The SPINDLE disruption-tolerant networking system

Conference Paper · December 2007

DOI: 10.1109/MILCOM.2007.4454942 · Source: IEEE Xplore

Ram Ramanathan
Raytheon BBN Technologies
111 PUBLICATIONS **8,348** CITATIONS
SEE PROFILE

Armando Caro
Raytheon BBN Technologies
39 PUBLICATIONS **572** CITATIONS
SEE PROFILE

Regina Rosales Hain
Raytheon BBN Technologies
17 PUBLICATIONS **2,145** CITATIONS
SEE PROFILE

David Mankins
HP Vertica
12 PUBLICATIONS **225** CITATIONS
SEE PROFILE

The SPINDLE Disruption-Tolerant Networking System*

Rajesh Krishnan, Prithwish Basu, Joanne M. Mikkelson, Christopher Small, Ram Ramanathan,
Daniel W. Brown, John R. Burgess, Armando L. Caro, Matthew Condell, Nicholas C. Goffee,
Regina Rosales Hain, Richard E. Hansen, Christine E. Jones, Vikas Kawadia, David P. Mankins,
Beverly I. Schwartz, William T. Strayer, Jeffrey W. Ward, David P. Wiggins, and Stephen H. Polit
BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, USA

Abstract— **DARPA's Disruption-Tolerant Networking (DTN) program is developing technologies that enable access to information when stable end-to-end paths do not exist and network infrastructure access cannot be assured. DTN technology makes use of persistence within network nodes, along with the opportunistic use of mobility, to overcome disruptions to connectivity.**

In this paper, we describe the SPINDLE Disruption-Tolerant Networking system and related technology being developed at BBN under the DTN program. Using an open-source, standards-based core with a plugin architecture and well-specified interfaces, we enable independent development and insertion of innovative DoD-relevant technology while allowing the core system to be refined and engineered within a COTS context.

SPINDLE technology innovations include: (i) routing algorithms that work efficiently across a wide range of network disruption, (ii) a name-management architecture for DTNs that supports progressive resolution of intentional name attributes within the network (not at the source), including support for "queries as names" and name-scheme translation, (iii) distributed caching, indexing, and retrieval approaches for disruption-tolerant content-based (rather than locator-based) access to information, and (iv) a declarative knowledge-based approach that integrates routing, intentional naming, policy-based resource management, and content-based access to information.

We present preliminary results that show that the DTN approach outperforms traditional end-to-end approaches across a wide range of network disruption.

I. Introduction

The US military is transforming into an agile distributed network-centric force. The new doctrine is critically dependent on access to mission information even under temporary disruptions to connectivity in the Global Information Grid (GIG). DARPA's Disruption-Tolerant Networking (DTN) program is developing technologies that enable access to information when stable end-to-end paths do not exist and infrastructure access cannot be assured. DTN technology makes use of persistent storage within network nodes, along with the opportunistic use of mobility, to overcome disruptions to connectivity.

Traditional TCP/IP networks rely on stable end-to-end connectivity – an identifiable path all the way to the destination. In the Department of Defense's wireless tactical networks, connectivity is often disrupted by terrain, weather, jamming, movement, or destruction of nodes. Such disruption makes it impossible to determine a path, halting the flow of data.

In contrast to TCP connections, DTNs communicate opportunistically using episodically or intermittently available links. Information is organized into *bundles* (a concept developed by the DTNRG, an Internet Research Task Force Working Group) and routed through "custodians" that augment the capabilities of traditional routers by persistently storing the bundles and then advancing them to the next available node en route to their destinations [1], [2]. DTN routing takes advantage of mobile nodes (e.g., unmanned aerial vehicles) in an entirely new way, by using them to "haul" data when there is an obstacle in the path – geographic or structural – or in the presence of an enemy threat.

In this paper, we describe the SPINDLE DTN system being developed at BBN under the DARPA DTN program. This system builds upon *DTN2*, the DTNRG reference implementation. This paper is a work-in-progress overview of ongoing technology development in this field at BBN. We are in the second phase of the program at the time of writing this paper. Continued research and development are underway in order to achieve a transitionable, and evolvable (long-lived) capability for the DoD.

Using an earlier prototype system in Phase 1, we clearly demonstrated the benefits of DTN technology. We demonstrated 100%-reliable delivery of data with less than 20% availability of links with greater than 80% utilization of link capacities. The DTN approach consistently outperformed traditional end-to-end approaches across a wide range of network disruption. Under certain worst-case network dynamics, DTN was able to deliver data reliably, while the traditional end-to-end approach broke down and delivered no data at all.

Figure 1 illustrates the performance of DTN in a 4x5 grid emulated network with adversarial link dynamics in which up times of adjacent links are negatively correlated. DTN store-and-forward routing delivers all the offered load for availability as low as 15% whereas the traditional approach that relies on contemporaneous end-to-end paths is seriously disrupted. Details of our work in Phase 1 can be found in [3].

Our vision for DTN is a radical departure from the current Internet model of locator-based access to information. The

* Invited Paper. SPINDLE stands for Survivable Policy-Influenced Networking: Disruption-tolerance through Learning and Evolution. Contact: Rajesh Krishnan *krash@bbn.com*. This work was supported by contract W15P7T-06-C-P638 under the DARPA DTN program. All information contained in this paper is derived from material that has been approved for public release by DARPA and CECOM. 1-4244-1513-06/07/$25.00 ©2007 IEEE

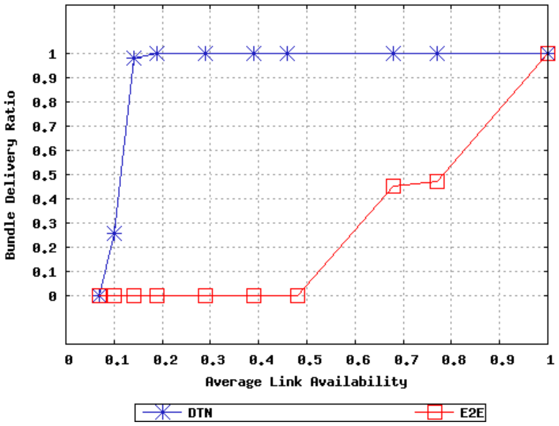


Fig. 1. DTN routing vs. end-to-end communications over traditional routing

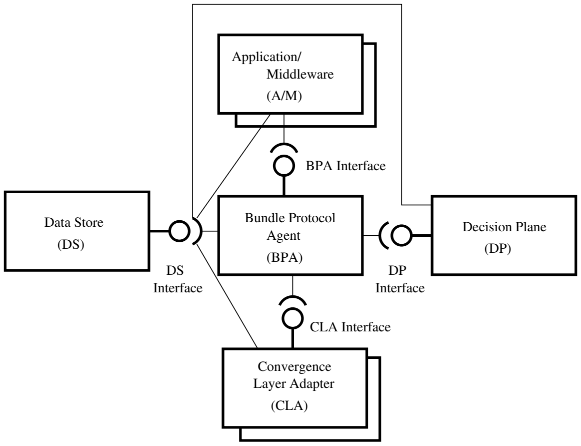


Fig. 2. Architectural Components and Interfaces

traditional model relies on strong connectivity to naming and search infrastructure, but today a new model is necessary, one that provides disruption-tolerant content-based access to information. When a warfighter needs a map and it happens to be cached on a computer in the team he is connected to, his query must be able to get the information even though he is disconnected from its usual source. To realize this enhanced vision, we are developing distributed, opportunistic algorithms for caching, indexing, and retrieval in a manner that maximizes the availability of information required for the mission, even when disconnected from the Internet.

In addition, we are performing research and development in the related areas of scalable routing, policy-based resource management, late binding of name attributes, and security management of persistent information within a DTN.

To faciliate transition, we are developing prototype deployable DTN systems (hardware and software) to be made available to selected DoD transition partners. These prototype systems are being developed with interfaces to selected commercial and military routers / links, and to form DTN overlays upon existing DoD networks. We are evaluating various deployment possibilities including alternative implementations, different physical hardware and virtual machine platforms, and potential for insertion into tactical radios.

The rest of the document is organized as follows. We describe the architecture and interfaces of the SPINDLE sytem in section II. We present four main focus areas of our research next: routing in section III, late binding of intentional name attributes in section IV, content-based access to information in section V, our declarative knowledge-based approach in section VI. We present plans for testbeds and evaluation in section VII. We conclude with a summary of the contributions of the SPINDLE project in section VIII.

II. SPINDLE System Architecture and Interfaces

In this section, we describe the modular SPINDLE system architecture [4]. A particular goal of this architecture is to enable the community to experiment with multiple solutions for persistent storage, routing, naming and late binding, policy, and DTN convergence layers, while continuing to build upon a common core of open DTNRG standards and software. This is accomplished by defining interfaces that allow the experimental features to plug into the common core.

The high-level components and interfaces of the system are illustrated in Figure 2. The components are implemented in separate processes to enable language and tool-chain independence across components. The components are implemented as plugins in the SPINDLE system; in other words, a newly developed module (implemented as a process) can be started on the node and its services will be automatically available to the other already running DTN processes.

Communications between the BPA and other components use an intercomponent communication protocol based on XML messages that are exchanged over either a multicast or unicast socket. The use of XML enables standards-based interoperability.

A. Bundle Protocol Agent

The bundle protocol agent (BPA) of a DTN node offers bundle protocol (BP) services. BPA functions include bundle forwarding, fragmentation and reassembly, custody transfer mechanisms, delivery to application, deletion, sending administrative bundles such as status reports, and security functions. The BPA executes procedures of the BP and the Bundle Security Protocol (BSP) with help from other components in the system. The BPA must comply with the DTNRG specifications for the BP [2] and the BSP [5]. The BPA manages links and supports the five types of links discussed in [1].

In addition, the BPA provides the bundle protocol agent interface that can be accessed by applications, and uses the DP interface, the CLA interface, and the DS interface.

The BPA is responsible for implementing the mechanisms of the bundle protocol, but any key decisions (for example, those based on particular policies or optimization strategies) that need to be taken during the execution of such mechanisms are the responsibility of the *decision plane (DP)* module. Typically these decisions are indicated within the BP and BSP specifications as local policy or implementation issues.

B. Decision Plane

The decision plane component includes four modules:

Policy module: Basic functions include interpretation and enforcement of user-specified policy. We have adopted an "event-condition-action" design that mediates the communications between the BPA and the rest of the DP; however, this module allows an alternative style of "consultation" if needed.

Router module: Functions of a router module include unicast and multicast route computation, generation of next hop(s) for bundles, replication and forwarding, bundle scheduling, and decision to take custody of a bundle, or to discard a bundle. In addition, the router module is responsible for determining *what* network state to distribute, and *to whom*, and *when*. It is also responsible for gathering network state from incoming dissemination bundles and the local CLAs. Future BPAs will support additional forwarding and scheduling functionality configurable through "rules" for enhanced performance. In such cases, the router module will configure these general rules rather than provide separate forwarding decisions for each bundle.

Naming and Late binding module: This module maintains and opportunistically shares name ontologies (or *schemes*, as referred to in [2]) among DTN nodes. Late binding is described in section IV. This module is called upon to resolve rich intentional names to canonical endpoint identifiers of care-of nodes that are subsequently used by the router module. . This is also responsible for registration and dissemination or synchronization of name KBs (in particular, intentional name to canonical name bindings stored within).

Content-based access module: This module is responsible for content caching/replication, distributed indexing, and content-addressable search. This may use several services from other DP modules, e.g., dissemination, late binding, external router.

C. Convergence Layer Adapter

A convergence layer adapter's function (as described in [2]) is to send and receive bundles on behalf of the BPA. It achieves this by utilizing the services of the native protocol, which is supported in one of the *underlying* networks that the node is homed on. A CLA thus adapts the data transmission service provided in the underlying network (e.g., TCP/IP, Bluetooth, other tactical links) to an abstract bundle transmission service that presents itself to the BPA.

A CLA is responsible for discovering and maintaining information about *links*. A measurement process (in the convergence layer adapter) can track and update the values of link attributes such as status, schedule, data rate, and delay. The CLA provides link information to the BPA by posting events. This information may then get relayed to the DP and may also be stored in the DS. The DP can pass messages to the CLA (through the BPA) to configure convergence layer parameters.

D. Data Store

The data store (DS) module offers a persistent storage service and is responsible for storing bundles, knowledge about bundle metadata, network state information (routing tables, name ontology data, content metadata, policy rules), and application state information (registrations and other metadata). The DS provides a service that is accessible from all other components of the system. In order to cater to a wide spectrum of deployment scenarios, a DS could exist in one of the following incarnations of increasing capability:

Key-value: A simple key-value store that allows other modules to add, retrieve and delete key-value pairs. Bundles, bundle metadata and network state information can exist in such a store. This may be applicable for resource-limited devices such as battery-powered sensor nodes.

Database: A database (e.g., based on an RDBMS) that allows other modules to add, delete, and get elements with multiple fields (by its key).

Knowledge Base: A knowledge base (KB) that supports deduction or inference by means of execution of rules (e.g., Prolog style) on stored facts. A KB includes support for data storage, either internally, or via a back-end storage system such as MySQL or Berkeley DB.

E. Application/Middleware

The application and middleware (A/M) agent uses the BP services to transmit and receive bundle payloads.

F. Interfaces

There are four public interfaces defined for the SPINDLE system. (1) The BPA implements the BPA interface that can be used by A/M and DP to send and receive bundles. (2) The DP implements the DP interface that is accessed by the BPA to serve various decision points identified in the BP such as routing, scheduling, name resolution, custody acceptance, and storage management. (3) The CLA implements the CLA interface which is accessed by the BPA for bundle transport and link management. (4) The DS implements the DS interface for storing bundles, bundle metadata, application registrations, connectivity information, and other system state. The DS interface can be accessed systemwide. The CLA may access the DS interface for storing and retrieving connectivity information that is discovered or scheduled/specified. The DP, for example, may access the DS interface for storing and retrieving information related to routing, naming, policy, and content caching/indexing.

There are four kinds of interface messages: event, request, query, and report. There are link-specific and bundle-specific messages for each message type. Request and event messages are loosely tied, but there is a tighter coupling between query and report. All messages are assumed to be transparently and reliably delivered by an XML-based intercomponent communication protocol.

A detailed specification of these interfaces is available [4]. The implementation of these interfaces is now publicly available as part of the DTNRG reference implementation.

III. ADAPTIVE DTN ROUTING

The routing problem in a DTN is unique in several respects. First, unlike in a mobile ad-hoc network (MANET), there

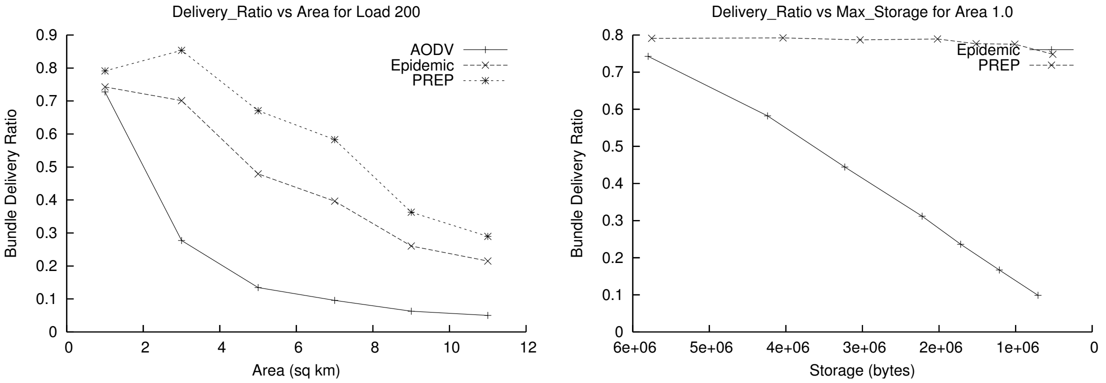


Fig. 3. (a) Bundle delivery ratio as a function of area; (b) bundle delivery ratio as a function of storage

may never be a contemporaneous end-to-end path and one has to exploit transitive contacts to get a packet delivered. Conventional MANET routing protocols typically drop packets in such situations and therefore are insufficient. Second, disconnection is often the norm rather than the exception, and therefore, controlled replication becomes much more important. Third, managing persistent storage and bandwidth-limited ephemeral contacts becomes an integral part of the routing in DTNs.

Further, real-life networks are seldom "pure" DTNs – they often exhibit varying levels of stability and disruption over space and time. Thus, it is important for routing to *adapt* to the environment and provide the best possible performance.

There has been a surge in research on DTN routing in the last few years. These range from replication-oriented strategies such as epidemic routing [6], probabilistic forwarding and purging [7], [8], [9], and future contact prediction approaches [10], [11]. A good survey of routing in DTNs is available in [12].

We are developing adaptive routing algorithms that are unique in relation to existing work: they are simple, robust, not reliant on the presence of extant patterns, and most importantly, work well in both DTNs and MANETs. Specifically, we have two algorithms called Prioritized Epidemic (PREP) and Anxiety-Prone Link State (APLS).

The key idea behind PREP is to impose a partial ordering on bundles for transmission and deletion. The drop priority of bundles is calculated as follows. We consider bundles that have a hop-count value greater than or equal to a configured threshold V_{hc}. Among these bundles, we give a lower priority to those that have a larger shortest-path cost to the destination as determined by the route computation described later. Specifically, the priority p_d(B) of a bundle B is equal to the cost of the lowest-cost path from the current node to the bundle's destination D. A lower value means a higher priority.

Transmission priority is given to bundles that are headed "downstream", that is, toward their destination. Within this, the priority $p_t(B)$ of a bundle B is equal to its ranking in a radix sort on (expiryTime(B) - currentTime) and (creationTime - currentTime). Ties are broken randomly.

Internode costs are based on a novel metric called *average availability* (AA). The AA metric attempts to measure the average fraction of time in the near future that the link will be available for use. Each link's AA is epidemically disseminated to all nodes. Path costs are computed using the topology learned through this dissemination, with cost of a link l set to $(1 - AA(l)) + c$ (a small constant factor that makes routing favor fewer number of hops when all links have AA of 1).

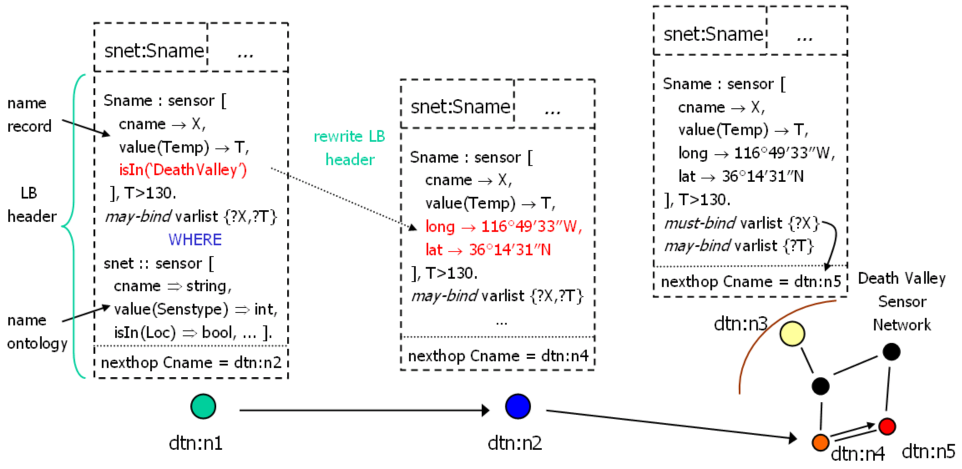
Additional details of PREP and its performance can be found in [13].

Anxiety-Prone Link State (APLS) builds upon PREP by introducing a *shortest cost path* (SCP) mode that is triggered whenever the cost to the destination is less than a configured threshold. Thus, in disrupted regions or at disrupted times, PREP-like replicated forwarding is employed while at stable times or regions, the behavior resembles conventional MANET forwarding (single copy sent over shortest cost path). APLS is highly adaptive, and expected to be competitive with protocols specifically designed for both disrupted and stable environments.

We present simulation results (using *ns-2*) comparing PREP to epidemic routing [6] and AODV [14]. Simulation parameters used were: 25-node network with random waypoint mobility model, node speeds chosen uniformly in 5-15 m/s, transmission range of 250m, data rate of 1 Mbps, area varied 1-9 square km, and load 40-200 bytes/sec.

Figure 3(a) shows the dependence of delivery ratio on the area of operation, for AODV, epidemic and PREP. Increasing the area makes the connectivity sparser and increases the level of disruption. While all three mechanisms are affected by disruption, AODV is affected the most, as expected. At a representative value of area=5 sq km, PREP's delivery ratio is 20 percentage points better than that of epidemic and 54 percentage points more than AODV. The initial rise for PREP can be attributed to the fact that at area=1 sq. km the reduced spatial reuse is more dominant.

Figure 3(b) shows the dependence of delivery ratio on storage. As storage decreases, epidemic's delivery falls approximately linearly whereas PREP's remains almost flat, with a factor of seven difference at low storage. This is due to the bundle-drop priority procedure that ensures that valuable bundles are not dropped. In contrast, epidemic drops bundles



```
snet:Sname  ...
Sname : sensor [
  cname → X,
  value(Temp) → T,
  isIn('Death Valley')
 ], T>130.
may-bind varlist {?X,?T}
         WHERE
snet :: sensor [
  cname ⇒ string,
  value(Senstype) ⇒ int,
  isIn(Loc) ⇒ bool, ... ].
nexthop Cname = dtn:n2
```

```
snet:Sname  ...
Sname : sensor [
  cname → X,
  value(Temp) → T,
  long → 116°49′33″W,
  lat → 36°14′31″N
], T>130.
may-bind varlist {?X,?T}
  ...
nexthop Cname = dtn:n4
```

```
snet:Sname  ...
Sname : sensor [
  cname → X,
  value(Temp) → T,
  long → 116°49′33″W,
  lat → 36°14′31″N
], T>130.
must-bind varlist {?X}
may-bind varlist {?T}
nexthop Cname = dtn:n5
```

Fig. 4. Progressive resolution of intentional name attributes

arbitrarily.

IV. Late Binding of Intentional Name Attributes

In traditional well-connected networks such as the wired Internet or terrestrial and cellular networks, destination names (e.g., DNS names) are typically resolved *at the source* to a canonical name in a *routable* namespace of identifiers (e.g., IP addresses). This is typically facilitated by a quasi-static hierarchy of name resolution databases (e.g., DNS hierarchy) that can be consulted by the source to perform resolution.

In networks subject to disruption, such information to map destination names to routable canonical destination identifiers may not be readily available at the source; also the nodes that may have such information may not be readily reachable from the source. For example, SPINDLE envisions the use of richly attributed names for addressing endpoints which the source may be unable to "bind" to routable canonical names. Instead, intermediate nodes in the network (or even the destination) may perform the resolution. This process of deferred name resolution is referred to as *late binding*.

The late binding module addresses the problem of progressively resolving the destination name (denoting individual, group or service endpoints) to the canonical name of a *care-of* node which either corresponds to the destination or can further the process of determining the canonical names of the destination. This process must continue until the canonical name(s) of node(s) to which the destination name is bound becomes known.

Key components of the SPINDLE late binding architecture include an expressive name scheme based on a declarative logic language, addition of a metadata extension block to the bundle protocol to carry information for name resolution, use of knowledge bases to store name management and resolution information, publish-subscribe mechanisms to exchange name management information, and name resolution procedures that are performed on DTN nodes. Deductive databases are a possible means to implement this architecture.

Figure 4 illustrates the process of progressive resolution of a multi-attributed intentional name that essentially denotes all sensor nodes in name schema "snet" that are temperature sensors located in Death Valley, California, *and* are reporting temperature above 130 F. Such names are essentially *queries* into a distributed database of name attributes that can be potentially satisfied by multiple nodes in the network.

The SPINDLE architecture allows for both the case in which intentional name resolution ([15] describes a solution for a traditional network) and bundle routing are decoupled from each other, and the case in which they go hand-in-hand. The former is interesting in scenarios involving the mere *translation* of a name from one namespace to another, not necessarily for routing; for example, a bundle with an encrypted destination name may get decrypted at a security gateway to a name known within the secure network.

Integrated resolution and routing is very useful in DTN when progressive resolution of name attributes can help in the routing process. A case in point is the `isIn(DeathValley)` attribute in the example in Figure 4. This attribute cannot be resolved at the source node `dtn:n1` but the bundle is forwarded to an intermediate node `dtn:n2` which possesses GIS databases and hence can resolve this attribute to the latitude/longitude coordinates for Death Valley. This new information is written into the metadata extension block in the bundle. When the bundle is routed to a node `dtn:n4` belonging to the Death Valley sensor network, the canonical name in the bundle is bound to `dtn:n5`, which is reporting high temperature above 130 F, but the exact value of the temperature can be known only after the bundle reaches the latter.

V. Disruption-Tolerant Access to Content

Currently information access on the Internet is based primarily on the location of content. For example, a user needs to provide a universal resource locator (URL), which includes a DNS name that must be resolved at the source.

Two key technologies significantly enhance our ability to efficiently access information on the Internet. The first is the indexing and search infrastructure that enables one to access information by collecting and maintaining *ex post facto* mappings of content to location. The second key technology is a caching infrastructure that maintains a mapping from content location to a cache location. Caching can reduce end-user latency and also alleviate the load on servers and access networks.

These two technologies have completely transformed the way we access information over the Internet. They are, however, based on assumptions of strong connectivity to the Internet, which means they are not tolerant to disruption. Content-based access is possible only when access to the search infrastructure is available. The consistent-caching algorithms that are the basis of extensive caching infrastructure on the Internet assume strong connectivity to the caching hierarchy.

Suppose we want a map of Baghdad and a neighboring node (within a mobile ad-hoc network cluster) has cached the map of Baghdad from a particular map provider. In the current approach, there is no easy way to discover and take advantage of this fact. We need to first connect to a search engine, identify one or more locators that matches “map of Baghdad,” then traverse a hierarchy of caches until we can successfully connect to a cache that has the content corresponding to one of the locators. Any of these steps could be easily disrupted.

In order to support disruption-tolerant access of information based on content, we are developing approaches for opportunistic caching, indexing, and retrieval of data in a distributed fashion in a manner that maximizes the availability of information required for the mission even when disconnected from the Internet (when access to a global caching infrastructure or search engine is unavailable).

With our approach, users simply describe what they want, not where it is stored, and the network moves information when and where it is needed. We provide a disruption-tolerant publish-subscribe system that supports push, pull, and third-party push metaphors. Content flow and caching self-organizes around supply (who provides what) and demand (who needs what); supply and demand are disseminated within the DTN. Through a rule-based framework, we allow content-flow policies to be specified in a flexible manner, and through an ontology-based approach, we allow for rich descriptions of content, services, and endpoints. In conjunction, we are developing an application that allows users to access information under disruption through a familiar web-browser facade.

VI. A Declarative Knowledge-based Approach

The SPINDLE system makes use of a declarative knowledge-based approach based on frame logic[16], [17] to integrate routing strategy selection, intentional naming, policy-based resource management, and content-based access to information. The knowledge base facilitates the declaration of facts, rules, and queries, thus capturing system and policy knowledge and enabling decision making on network nodes. A similar approach has been described by Loo et al.[18].

Other modules of the decision plane can query the knowledge base using a declarative query language. The knowledge base also allows us to define rules in order to derive complex facts from simple ones. For example, a predicted future adjacency from a ground node to a UAV node can be represented as a first order F-logic rule as follows:

```
predictedAdjacency :: spindleAdjacency.
S:predictedAdjacency [fromNode -> X,
    toNode -> U, adjType -> ’PREDICTED’,
    adjUpAt -> T1, adjDownAt -> T2 ] :-
  walltime (Tnow),
  U[trajectory -> Tr1], X[trajectory -> Tr2],
  trajectory_xing(Tnow, Tr1, Tr2, [T1, T2]), !.
```

Our knowledge base can thus expose such facts (both simple and complex) to the decision plane using a uniform interface. Rules are executed internally to retrieve derived facts. For example, a routing algorithm could make use of derived predicted adjacencies as well as other explicitly asserted adjacencies in its computations.

The knowledge base is also used to support rules for content management, content and endpoint ontologies to express supply and demand information, ontologies for descriptive naming of endpoints, and policies as well. In our approach, each piece of content will be tagged with metadata describing it. Such metadata can be represented as a frame which may be a conjunction of multiple attribute-value pairs and can then be inserted as a fact in a persistent KB. For instance, a map of a certain portion of Baghdad can be described using:

```
m1:map [city -> ’Baghdad’, country -> ’Iraq’,
  minLat -> ’33:14:20:N’, maxLat -> ’33:14:50:N’,
  minLon -> ’44:22:10:E’, maxLon -> ’44:22:30:E’,
  creat -> 200601011200, modif -> 200603011500,
  originator -> ’dtn://n0’, digSign -> S1234,
  cachedAt -> {’dtn://n1’, ’dtn://n3’},
  sha1Check -> 23412456, blobKey -> ’m1’].
```

User demand for content is fulfilled through a combination of query propagation and sharing of content metadata. Geographical queries such as “give me maps of the areas of Baghdad that are residential” can be supported by this architecture if appropriate rules are written to resolve those queries and pertinent GIS data is available.

VII. Testbeds and Evaluation

It is vital for the community to gain operational experience with the technology at this stage. Therefore, we are planning to evaluate the SPINDLE system using several approaches.

First, we are constructing an in-house testbed at BBN called ElevatorNet, which includes static and mobile nodes (in elevators) to form a test network prone to disruptions. Second, we are collaborating with the DieselNet project at UMass-Amherst in order to gain operational experience with the SPINDLE system running on their bus-based network [11]. Third, a valuable by-product of our research effort is an evaluation platform called MINAS (see Figure 5), which we use to evaluate DTN system software over an emulated DTN [3]. MINAS integrates OS virtualization based on User-Mode Linux, virtual Ethernet bridging and firewalling, and network emulation based on ns-2 (with some modifications) to provide a powerful and flexible system evaluation capability

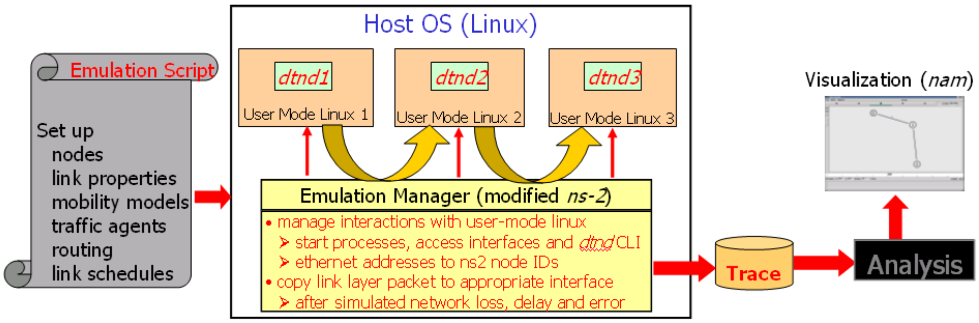

Host OS (Linux)
Emulation Script
Set up
nodes
link properties
mobility models
traffic agents
routing
link schedules
dtnd1
User Mode Linux 1
dtnd2
User Mode Linux 2
dtnd3
User Mode Linux 3
Emulation Manager (modified ns-2)
• manage interactions with user-mode linux
➢ start processes, access interfaces and dtnd CLI
➢ ethernet addresses to ns2 node IDs
• copy link layer packet to appropriate interface
➢ after simulated network loss, delay and error
Trace
Analysis
Visualization (nam)

Fig. 5. Components of the MINAS Environment

for continued system development and testing. Finally, a demonstration is planned at a military facility.

VIII. Summary

In this paper we described a disruption-tolerant networking system and related DTN technology being developed within the SPINDLE project at BBN. Key contributions of the project are:

- A modular architecture for a DTN system, in which the core technology (based on open DTNRG standards and software) is refined and engineered in a commercial setting, overlaid on available routers and links, while enabling DoD-specific value-added plugins to be developed independently and then easily integrated.
- A robust deployable system prototype to be demonstrated in a military-relevant setting.
- Distributed caching, indexing, and retrieval for disruption-tolerant content-based access to information.
- Development of disruption-tolerant routing strategies that outperform traditional approaches to route computation and network-state dissemination.
- An architecture for policy-based resource management in DTNs, including a machine-understandable language to express DTN policy, and an approach to process policy within DTN nodes.
- A name-management architecture for DTNs that supports progressive resolution of intentional name attributes within the network (not at the source), including support for "queries as names" and name-scheme translation.
- A declarative knowledge-based approach that integrates routing, intentional naming, policy-based resource management, and content-based access to information.
- Research infrastructure – a flexible emulation platform and a testbed – to evaluate DTN systems and applications.

DTN can transform military communications in a profound way, with potential applications spanning a wide spectrum of DoD communities including tactical (situational awareness), strategic operations (intelligent content flow management), special operations, and intelligence (electronic drop boxes and data mining). Significant progress is being made in this area.

Acknowledgments

The authors would like to thank the participants of the DTNRG, MITRE, and other participants of the DARPA DTN program for their collaboration on the system architecture and interface specifications. We specifically thank Daria Antonova, Alex M. Colvin, Talib S. Hussain, Carl Livadas, Mitchell Tasman, David J. Montana, David R. Moran, and Partha P. Pal for their contributions.

References

[1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant network architecture," April 2007, RFC 4838.

[2] K. Scott and S. Burleigh, "Bundle protocol specification," April 2007, Internet-Draft.

[3] R. Krishnan. SPINDLE (Phase 1) final report. BBN Technologies. [Online]. Available: http://www.ir.bbn.com/projects/spindle

[4] "Draft SPINDLE architecture," BBN Technologies, 2006.

[5] S. Symington, S. Farrell, H. Weiss, and P. Lovell, "Bundle security protocol specification," April 2007, Internet-Draft.

[6] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep. CS-200006, April 2000.

[7] K. Harras, K. Almeroth, and E. Belding-Royer, "Delay tolerant mobile networks (DTMNs): Controlled flooding schemes in sparse mobile networks," 2005.

[8] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected wireless networks," in *Proc. ACM Sigcomm Workshop on DTN*, 2005.

[9] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using redundancy to cope with failures in a delay tolerant network," in *ACM SIGCOMM*, 2005.

[10] A. Lindgren, A. Doria, and O. Scheln, "Probabilistic routing in intermittently connected networks," in *Proc. ACM Mobihoc*, 2003.

[11] B. Burns, O. Brock, and B. Levine, "MV routing and capacity building in disruption tolerant networks," in *Proc. IEEE Infocom*, August 2005.

[12] Z. Zhang, "Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: Overview and challenges," *IEEE Communication Surveys and Tutorials*, Jan 2006.

[13] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *MobiOpp'07 (to appear)*, June 2007.

[14] C. E. Perkins, E. M. Belding-Royer, and S. Das, "Ad hoc on demand distance vector (AODV) routing," RFC 3561.

[15] W. Adjie-Winoto, E. Schwartz, H. Balakrishnan, and J. Lilley, "The design and implementation of an intentional naming system," in *ACM SOSP*, 1999.

[16] M. Kifer, G. Lausen, and J. Wu, "Logical foundations of object-oriented and frame-based languages," *Journal of ACM*, May 1995.

[17] G. Yang, M. Kifer, C. Zhao, and V. Chowdhary. Flora-2: User's manual. [Online]. Available: http://flora.sourceforge.net

[18] B. T. Loo, J. M. Hellerstein, I. Stoica, and R. Ramakrishnan, "Declarative routing: Extensible routing with declarative queries," in *ACM SIGCOMM*, Philadelphia, PA, August 2005.