Internet Research Task Force
Internet-Draft
Intended status: Experimental
Expires: September 9, 2012

S. Farrell
A. Lynch
Trinity College Dublin
D. Kutscher
NEC
A. Lindgren
Swedish Institute of Computer Science

March 8, 2012

Bundle Protocol Query Extension Block
draft-farrell-dtnrg-bpq-01

Abstract

The Bundle Protocol (BP) provides store-and-forward networking for Delay- and Disruption-Tolerant Networks. This document defines the BP query extension block (BPQ) which allows applications to query the stores of nodes on the path along which a bundle containing a bundle query extension block is routed.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 9, 2012.

Copyright Notice

carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction

The Bundle Protocol (BP) specified in RFC 5050 [RFC5050] provides store-and-forward networking for Delay- and Disruption-Tolerant Networks (DTNs). RFC 4838 [RFC4838] This document defines the BP query extension block (BPQ) which allows applications to query the stores of nodes on the path along which a bundle containing a bundle query extension block is routed.

The DTN architecture and the Bundle Protocol can be used for different applications and provide a certain degree of flexibility for naming sources and destinations, as well for deciding how to process and forward bundles at nodes in a DTN network.

In some applications contexts, the Bundle Protocol is used for literally transmitting some payload data from one endpoint to another -- potentially leveraging store-and-forward capabilities of intermediate nodes to overcome disruptions. How intermediate nodes perform their forwarding decisions is not specified by either the DTN architecture nor the Bundle Protocol specification, but often the destination endpoint identifier (EID) would be considered.

But EIDs in a DTN network do not necessarily have to represent single nodes -- they can be used for representing receiver groups or for specifying some requested service in the network. This flexibility, together with the option of using different approaches for disseminating data to nodes in a network, has made DTN an attractive candidate technology in a range of content distribution scenarios, for instance for publish-subscribe-based content distribution [ref.dpsp] and for time-aware content dissemination through info stations [ref.taco-dtn].

In some scenarios, DTN bundles can have query semantics, i.e., a bundle is sent in order to query for some information object -- or a copy of it that can be available on some DTN node as the result of a specific dissemination/routing strategy. Thus, sometimes when you send a query in a DTN, an intermediate BP node already has the data you want, and there should be a way to get that data, without having to go all the way to the "source" of the data which is, of course, the destination for a query bundle.

The BPQ that is specified in this memo is intended to allow such queries that can be answered by intermediate BP nodes, where those nodes do not necessarily have to be addressed by the destination EID of a corresponding request message.

A use case: Alice and Bob both want to get a video. Alice first asks for this using the BP. Now Bob, who's nearby Alice also wants to see

the same video, and as it happens, due to the routing scheme in force, the video is still stored at Bob's "next hop" DTN router. Wouldn't it be nice if Bob could just query the DTN as a whole and in this case, get the response he wants from a nearby node via probably far less delayed or disrupted links. The BPQ extension block defined here provides a way to enable this kind of re-use of DTN router storage.

The BPQ extension block is intended as an enabling mechanism for such applications without anticipating a specific behaviour with respect to EID semantics and routing strategies. Also, it is intended as an optional enhancement to DTN node implementations and does not require all nodes in a network to actually support the extension to be useful. In Section 2 we provide an overview of the general protocol operation, Section 3 specifies the actual BPQ block format, and Section 4 provides the processing requirements for DTN nodes. Section 5 describes a few non-normative application considerations for BPQ, and Section 6 refers to related work.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Protocol Overview

The basic idea of the query extension block is that a "query" bundle can contain whatever information is required for the query to succeed when that bundle reaches its destination. This information is typically contained within the BPQ extension block but may also be sent in the application layer payload.

One, though not the only, possibility is that the query-response payload has a name (e.g. a file name or a name derived from the query or response payload via hash functions). In such cases, the BPQ extension block can simply contain the data required (plus ancillary data as described below).

BP nodes that do not support BPQ simply (store, and) forward query bundles and response bundles as normal and are unaffected.

BP nodes supporting BPQ compare the value of the BPQ in an inbound bundle against their bundle cache (details below) and when a matching bundle is found they then respond to the source of the query bundle with a bundle containing the payload of the matching bundle, together with BPQ data that allows other DTN nodes to also successfully match the query and response.

When a match is found in the cache, the response bundle that is sent SHOULD be a copy of the bundle in the cache. If the original bundle is sent more than once, subsequent BP nodes may discard the bundles as a duplicates. As the BP uniquely identifies bundles from their creation time-stamp (seconds and sequence number) and their source EID, only changing the destination is not sufficient to avoid duplicate detection.

When a response is copied the source EID of the response bundle MUST be set to the local EID and the creation time-stamp MUST be updated to the current time. This changes the bundle ID and prevents it from being discarded as a duplicate. To retain the original ID, an immutable copy of the original creation time-stamp and source EID is stored in the BPQ.

When a bundle is copied and its time-stamp is updated, the bundle's lifetime MUST also be adjusted. The lifetime value is relative to the creation time-stamp. If left unchanged, copying a bundle would inadvertently extend the bundles lifetime.

Various schemes could be used in order to allow matching the query bundle with a bundle stored on a node, but the simplest way to handle this is simply for there to be an identical unique BPQ value in the response bundle. (The response bundle must also be marked as a response in order not to confuse Alice and Bob's separate queries for the same video.)

If a node supporting BPQ finds a complete (i.e. not fragmentary) matching response bundle, then it can produce a copy-response bundle for the requester with the same payload and send that back to the source of the query bundle. In this case, the query bundle need not be forwarded further and can be deleted. For this reason payloads contained in query bundles are not guaranteed to be delivered to their destination.

[question: what to do with status reports in this case if the query bundle asked for them? they're not a good idea in any case but I suppose we should say]

BPQ aware nodes can store and match against response fragments as well as complete responses. Depending on the content, it does not always make sense for fragments to be cached. For example, the use of intentional names can result in more than one response sharing the same name while having different payloads. In this case nodes SHOULD NOT try to cache fragments or partially answer queries, since fragments from different payloads SHOULD NOT be recombined. For this reason there are two kinds response, normal responses for which fragments may be cached and reassembled, and responses where this is

restricted. This in no way influences caching of complete bundles.

In the event that the matching response bundle is only a fragment, then the node discovering that match responds with a copy-response bundle containing the fragment. The node SHOULD forward a modified query bundle which reflects the matched fragment, so that other fragments may be retrieved from elsewhere on the query bundle's path.

Fragments already marked in the query as matched SHOULD NOT be re-sent. A matched fragment that is a super-set of a previously matched fragment (including a complete match) MAY be returned. If the node finds a set of matching fragments that fully cover the payload, then the node SHOULD NOT forward the query bundle.

[question: should a single copy-response be sent, thus re-assembling fragments, or should the just send each fragment as a bundle as it itself received? think about PIB in that case. - ans below.]

Fragments MAY be re-assembled by a cache before being returned. This is NOT required as fragments originating from different bundles may contain additional (different) extension blocks. Intermediate nodes MAY respond with multiple fragments for the destination to recombine.

In order to allow matching, response bundles (and all fragments thereof) sent out by the "source" of the response, also include a BPQ. In this way, nodes that support BPQ can easily match queries and responses.

In principle, there could be many ways to match a query bundle with a response bundle. For example, the query bundle could contain a SQL-like query and the response bundle BPQ extension could contain a database that returns a non-Null response when the query is "executed" by a node. This document however, only specifies an "exact match" matching rule, where the query and response bundles only match if both contain the same set of bits. Other matching rules may be defined in future.

Since response bundles containing the BPQ are intended to be re-used, it would appear to be sensible to store such bundles for as long as possible, regardless of routing decisions. (Routing schemes may also call for bundles to be stored, even after having been forwarded one or more times.)

[question: the bundles currently live in the cache until they expire or are evicted due to size restrictions. Popular content could have its lifetime extended but this prevents the bundle expiry being used for cache coherency.]

3. BPQ Block Format

The BPQ consists of:

- Block type code (1 byte) defined as in all bundle protocol blocks except the primary bundle block (as described in the Bundle Protocol). The block type code for the BPQ Block is 0xC8 (this is an experimental type)

- Block processing control flags (SDNV) - defined as in all bundle protocol blocks except the primary bundle block. SDNV encoding is described in the Bundle Protocol. If a bundle node receives a bundle with a BPQ block and it is capable of supporting the BPQ block but it is not able to parse and process the BPQ value itself, either because it does not support the kind or type being used or because the data is not well-formed, the bundle node MUST process the bundle as if it cannot process the BPQ block. That is, it must operate according to the settings of the Block Processing Control Flags, including the "Delete bundle if block can't be processed" flag and the "Discard block if it can't be processed" flag. The "Block must be replicated in every fragment" bit MUST be set in all BPQ extension blocks.

- Block data length (SDNV) - defined as in all bundle protocol blocks except the primary bundle block. SDNV encoding is described in the bundle protocol.

Block-type-specific data fields as follows:

- A BPQ-kind field (1 byte), with 0x00 meaning a query, 0x01 meaning a response, and 0x02 meaning a response for which fragments MUST NOT be cached. Other values are reserved.

- A matching rule type (1 byte) that tells routers how to match the BPQ from a query with the BPQ of a response. Only matching rule type 0x00 is defined here, which represents the "exact match" rule as further defined below. The matching rule MUST be the same in both a query and response in order for there to be a match.

- An original creation time-stamp (seconds and sequence number). This is the same pair of SDNVs that are initially set in the bundle's primary block. This time-stamp MUST NOT be modified once set. This MAY be left unset for query bundles.

- An original source EID length field (SDNV) the contains the length of the original source EID.

- An original source EID. Since this may be changed in the primary block when the bundle is copied, this original copy is required to uniquely identify the bundle. This EID MUST NOT be modified once set. This MAY be left unset for query bundles.

- A BPQ-value-length field (SDNV) the contains the length of the BPQ-value.

- A BPQ-value field with the length indicated by the BPQ-value-length field, that identifies the relevant response payload and is interpreted according to the matching rule field.

- The number of fragments already returned (SDNV)

- Where the number of fragments already returned is non-zero, an ordered list containing offset, length pairs (encoded as SDNVs) indicating previously matched fragments. This MAY be recalculated as new fragments are found and overlapping (or adjacent) fragment information MAY be merged.

4. BPQ Processing

If no match is found, then the node MUST forward the query bundle as if the BPQ block were not present.

When creating the copy-response bundle, the source EID of the response bundle MUST contain an EID for the node that found the match. The reply-to EID of copy-response bundle SHOULD be set to the reply-to EID of original response bundle and the destination EID of the copy-response bundle MUST be set to the source EID of the query bundle.

The creation time-stamp of the bundle MUST be set to the current time. The difference (in seconds) between the previous time-stamp and the current time MUST be subtracted from the bundle's lifetime value.

The payload and all other extension blocks present in the response bundle MUST be copied into the copy-response bundle.

Basically, the only difference between a response bundle and a copy-response bundle is the bundle identifier and the source and destination EIDs and the time-stamp and lifetime values.

[note: need to check other primary block fields and say what, if anything, to do for each, e.g. for current custodian etc. - a bit of thought needed.

When a node is comparing a query bundle against a potential matching bundle using the exact match matching rule, the bundles match iff the BPQ-value field of both are identical.

If a matching response bundle is not a fragment, then the query bundle SHOULD NOT be further forwarded by the node in question but SHOULD be deleted after the response bundle has been queued for transmission, as the query has been satisfied.

If a matching response bundle is a fragment, then the node SHOULD continue searching in case it has other fragments that match the query. In that case, each fragment is sent as a separate copy-response bundle. That is, the node finding the match SHOULD re-assemble the fragments of the entire bundle, if the node knows how to combine the different sets of extensions blocks of the fragments. If the matching node does not know how to combine the extension blocks, it MUST NOT re-assemble the fragments. The reason is that each fragment could have different sets of extension blocks present and the node might not know how to combine those properly.

If a matching response bundle is a fragment, and the node does not have a full set of fragments (that together contain the entire payload) then the node MUST forward the query bundle as would have happened had no match been found.

Custody and status report settings for the copy-response bundle bundle SHOULD be set to the same values are were present in the matching response bundle unless the node is specifically configured to do otherwise.

[question: is that right? do we really want all those reports and custody acks? There may be a wrinkle there with custody.]

[Lots more tedious but obvious detail TBD.]

5. Application Considerations

This section provides some non-normative considerations on how BPQ can be used.

5.1. Usage of Endpoint Identifiers in Bundles

DTN EIDs usage for BPQ queries and replies needs to be considered for:

- source EIDs in query bundles

- destination EIDs in query bundles
- source EIDs in (copy-) response bundles
- destination EIDs in (copy-) response bundles

Source EIDs in query bundles should normally be set to the node EID of query originator.

For the destination EID in query bundles, there are three different options:

1. Some destination EID if the source of some content is actually known). This could also be interpreted as a default EID for a node to which the query should be forwarded to.
2. Some namespace or application identifier such as "dtn:appX"
3. An actual information object ID (perhaps with a namespace prefix) such as "dtn:appX:45a87e5d"

Ideally, the destination EID for queries should allow non-BPQ-aware DTN nodes to "do the right thing", i.e., forward the bundle to a node with (a copy of) the requested resource. More concretely, specific DTN routing protocols should still work as intended, and these protocols normally perform decision based on the destination EID.

For the source EID in (copy-) response bundles, there are essentially two options:

1. The EID of the origin DTN node for the requested resource
2. The EID of the node that generating the reply, which could be an intermediate BP node that happens to have a matching resource for the request. This option would make the operation of intermediates visible to the actual receivers (normally considered a desirable property) but would end-to-end security (that is based on the source EID).

The destination of (copy-) response bundles should normally set to the node EID of the original sender of the request to enable a DTN network to forward the response bundle to this node. However, specific application scenarios may want to leverage DTN multicast capabilities, e.g. when many nodes are interested in a specific resource so that other EID naming strategies become more attractive.

5.2. Advanced Processing of Query and Copy-Response Bundles

In some named-content distribution scenarios, BPQ nodes can perform additional operations compared to either returning matched bundles or forwarding request bundles. An intermediate BPQ node could also keep an interest table for the requested resource and then later, when a matching resource is available, satisfy the pending requests. This mode of operation could be extended to fragments as well: an intermediate BPQ that has received a request for resource A, for which it has only fragments, could decide to send the fragment(s) directly, or -- e.g. in case of a disruption -- maintain the pending request and complete the response bundle with received fragments until a (more) complete response bundle is eventually sent.

When an intermediate BPQ node follows the strategy of maintaining a list of pending requests, there might be a number of requests for the same resource, e.g., for popular content. For such scenarios it would be beneficial to not have to create individual response bundles for the same resource to be sent to each interested node on the network. Domain-specific routing protocols and adequate usage of destination EIDs could be employed in these cases.

6. Related Work

Using the Bundle Protocol to query the network for near-by resources has been explored in different approaches. Greifenberg and Kutscher have described a DTN Publish-Subscribe Protocol (DPSP) in [ref.dpsp] that allows interested nodes to register interest in some names resource to the network. DPSP nodes would aggregate such subscriptions and forward it towards the direction of an origin node. Corresponding content bundles would be distributed along a tree that has been built implicitly by the subscription messages. In DPSP, destination EIDs in subscription bundles specify the named resource (e.g. content channel), and the subscription information is conveyed in an extension block to enable inter-working with unmodified DTN nodes and routing protocols.

Sollazzo, Musolesi and Mascolo have described a Time-Aware COntent-based dissemination system for DTNs (TACO-DTN) in [ref.taco-dtn] that takes time-based information into account to optimize content dissemination in a subscription-based approach. Temporal profiles are associated to each subscription and allow the construction of temporal profiles of info-stations.

More general, the idea of accessing named information objects in the network, regardless of the actual object location, is a key notion in different Information-Centric Networking approaches. Ahlgren,

D'Ambrosio, Dannewitz et al have developed an elaborate information model for such information objects in [ref.netinf-design]. In the Network of Information approach, information objects can be accessed by unique names that provide additional properties such as self-certification, i.e., provide a cryptographic relation between the object and the name. In such an approach, interested nodes would query the network for specific named objects and the network would perform name-based routing and/or resolution to locators to satisfy such requests.

A similar approach is the Content-Centric Networking (CCN) approach described by Jacobsen, Smetters, Thornton et al in [ref.ccn]. In CCN, network nodes receive so-called Interest Packets for names content from interested nodes. Such Interest Packets can be aggregated, and forwarded according to name-based routing information. Corresponding data packets are forwarded in the reverse direction, based on Interest Table state that is maintained at intermediate nodes -- quite similar to the DPSP approach described above.

The Query Extension Block as described in this memo could be used to inter-connect DTNs to such Information-Centric Networks and/or to implement Information-Centric Networking with the Bundle Protocol.

7. IANA Considerations

We'll want an extension block number and maybe a new registry for query kinds and matching rule types if we stick with the above.

8. Security Considerations

The BPQ in principle allows a node to probe the storage of another node. If BPQ-values are guessable, then this would work. If this is a concern, the unguessable BPQ-values SHOULD be used.

The BPQ imposes a load on nodes that support it. If such a load is considered a potential DoS vector, then nodes SHOULD implement some controls on the amount of searching they are willing to carry out. This could be a simple limit, or could depend on the source (or authentication status) of the query bundle.

Since the copy-response comes from the matching node, the response bundle's authentication information (e.g. PIB) will not be usable with the copy-response.

[note: not sure what to do about this as yet.]

If confidentiality of the query-response payload is required the PCB block can be used to provide that service. However, BPQ values could leak information about the payload, for example if the BPQ value were a hash of the payload, then the BPQ value would allow an attacker to check whether a guess of the payload value was correct or not. If this is a concern, then BPQ values SHOULD be chosen so as not to leak information about the response payload.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5050] Scott, K. and S. Burleigh, "Bundle Protocol Specification", RFC 5050, November 2007.

9.2. Informative References

[RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst, R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant Networking Architecture", RFC 4838, April 2007.

[ref.ccn] Jacobsen, K, D, F, H, and L, "Networking Named Content", CoNEXT 2009 , December 2009.

[ref.dpsp] Greifenberg and Kutscher, "Efficient Publish/ Subscribe-based Multicast for Opportunistic Networking with Self-Organized Resource Utilization", The First IEEE International Workshop on Opportunistic Networking (WON-2008), March 2008.

[ref.netinf-design] Ahlgren, D'Ambrosio, Dannewitz, Marchisio, Marsh, Ohlman, Pentikousis, Rembarz, Strandberg, and Vercellone, "Design Considerations for a Network of Information", Re-Arch 2008 Workshop , December 2008.

[ref.taco-dtn] Sollazzo, Musolesi, and Mascolo, "TACO-DTN: A Time-Aware COntent-based dissemination system for Delay Tolerant Networks", MobiOpp 2007 Workshop , 2007.

Appendix A. ChangeLog

This section to be deleted later. The most recent changes should be added to the end of the list.

Stephen: initial version

Dirk: added some text to the introduction

Dirk: moved some text from introduction to separate section "protocol overview"

Dirk: changes processing requirements for fragmented response bundles as discussed

Dirk: added section on Application Considerations

Dirk: added text to related work section

Aidan: Added text about adding already-returned fragments to the query

Stephen: Added payload confidentiality sec. cons. note

Aidan: Added text intentional naming and combining fragments using original src eid

Authors' Addresses

Stephen Farrell
Trinity College Dublin
Dublin, 2
Ireland

Phone: +353-1-896-2354
Email: stephen.farrell@cs.tcd.ie

Aidan Lynch
Trinity College Dublin
Dublin, 2
Ireland

Phone:
Email: lyncha6@tcd.ie

Dirk Kutscher
NEC
Kurfuersten-Anlage 36
Heidelberg,
Germany

Phone:
Email: kutscher@neclab.eu

Anders Lindgren
Swedish Institute of Computer Science
Stockholm,
Sweden

Phone:
Email: andersl@sics.se