Supporting Predicate Routing in DTN over MANET[*]

Gabriele Ferrari Aggradi Flavio Esposito Ibrahim Matta
Computer Science, Boston University
{gfa, flavio, matta}@cs.bu.edu

Technical Report BUCS-TR-2008-013

ABSTRACT

We consider a Delay Tolerant Network (DTN) whose users (nodes) are connected by an underlying Mobile Ad hoc Network (MANET) substrate. Users can declaratively express high-level policy constraints on how "content" should be routed. For example, content may be diverted through an intermediary DTN node for the purposes of preprocessing, authentication, etc. To support such capability, we implement Predicate Routing [7] where high-level constraints of DTN nodes are mapped into low-level routing predicates at the MANET level. Our testbed uses a Linux system architecture and leverages *User Mode Linux* [2] to emulate every node running a DTN Reference Implementation code [5]. In our initial prototype, we use the On Demand Distance Vector (AODV) MANET routing protocol. We use the network simulator *ns-2* (ns-emulation version) to simulate the mobility and wireless connectivity of both DTN and MANET nodes. We show preliminary throughput results showing the efficient and correct operation of propagating routing predicates, and as a side effect, the performance benefit of content re-routing that dynamically (on-demand) breaks the underlying end-to-end TCP connection into shorter-length TCP connections.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Design, Experimentation, Performance, Reliability, Security

Keywords

Delay Tolerant Network, Mobile Ad hoc Network, Emulation, Routing, Predicate Routing, Transmission Control Protocol

1. INTRODUCTION

Delay Tolerant Network (DTN) architectures have emerged to allow content-based networking in the presence of potentially unreliable or high-delay communication links/paths [1]. A communication substrate that is plagued with unreliability and path disruptions is that of a Mobile Ad hoc Network (MANET). We consider DTN users (nodes) communicating over a MANET network. We augment such DTN-over-MANET architecture with the ability of DTN users to express high-level requirements on the routing of content. For example, for content from node S and destined to DTN user (node) D, an additional constraint might be injected into the system to re-direct such data through an intermediary DTN node I for functions such as content pre-processing, authentication, etc. This content re-direction is done on demand by propagating low-level predicates to affect routing at the MANET level.

Our Contributions: We integrate two different network architectures — the content-aware DTN overlay and the underlying (often resource-constrained) MANET substrate — as follows:

- We implement a reliable DTN neighbor discovery mechanism that leverages AODV's HELLO messages to propagate DTN node names. The convergence layer of the DTN stack then maintains the mappings from DTN node names to IP (MANET) node addresses.
- In addition to DTN node names, AODV's HELLO messages are also used to propagate low-level MANET routing predicates. These latter predicates are mapped by the convergence layer from given DTN-level requirements on routing content.
- To demonstrate our implementation, we realized a testbed depicted in Figure 1. Our base system is a host running an *Ubuntu* distribution of Linux (version 7.10). User Mode Linux (UML) [2] is used to create virtual machines, which are connected with "tap" interfaces [3]. Each virtual machine hosts one emulated node with an AODV routing [4] daemon, as well as a DTN reference implementation [5]. We use the *ns-2* simulator (emulation version [6]) to simulate mobility and wireless connectivity of nodes, including MANET-only nodes. An additional bridged network is setup for configuring and controlling our experiments.
- We present preliminary throughput results showing the

[*]This work has been partially supported by a number of National Science Foundation grants, including CISE/CCF Award #0820138 CISE/CSR Award #0720604, CISE/CNS Award #0524477, CNS/ITR Award #0205294, and CISE/EIA RI Award #0202067.

CHANTS'08, September 15, 2008, San Francisco, California, USA.

efficient and correct operation of propagating routing predicates, and as a side effect, the performance benefit of content re-routing that dynamically (on-demand) breaks the underlying end-to-end TCP connection into shorter-length TCP connections.

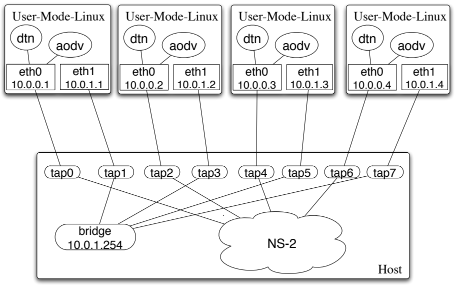

Figure 1: Testbed Architecture.

2. PREDICATE ROUTING

Quoting Roscoe *et al.* [7], predicate routing defines *"the state of the network declaratively as a set of boolean expressions associated with links which assert which kind of packet can appear where"*. From a user point of view, a predicate is a high-level constraint on the routing of content, *e.g.* direct all images captured by the camera on node S to DTN node I for pre-processing and authorization before sending them to the DTN user at node D.

From the network point of view, the predicate is a set of rules that each MANET packet, or DTN bundle, has to fulfill. Table 1 shows an example of predicate routing where the predicate directs all data to DTN node I for pre-screening: the first rule directs all data destined to node D but not yet pre-screened at node I to node I first for pre-screening; and the second rule forwards pre-screened traffic directly to node D.

Our system maps declarative user policies to such network-level routing predicates. We next show how routing predicates get propagated and installed as MANET-level forwarding rules.

Predicate	Next Hop
src $= \neg I \wedge$ dest$= D$	to I
src $= I \wedge$ dest$= D$	to D

Table 1: Direct all D-traffic to an intermediate DTN node I.

3. DESIGN AND IMPLEMENTATION

3.1 Overall Architecture

Figure 2 shows the DTN-MANET stack—our modified and added components are marked by "stars".

The Application Programming Interface (API) of the DTN reference implementation is extended to allow an application to inject high-level constraints on content routing, *i.e.* predicates that involve DTN node names. We refer to this modification as *Predicate Routing API* (*PR-API*).

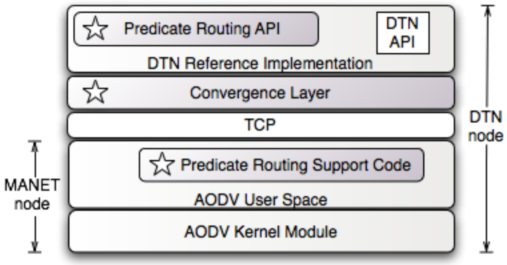

Figure 2: Architecture of a DTN node running over a MANET substrate. The node is Predicate Routing (PR) enabled.

The *Predicate Routing Support Code* (*PRSC*) component, implemented in the AODV user space, mainly implements two functionalities: (1) it uses the *iptables* Linux facility [8] to install predicate MANET routing rules, so that MANET packets carrying content/DTN bundle(s) are routed based on high-level routing constraints; and (2) it creates and manages new AODV extensions to discover other DTN nodes and propagate MANET routing predicates.

The *Convergence Layer* (*CL*) interfaces DTN and MANET by maintaining the mapping between DTN node names and IP/MANET addresses. The mappings are used to translate routing predicates on DTN node names to routing predicates on corresponding IP/MANET node addresses.

3.2 Extending On Demand Distance Vector

Our predicate routing implementation requires two extensions to the current AODV protocol (RFC 3561 [4]): (1) extension #29 for discovering DTN-capable nodes (DTN neighbor discovery), already introduced by Ott *et al.* in [9]; and (2) a new extension we introduce for propagating MANET routing predicates.

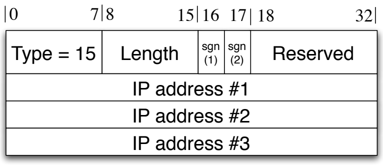

Figure 3: Predicate routing packet extension.

Figure 3 shows our initial AODV packet extension, which allows a routing predicate of the form $S \wedge D \Rightarrow I$, where any of the two boolean variables S and D could be negated. The fields that follow the standard Type and Length, are used to express and build a routing predicate: the two sign bits, $sgn(\cdot)$, represent the existence or lack thereof of the boolean logic negation, and the three IP (MANET) addresses represent the source and destination of the MANET packet satisfying the predicate, and the next-hop that should be used if the predicate is satisfied.

Ott *et al.* [9] use extension #29 in the route-request RREQ and route-reply RREP packets of AODV. Thus, DTN-capable nodes are only discovered as AODV attempts to find routes to destinations on demand. This approach may fail to discover DTN-capable nodes or may discover them in a non-timely fashion. This limitation is more serious or unacceptable when one wants to propagate routing predicates to exert control over the underlying MANET routing. In this case, timely dissemination of predicates to all MANET nodes for consistent/reliable routing is crucial. To this end, we include both AODV extensions in the AODV HELLO message, which is periodically advertised (every 1 second by default). Whenever a node comes in contact with another, they exchange in their HELLO messages their knowledge of their mappings of DTN names to IP (MANET) addresses, as well as MANET routing predicates.

3.3 Predicate Propagation

We give a concrete example to illustrate how a routing predicate is propagated, installed and used. Consider a management application injecting a routing predicate that directs all DTN bundles from DTN node S to an authenticator DTN node I before being routed to destination DTN node D. This is done using the PR-API.

Initially, every DTN node will advertise its presence on the MANET network. This is done by attaching its DTN node name to the (periodic) HELLO message (using extension# 29). The mapping between DTN name and IP (MANET) address (e.g. $dtn : //nodeI.dtn \iff 10.0.0.2$) is learned by neighbor nodes (and maintained by their Convergence Layer), which in turn advertise their mappings to their neighbors, and so on.

The PR-API creates the routing predicate after mapping the DTN names to IP (MANET) addresses by consulting the CL. Then, the PRSC component creates the new predicate extension attaching the MANET-level routing predicate to the outgoing HELLO message. Upon receiving a MANET routing predicate, a MANET node installs that predicate using the iptables Linux facility.

Once the routing predicate, “src $= S \wedge$ dest $= D \rightarrow$ to I”, has been propagated through the network, whenever node S sends a bundle to node D, the MANET packets carrying the bundle are first directed to node I. The authentication application running at node I listens to DTN bundles in promiscuous mode. Once a bundle is received, it gets processed and authenticated by the authentication application. Meanwhile, the authentication application at node I installs a local DTN-specific predicate to drop the original copy of the bundle from node S—this capability is implemented by mainly modifying the *should_fwd* forwarding method in the DTN reference implementation. After authentication is done, the bundle is then forwarded to the original destination at node D, which is realized by the installed MANET routing predicate “src $= I \wedge$ dest $= D \rightarrow$ to D”. Note that a bundle sent from node S to node D through node I is reliably transmitted over two separate underlying TCP connections.

4. PERFORMANCE RESULTS

We continue with the use case in the previous section, of re-routing DTN bundles through an intermediary DTN node. We consider the static topology in Figure 4. A link between two nodes means that the nodes are within communication range. The MAC protocol is 802.11, and link rates are 1Mbps.

Figure 5 shows the amount of data received (including duplicates) over time when DTN node 6 sends a 1MB data bundle to DTN node 2 with and without going through DTN node 4. Each point in the plot was obtained by averaging five independent runs. The plot shows a delay in receiving data at the destination when going through the intermediary authenticator as the bundle gets reassembled from the MANET packets and gets processed. However, the 1MB-data is ultimately delivered earlier at the destination because data gets transported over two separate shorter-length TCP connections, which perform better in terms of both throughput and goodput especially over lossy wireless hops.

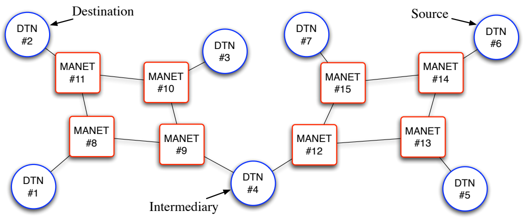

Figure 4: Validation scenario.

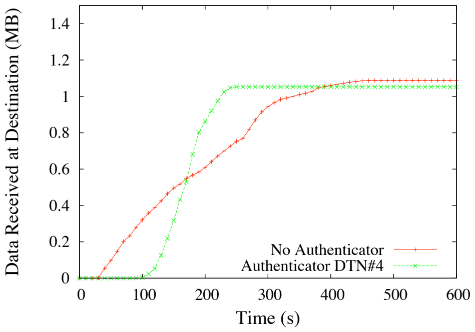

Figure 5: Data delivered at the destination vs. time for 2% packet loss probability.

5. REFERENCES

[1] K. Fall, “A Delay-Tolerant Network Architecture for Challenged Internets,” in *SIGCOMM ’03: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications.* ACM, 2003, pp. 27–34.

[2] J. Dike, *User Mode Linux*, ser. Bruce Perens’ Open Source Series. Prentice Hall, 2006.

[3] http://vtun.sourceforge.net/tun/.

[4] C. Perkins, E. Belding-Royer, and S. Das, “Ad Hoc On-Demand Distance Vector (AODV) Routing,” RFC 3561, Internet Engineering Task Force (IETF), July 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3561.txt

[5] [Online]. Available: http://www.dtnrg.org/wiki/Code

[6] http://www-ivs.cs.uni-magdeburg.de/EuK/forschung/projekte/nse/.

[7] T. Roscoe, S. Hand, R. Isaacs, R. Mortier, and P. Jardetzky, “Predicate Routing: Enabling Controlled Networking,” *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 65–70, 2003.

[8] Iptables. http://www.netfilter.org/.

[9] J. Ott, D. Kutscher, and C. Dwertmann, “Integrating DTN and MANET Routing,” in *CHANTS ’06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks.* New York, NY, USA: ACM, 2006, pp. 221–228.