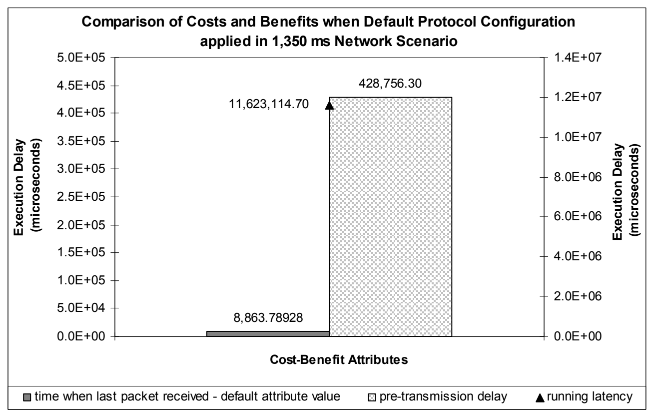

Figure 4. Cost-Benefit Comparison - Default Heartbeat Value

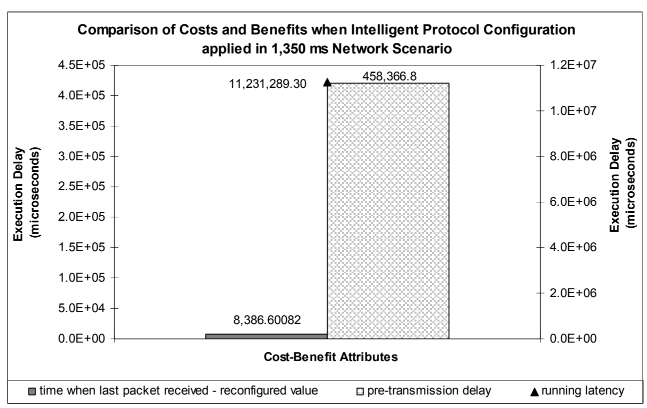

Figure 5. Cost-Benefit Comparison - Intelligent Heartbeat Value Configuration

Scenario 3, the cost-benefit impact of suspending a transmission once it has begun is explored. This scenario, involves a short- (500 milliseconds) to long-distance (2,000 milliseconds) connection communicating a 50-second VoIP transmission between two nodes over a 1 megabyte link.

Once the propagation delay extends to 2,000 milliseconds, the ability to achieve interactivity and real-time response is compromised. The CAB therefore enforces a transmission suspend, allowing observation to determine if distance between nodes will decline within a pre-defined period and allow latency QoS to be achieved. In this scenario, the CAB re-starts transmission after identifying that propagation distance between nodes is not declining. Although the optimum latency QoS will not be achieved, transmission re-starts because the application is mission-critical.

The CAB's effect on packets sent and received can be observed in Fig. 6. Once it detects that latency between packet dequeue from source and enqueue at destination (1.00512 seconds) is greater than the application's maximum latency as defined in the Application MIB (0.15 seconds), it enforces a transmission suspension to identify if propagation distance between nodes is changing and will allow latency QoS, or at least a higher level of QoS to be achieved within the application's maximum overall latency. The transmission of non-application monitoring packets is enforced to assist in network operation observation during the suspend period. After monitoring propagation distance between communicating nodes and identifying that it is static, the CAB enforces transmission completion on a BE basis. While latency QoS is not achieved, the CAB ensures that the mission-critical application achieves reliability QoS.

CAB overhead monitoring latencies during the contextual transmission are shown (Fig. 7), representing the time for execution of monitoring functions to identify if achieved performance levels are acceptable, given application QoS requirements. Maximum monitoring latencies are incurred when evaluating the Historical MIB. It is within this process that node locations and propagation distances are identified, and it is using this information that the CAB decides to re-start transmission of application traffic once it detects that propagation distances are not declining with time. Latencies are higher during this function as the CAB must perform evaluations in response to network changes. This is in contrast to the execution of monitoring processes where high-level evaluations detect that lower-level analysis is not required.

The range of results presented in Section III demonstrate the CAB's influence when striving to optimise application QoS achieved within constrained operating environments. Core to this analysis is consideration for additional costs incurred – we are not suggesting that performance improvements are the only outcome achieved. Our overall objective in constructing the CAB is to optimise the costs incurred while achieving application requirements. In the following section, relationships between cost and benefit consequences in the experimental scenarios are discussed.

A. *The CAB's Cost-Benefit Impact*

A matrix of observed impacts on performance is shown in Table I. to represent CAB operation when transmitting different applications in various network environments.

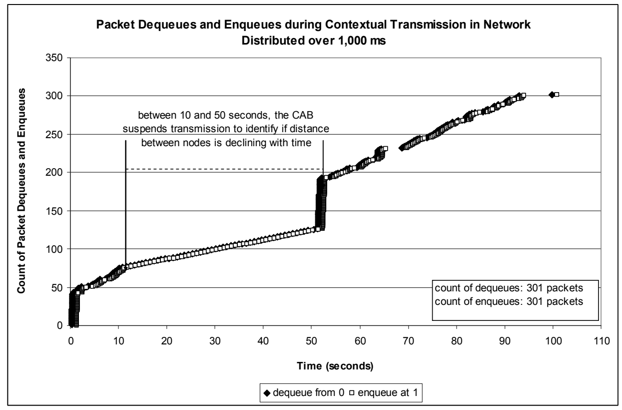

Figure 6. Packet Dequeues and Enqueues during Contextual Transmission

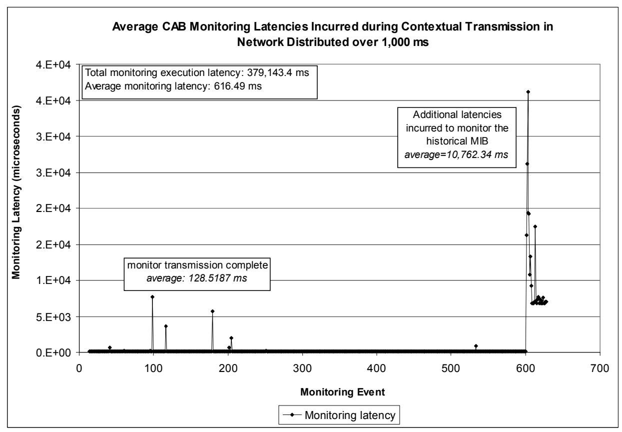

Figure 7. Monitoring Latencies during Contextual Transmission

Performance results are compared against those achieved when the CAB is not applied to the transmission. The objective of the performance matrix is to enable an evaluation of situations where the CAB has the greatest impact on improving performance.

In sending the transmission to the Sleep state during Phase 2 Evaluation in *Test Scenario 1* due to propagation distance between nodes, the CAB's objective is to minimise network resource consumption when latency QoS will not be achieved. This has the cost, however, of not fulfilling the transmission in terms of reducing throughput volume and reducing reliability, timeliness, and interaction associated with the application request. In this scenario, the CAB prioritises resource consumption over transmission of the non mission-critical application to improve the potential that resources will be available when necessary for mission-critical applications.

In the intelligent protocol configuration in *Test Scenario 2* involving the SCTP HB interval, the CAB demonstrates ability to optimise transmission latency QoS and therefore increases timeliness and interaction. When the HB is set

Table I. The CAB's Impact on Performance

Parameters experience either cost/benefit impact on performance									
Increased …					**Reduced …**				
Reliability	**Timeliness**	**Throughput**	**Accuracy**	**Interaction**	**Reliability**	**Timeliness**	**Throughput**	**Accuracy**	**Interaction**
***Test Scenario 1* – CAB Action: Pre-Transmission Suspension**									
-	-	-	-	-	✓	✓	✓	-	✓
***Test Scenario 2* - CAB Action: Protocol Configuration**									
-	✓	-	✓	✓	-	-	✓	-	-
***Test Scenario 3* - CAB Action: Transmission Suspension**									
-	-	✓	-	-	-	✓	-	-	✓

more accurately in relation to the transmission scenario, throughput is reduced because protocol timeouts do not occur inappropriately, forcing retransmissions when not required. For this reason, the contextual transmission is also considered to be more accurate.

The transmission suspension in *Test Scenario 3* displays a contrasting impact on performance achieved. The CAB's impact in this scenario includes increased throughput, reduced timeliness, and reduced interaction. Throughput is increased because the CAB enforces suspension after identifying that latency QoS is not being achieved, and starts a non-application stream to monitor network events during the suspend period, which results in increased traffic flow. Timeliness is reduced when the CAB is applied in comparison to when not due to the suspension and restart after the suspend period. When the CAB is not applied, the transmission completes with the same level of QoS achieved when the CAB is applied and without the suspend interruption. Finally, the CAB results in reduced interaction between the two communicating nodes due to interruption during the CAB suspend. The overall positive impact of the CAB's application in this scenario is therefore more restricted due to requirements of the application being transmitted and the static operating environment. The risk, however, of this occurrence must be embraced to allow more positive CAB impacts to occur when possible.

IV. CONCLUSION AND FUTURE WORK

The CAB development was introduced in Section I as being an attempt to minimise network transmission cost in remote regions such that their rate of deployment may be improved and our understanding of deep space expanded. Results presented in this paper validate that performance improvements as a consequence of context-aware policy-based decision-making are achievable, and work towards achieving QoS or at least a higher level of QoS when environment constraints exist. Additional costs are also incurred, potentially both in terms of increased node resource consumption, processing latency, and throughput, and also with the risk of occurrence without performance improvements, as in the case of *Test Scenario 3*. The CAB has been designed to optimise operational performance for mission-critical applications to maximise the chances that resources are available when required, given the importance of being able to achieve network connectivity in deep space. The risk that costs will be incurred without performance improvements is one which must therefore be accommodated.

With regard to future work, we anticipate extending the CAB's operation to include other protocol layer requirements and improve autonomic ability. Future work also includes continuing CAB development to enhance the optimisation of costs and benefits achieved. The algorithm will be extended for application in other domains with specific network challenges that may be overcome through the CAB's integration. This refers to current research within the India-UK Advanced Technology Centre of Excellence in Next Generation Networks, Systems and Services (IU-ATC). Optimising the energy-efficiency of network communications is a specific research focus of the IU-ATC, and it is believed that the CAB can be extended to accommodate this networking challenge.

REFERENCES

[1] National Aeronautics and Space Administration, "Jupiter Fact Sheet," Nov. 2007; Available at [Accessed: Mar. 29, 2010]: http://nssdc.gsfc.nasa.gov/planetary/factsheet/jupiterfact.html

[2] National Aeronautics and Space Administration, "Juno Spacecraft," Mar. 2009; Available at [Accessed: Mar. 29, 2010]: http://www.nasa.gov/mission_pages/juno/spacecraft/index.html.

[3] National Aeronautics and Space Administration, "NASA's Phoenix Spacecraft Reports Good Health After Mars Landing," May 2008; Available at [Accessed: Mar. 29, 2010]: http://www.nasa.gov/mission_pages/phoenix/news/phoenix-20080525c.html.

[4] National Aeronautics and Space Administration, "Now a Stationary Research Platform, NASA's Mars Rover Spirit Starts a New Chapter in Red Planet Scientific Studies," Jan. 2010; Available at [Accessed: Mar. 29, 2010]: http://marsrover.nasa.gov/newsroom/pressreleases/20100126a.html.

[5] C. Peoples, G. Parr, B. Scotney, and A. Moore, "A Reconfigurable Context-Aware Protocol Stack for Interplanetary Communication," in *Proc. 3rd IEEE Int. Workshop on Satellite and Space Communications*, Sep. 2007, pp. 281-285, doi: 10.1109/IWSSC.2007.4409431.

[6] Network Simulator *ns-2.30*; Available at [Accessed: Mar. 29, 2010]: http://isi.edu/nsnam/ns/.

[7] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider Transmission Protocol – Specification," *Internet Engineering Task Force Request for Comments 5326*, Sep. 2008.

[8] K. Scott and S. Burleigh, "Bundle Protocol Specification," *Internet Engineering Task Force Request for Comments 5050*, Nov. 2007.

[9] L. Wood, J. McKim, W. Eddy, W. Ivancic, and C. Jackson, "Saratoga: A Scalable File Transfer Protocol," *work in progress as an Internet Draft*, Oct. 2009.

[10] Opportunistic Network Environment (*ONE*); Available at [Accessed: Mar. 29, 2010]: http://www.netlab.tkk.fi/tutkimus/dtn/theone/.