Deliverable FI3-D19
Report on Test Deployment of a Delay Tolerant Network in the Kemi Mine

Mikael Andersson (Aalto), Teemu Kärkkäinen (Aalto), Aki Ruotsalainen (Cybercube OY), Philip Ginzboorg (Nokia), Jörg Ott (Aalto)

(mikael.andersson@aalto.fi, teemuk@netlab.tkk.fi, aki.ruotsalainen@cybercube.fi, philip.ginzboorg@nokia.fi, jo@netlab.tkk.fi)

Tivit Future Internet Program
(Tivit FI)

Period: 1.4.2012 – 30.4.2012

Tivit, Strategisen huippuosaamisen keskittymän tutkimusohjelma

Rahoituspäätös 1171/10, 30.12.2010, Dnro 2790/31/2010

www.futureinternet.fi

www.tivit.fi

This work was supported by TEKES as part of the Future Internet programme of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT).

Executive summary / Internal release

Content: This deliverable documents the final testing of the DTN router implementation performed at the Outokumpu Kemi mine between 20.4. and 28.4.2012. The test setup includes multiple nodes installed in a control room, intermediate carriers and mining equipment. The tests are run in a production environment and include a full end-to-end network setup. This is augmented with measurements taken over a two-week period from a smaller set of devices in the same environment.

The full test setup including software, hardware and equipment is documented and described. Collected data is recorded and analyzed and conclusions based on the analysis are presented in this document. We also conducted simulation tests with real contact patterns and synthetic traffic generation. Lessons learned and possible future development directions are also described.

1 Table of Contents

Deliverable

2 Test Description

We have installed a test setup in Outokumpu's chromium mine in Kemi, Finland. The setup consists of four nodes running our own implementation of a DTN Forwarder and a file transfer solution. Two of the nodes are in mining machines. Such machines are often outside of WLAN coverage while doing work in the mine shafts. One node is installed in a pickup truck used to transport personnel around the mine, and one node is installed as a stationary server in an office / control room.

The mining machines nodes act as message sources, generating periodic data to be delivered to the control room server (message sink), either directly or with the help of the pickup node (carrier).

The Outokumpu Kemi mine is a chromium ore mine first opened in 1966. It was initially mined as an open pit mine, but underground mining was started in 1999 with a current designed capacity of 2.7 Mt/y of ore. It currently extends to a depth of 500 meters below ground level.

The mine is worked in three shifts around the clock, including weekends. Personnel are transported around the mine in Toyota Hilux pickup trucks, and there are some several diesel powered mining vehicles such as drilling rigs and bolters.

The Kemi mine has got an unusually well-developed IT infrastructure, with a wireless LAN reaching many parts of the mine and full WLAN coverage in offices and crew facilities. WLAN base stations are also located throughout the working areas of the mine, mostly along the main tunnels. Each production level has approximately 20 WiFi access points connected to the mine backbone.

Despite the broadly spread wireless network, WLAN coverage is still often not available in remote tunnels, and the mining machines often stay out of WLAN coverage for hours to days while stationed at the work site. The WLAN radio waves cannot propagate through the bedrock, so a line-of-sight to the access point is required. Thus, uploading of data from the mining machines has to wait until the machine is moved out of the remote work area and back into WLAN coverage.

Expanding the wireless network to cover all the mining shafts would be impractical. Many more access points would be needed, and new tunnels to cover are constantly being dug. Many tunnels are also frequently blasted, with a risk of destroying any permanently installed networking infrastructure. The solution idea for these issues has been described in [1].

2.1 Technical Setup

A file transfer solution running on top of the DTN Bundle Protocol [2] has been developed and deployed in the mine. By taking advantage of mobile nodes in the mine, DTN can be used to extend the network coverage to include areas outside normal WLAN coverage. The pickup trucks used by the personnel to move around in the mine periodically visit many of the working sites around the mine, often situated outside WLAN coverage, and is also frequently in areas covered by the WLAN network while moving between work sites. The trucks can thus be used as data carriers to physically move messages between mining machines outside WLAN coverage and the wireless network.

2.2 DTN Router

The software implementation consists of three components: 1) a DTN Forwarder that provides store-carry-forward style routing based on the Bundle Protocol and the TCP Convergence Layer, 2) a file transfer application that uses the DTN bundle delivery service of the DTN Forwarder to transfer files across the network, and 3) an example client application that generates dummy data in the form of XML-files describing work progress in the mining machines. The DTN Forwarder and the file transfer utility have been implemented at Aalto University, using Java Standard Edition 1.5, and the client application has been implemented by the partner company Cybercube Oy using C# for Microsoft .net.

The DTN Forwarder is as a store-carry-forward router that can discover peers within the same local area network, open links to the discovered peers, transfer bundles over the links and store bundles while waiting for new contacts. Bundles and router state are stored persistently on disk (or in this case, CompactFlash memory card) in order to survive reboots.

The forwarder incorporates a Java-implementation of the TCP Convergence Layer (TCPCL) previously done at Aalto. TCPCL works as an interface between the DTN forwarder and the lower TCP layer, and provides services such as establishing contacts and sending bundles point-to-point.

The DTN Forwarder performs epidemic routing, so that carried bundles are forwarder to all encountered nodes, unless the bundle has been previously received from or sent to that node, and unless the bundle originates from that node. The receiving node discards incoming bundles that have already been received. A bundle is deleted from the Forwarder whenever its time to live expires, acknowledgement has been received, or if the router runs out of space and has do discard bundles in order to be able to receive new ones. In the test setup the time-to-live field in all bundles have been set to 24 hours.

To discover other nodes in the neighborhood, the DTN Forwarder uses an implementation of the DTN IP Neighbor Discovery (IPND) mechanism. Beacon packets identifying the node are sent over UDP broadcast every three seconds. If no beacons are received from a neighbor within 20 seconds, it is reported as lost. Connections to peer nodes are, however, in practice kept open as long as the TCP timeout allows them to, because this gives a more stable behavior and avoids the frequent connections and disconnections that would otherwise result when nodes are near the edges of radio contact. The discovery mechanism can also be configured use unicast or multicast, and will automatically start using any network interface that appears on the host computer, unless configured otherwise.

By default, the discovery mechanism will automatically start using any network interface that appears on the host computer and send beacon packets addressed to the local IPv4 broadcast addresses of those interfaces. It can also be configured to use unicast or multicast, or to only bind to specific interfaces. In the test setup all the mobile nodes, i.e. the mining machine tablets and the router node in the pickup truck, are configured to run two discovery instances, one broadcasting and listening on all interfaces and the other sending targeted unicast beacons addressed to the stationary server node in the control room. The reason for this is that the server node is located in a different IPv4 subnet and would not be able to receive the broadcast beacons. This also means that the server will be the one opening the TCP connections towards the other nodes when they are reachable, based on the beacons it receives. During encounters between the pickup truck node and the mining machine nodes, both nodes will normally receive each other's beacons and the node with the alphabetically lower EID (Endpoint Identifier) will be the one initiating the TCP handshake.

On top of the DTN Forwarder is also a simple file transfer application. It provides a generic file transfer solution using the Bundle transfer service of the DTN Forwarder, and implements reliability by using retransmissions and acknowledgements. Files are identified based on name, time modified and a CRC32 checksum, and are acknowledged by the destination node on reception. If the original sender node fails to receive an acknowledgement within a timeout, the file is retransmitted. Subsequent retransmissions occur whenever the timeout time passes again. In the test setup the retransmission time has been set to 24 hours.

The interface of the file transfer consists of a directory in the file system that is being watched and checked for new files every 30 seconds. When new files are found in the directory they are sent, and when they are acknowledged by the recipient they are deleted from the directory. If files are placed in subdirectories to the watched send directory, they are sent to EID:s based on the name of the subdirectories, otherwise the destination EID is provided by a configuration file. In the test setup all nodes are configured to send files to the server node.

When a node receives a file, the file is stored to the file system in a "receive" directory specified in the configuration file. If multiple (distinct) files with the same name are received at a node, a timestamp is added to the file name to avoid collisions. The server node stores received files in a directory on its CompactFlash memory card. All the nodes in the test setup run a complete instance of the DTN Router with the file transfer utility.

The DTN software generates log files that are written to the local disk (or CompactFlash card) in a rotating set. A copy of each completed log is also compressed with the ZIP compression scheme and put in the send directory, so that all the log files from all nodes are sent to the server node using the DTN file transfer solution. From the server node the log files can then be retrieved and inspected, along with other files sent over the DTN network. The nodes have been configured to log fairly detailed data, and will start a new log file whenever the current uncompressed log file reaches a length of 200000.

2.3 Devices

The tests included two types of devices, 1) embedded Linux router node, and 2) mining machine tablets.

2.3.1 Embedded Linux Router Nodes

The mobile node in the pickup truck, as well as the server in the control room, are rugged, embedded computers running Linux. The hardware setup is based on a special assignment work previously done at TKK, and consists of a commercially available main board, the PC Engines ALIX.3D3, with an embedded AMD Geode LX800 processor and 256 MB of DDR RAM. The AMD Geode is a low power x86 processor for embedded computers, running at a clock speed of 500 MHz. A 16 GB CompactFlash (CF) memory card is used for storage. Two MiniPCI 802.11bg WLAN interfaces of type Compex WLM54G are attached to the main board. They have a transmission power of 200 mW, and are connected to external antennas.

The computer board is installed in a rugged, metallic case for outdoor use, with an outdoor dipole antenna for each of the two WLAN interfaces attached on the outside of the case. The power usage of the machine is about 5W.

These embedded computers run the Debian 6.0 Squeeze GNU/Linux distribution with the Linux 2.6.32-5-486 kernel. The HotSpot Java virtual machine provided by Oracle is installed to run our DTN forwarder software written in Java. The CompactFlash system drive is partitioned as a single drive, and uses the ext3 file system.

The DTN Router and File Transfer application is run at system startup by a startup script in /etc/init.d, and guarded by a watchdog shell script that restarts the software if it shuts down, unless one closes it intentionally with the shutdown script. The Linux screen utility is used to keep the DTN software running in the background. It takes around 30 seconds for system to boot up and the DTN software to start running after a system power-on.

The two WLAN interfaces are configured so that one of the interfaces is set to connect to the mine's infrastructure WLAN (SSID: COMMON), while the other interface acts as a wireless access point (broadcasted SSID: MiningDTN) for the mining equipment to connect to. Both of the interfaces have statically configured IPv4 addresses, with the first interface configured according to the IP address range of the mine's WLAN network and the access point interface configured with a private IP address from the 192.168/16 range.

The WLAN networking abilities are provided by the wireless-tools package from the standard debian software repository. In order to run one of the interfaces in access point mode, the hostapd debian package has also been installed. Access to WPA-protected networks is provided by the package wpasupplicant, but this is not required in this test setup. The base network configuration resides in /etc/network/interfaces, while the software access point configuration for hostapd is in /etc/hostapd/hostapd.conf.

The plain wireless network tools in this Linux installation do not provide reliable reconnection to lost wireless networks in case the connection is lost and the network again becomes available. This has been solved with a very simple BASH script that periodically (every five seconds) checks the connection state of the wireless interface that has been set to connect to the infrastructure WLAN. If the interface is disconnected, it is restarted with the ifdown and ifup commands. Thisforces the interface to re-read its configuration file and attempt to connect to the network specified in the configuration. Changes in the network connection state are logged to the system log using the logger utility.

The system clocks of the nodes in the DTN network should be approximately correct and synchronized, preferably within a couple of minutes. This facilitates analysis of log files and network performance, and since the DTN Bundle protocol assigns a time to live to each bundle, a severely misconfigured clock could lead to premature deletion of bundles or retention of bundles whose expiry time is long overdue. The system board of the DTN router node is equipped with a battery to keep the clock running while it is not connected to a power source. The synchronization of the system clock has been done manually, and synchronization was performed on 19.4.2012 shortly before the most recent test run.

2.3.2 Mining Machine Tablets

Most or all of the mining machines in the Outokumpu Kemi Mine are equipped with a tablet computer, mounted by the control panel inside the cab. Two of these tablets are now also running our DTN software, working as DTN routers and providing a DTN file transfer service. The computers are of type Advantech MARS-3100R, equipped with 1 GB RAM, 40GB HDD and Intel Core Duo 1.2 GHz processor. They run Microsoft Windows 7.

The tablets have got two WLAN interfaces. A built-in WLAN interface, configured to connect to the fixed wireless network in the mine, as well as an additional USB WLAN interface configured to connect to the wireless network provided by the DTN Router boxes in the trucks. The USB WLAN adapter is significantly more powerful than the build in interface, transmitting at a power of 1000 mW versus 200 mW, and has an external, dedicated dipole antenna providing better reception.

The build-in WLAN radio in the tablet proved to be problematic, with weak reception and transmission, assumingly for the most part due to a sub-optimal antenna built in to the computer case. With the wireless signal being attenuated by both the thick steel chassis of the mining machine and the cab of the pickup truck, the transmission quality between the mining machine tablet and the DTN router box in the truck was reduced to such a degree that the pickup had to move within a couple of meters of the mining machine to get a reliable connection. After we added the external USB WLAN interface and dedicated antenna, the connection between the pickup truck and the mining machine is possible at distances up to some tens of meters. Having separate interfaces for the infrastructure WLAN and the DTN WLAN also simplifies configuration.

2.4 Test Setup

The DTN Router and file transfer software has been installed in four nodes in the mine. Two of them are embedded Linux router boxes, one installed as a stationary server node in an office on the ground level and permanently connected to the infrastructure network (Server), and the other one mounted inside a Toyota Hilux pickup truck used by the foreman when wisiting many of the work sites (Router). The node in the pickup truck is powered from the standard 12 volts ‘cigarrette lighter’ power outlet, while the server node is plugged to the mains power outlet.

The remaining two nodes are tablet computers mounted in mining machines. Tablet1 is installed in an Atlas Copco Boomer E2 C tunneling and mining rig, and Tablet2 in an Atlas Copco Cabletec LC cable bolter.

2.5 Data Collection

A contiguous set of log files from all nodes has been collected from a time period of seven days, from 20.4.2012 00:00 until 27.4.2012 23:59. The test setup has been operational for a longer time, but the data used in the analysis had to be truncated due to the node in the pickup truck being inadvertently disconnected from its power source for an extended period of time.

Each DTN Router instance generated log files containing time stamped descriptions of events like the discovery of new files to send, creation of bundles, sending and reception of bundles, reception and acknowledgement of files, available network interfaces, discovery and disappearance of peers as well as the state of TCP connections to peers. Also any discovered error situations are reported. The log files were sent to the server node using the DTN transport. Each log file was about 200000 bytes long, and sent zipped at a size of about 12000 bytes.

In addition to the log files, the example user application in each of the mining machine tablet computers generated a XML file of 6 kB every half hour, consisting of fictitious but realistically formatted data describing work processes. The files were put in "send" directory of the DTN file transfer utility and sent to the server node.

During this time period 212 log files were collected, with a total of about 210000 rows. The amount of log messages generated by a node depends on circumstances such as how many encounters with peers it experiences, so that the server generated 51 log files, the router node in the pickup 88, and the mining machine tablets 21 and 44.

3 Evaluation

In this section we present analysis and evaluation of the data collected from the Kemi test. All analysis is based on the data collected during the period of 20.4.2012 to 28.4.2012, during which all the installed devices were active and in use.

We first describe the contact patterns resulting from node movement, then analyze observed message delivery from the drilling machines to the server, and finally present a simulation study based on the measured connectivity which allows us to compare message delivery when the DTN-based system is used and when it is not used.

3.1 Contacts

Contacts are the fundamental enabler for communications between nodes in the mine. Contacts occur when two nodes come within a radio range, open a transport link and exchange messages. The duration and volume of a contact is a function of radio powers, antenna gains, interference, reflections, etc. These can be modeled in various theoretical ways, but the measurements taken during the experiment allow us to directly calculate the contacts.

The physical mobility patterns of the nodes participating in our tests can be summarized as follows. During the nine days of the test the machine carrying Tablet 1 was in maintentace. (The rear-end of that machine got smashed.) Tablet 1 was frequently powered down; often for a prolonged periods of time. When Tablet 1 was powered on, it was usually in the range of the WLAN network of the mine.

The machine carrying Tablet 2 was doing production work in the tunnel during the nine days of the trial. It was never in direct contact with the WLAN network of the mine during these days.]

Figure 1 to Figure 6 show the cumulative distribution functions (CDFs) of contact time and inter-contact time for meeting between nodes.

From Figure 1 and Figure 3 we can see that for both tablets 80% of the contacts last 20 minutes or less, and a significant fraction (40-50%) of contacts are very short (5 minutes or less). However, Tablet1 has 10% of long contacts lasting up to three hours while Tablet2 has no contacts lasting over 25 minutes. This is expected since all the contacts from Tablet2 were to the pickup truck, which is not expected to stay close to the drill for extended periods, while Tablet1 had contacts directly to the mine network which can potentially last for extended periods.

While contact time distributions for the two tablets are very similar, the inter-contact time (time between contacts) distributions shown in Figure 2 and Figure 4 are markedly different. The majority (80%) of inter-contact times are less than 2 hours, while Tablet2 has practically no inter-contact times less than 2 hours. However, even Tablet1 has a long tail with 10% of inter-contact times longer than 15 hours.

Finally, Figure 5 and Figure 6 show the contact and inter-contact behavior for the pickup truck, which acted as a data carrier, to the server through the mine network. We can observe that the pickup had frequent contact with the mine network as expected (50% of inter-contact times were less than 30 minutes) and that both short and long contacts were produced (50% were less than 30 minutes, while contacts up to multiple hours were observed).

It can be noted that both contact and inter-contact time distributions for all measured nodes fit well with the exponential model.

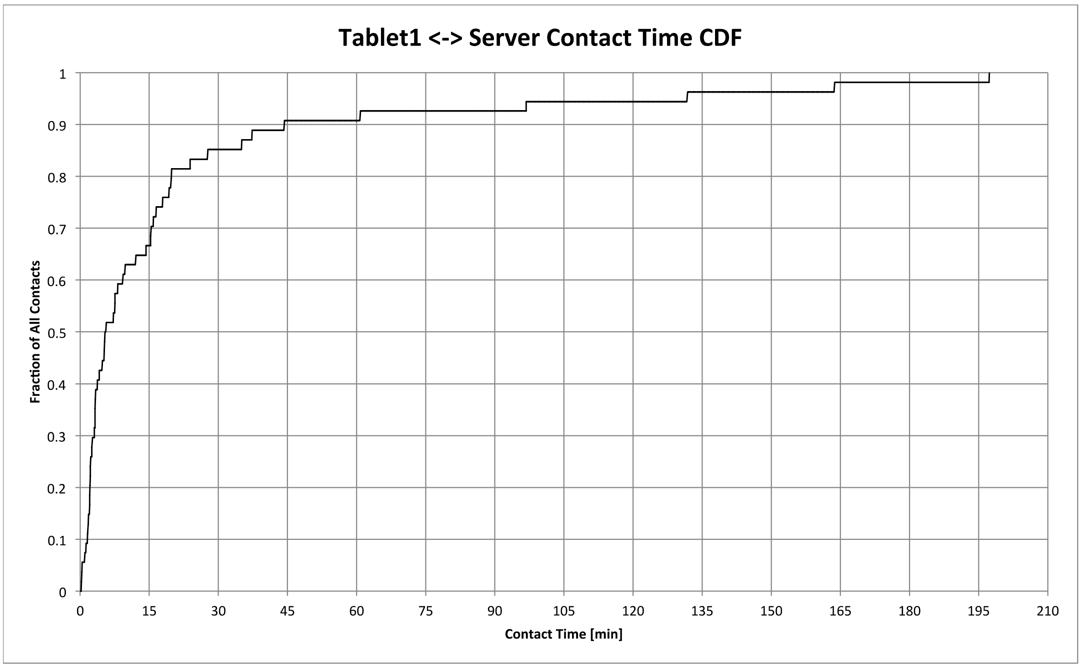

Figure 1 - Tablet1 to Server Contact Time CDF

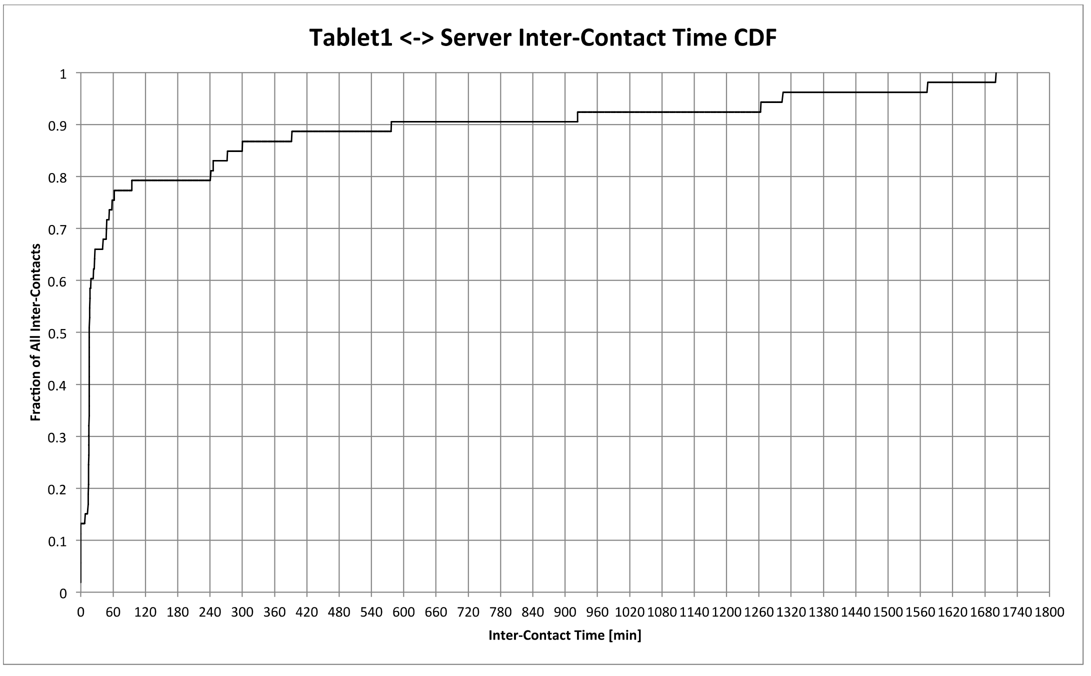

Figure 2 - Tablet1 to Server Inter-Contact Time CDF

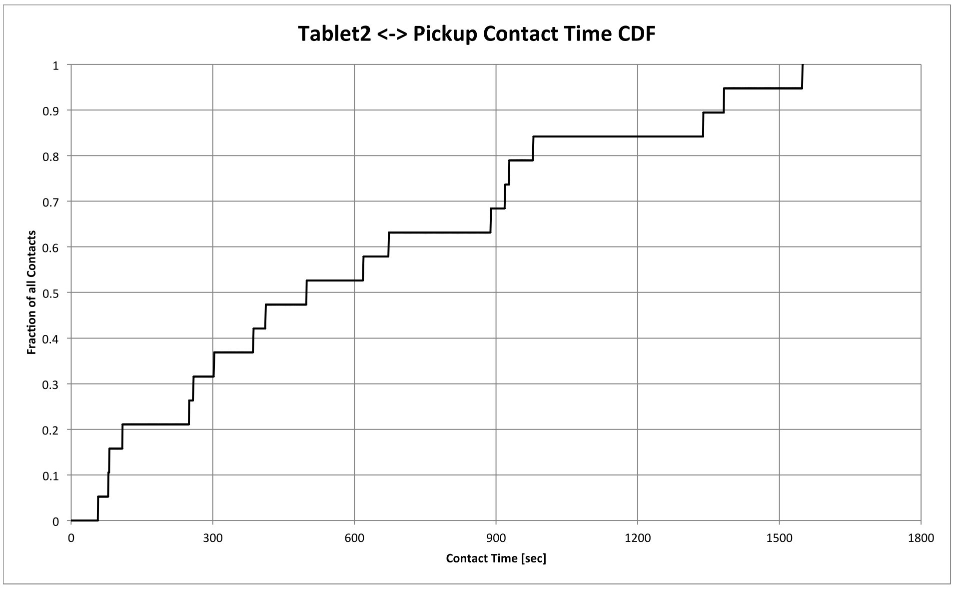

Figure 3 - Tablet2 to Pickup Contact Time CDF

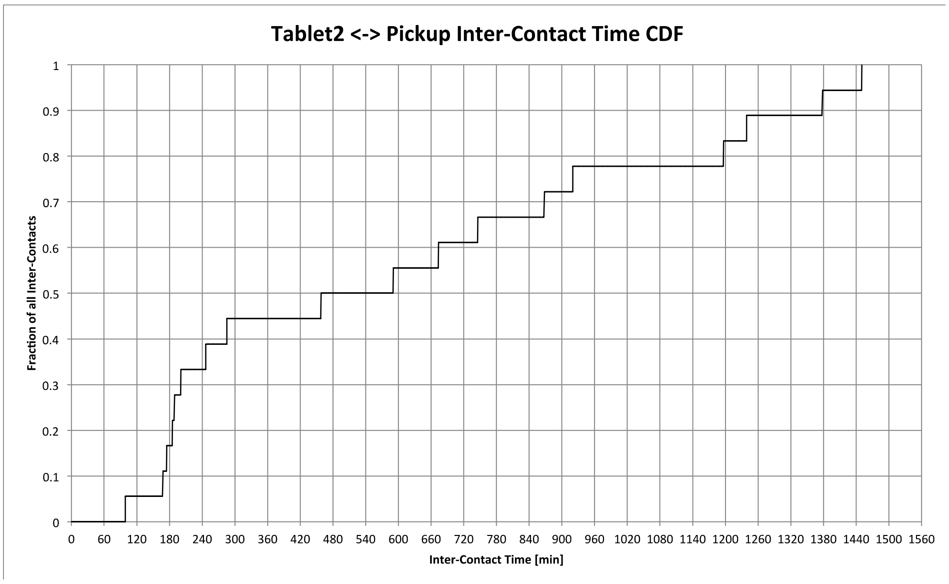

Figure 4 - Tablet 2 to Pickup Inter-Contact Time CDF

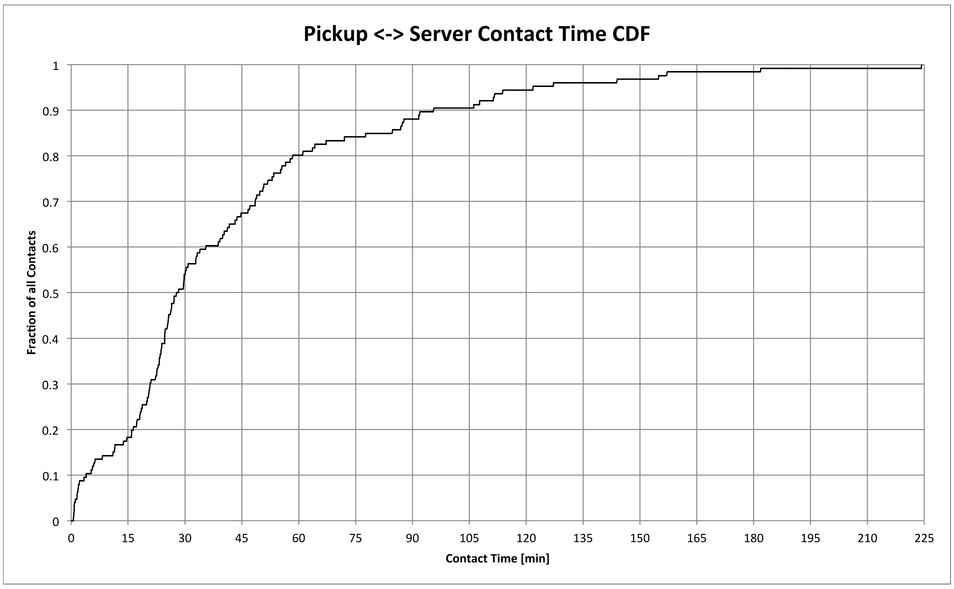

Figure 5 - Pickup to Server Contact Time CDF

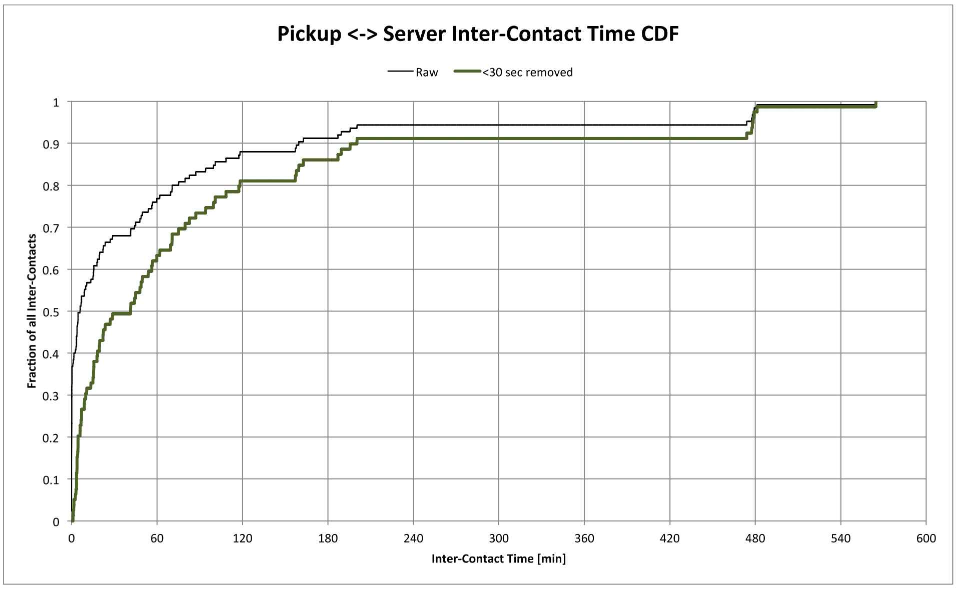

Figure 6 - Pickup to Server Inter-Contact Time CDF

3.2 Message Delivery

During the measurement period the tablets located in the drill machines generated messages destined for the sink node at the control room. These messages were both log files (generated infrequently) and application traffic (generated once per hour).

Messages generated in Tablet 1 were all transmitted directly to the sink over the mine infrastructure network when the node was within range. Messages from Tablet 2 were all carried from the tablet to the sink node by the intermediary node in the pickup truck. The resulting topology is shown in Figure 7.

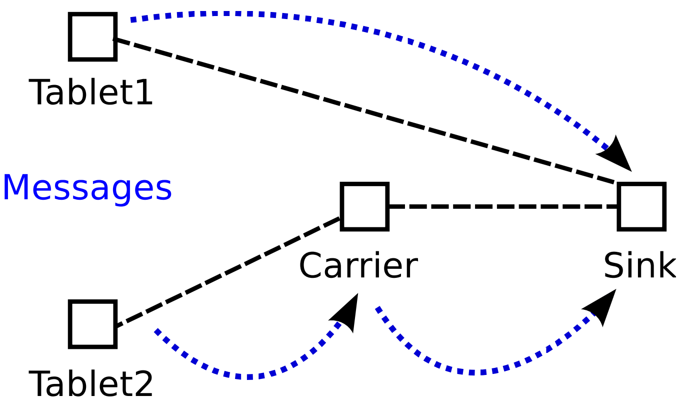

Figure 7 - Message Transfer Topology

Message generation and reception events are shown on a timeline in Figure 8. The arrows represent the delivery of data from the source to the destination starting from the creation time instance and ending at the reception time instance. Time axis is the number of hours from the start of the experiment.

For Tablet2 on the left side we can see that messages are generated continuously from the tablet, while on the Server side we can see that they are received in bursts at the server. This bursty reception is a natural consequence of the store-carry-forward message transmission where the data is collected and buffered for an extended period until an intermediary node becomes available and carries the data to the destination. We can see that some messages are transmitted very quickly when they are generated close to a time when the pickup truck collected data, while other messages had to wait for an extended period, up to 20 hours, until they were picked up and transmitted by the intermediary.

For Tablet1, the message creation was more sporadic due to the tablet being powered off while not in use. Since tablet 1 was in direct contact with the mine infrastructure network, there are many messages that get transmitted almost instantaneously.

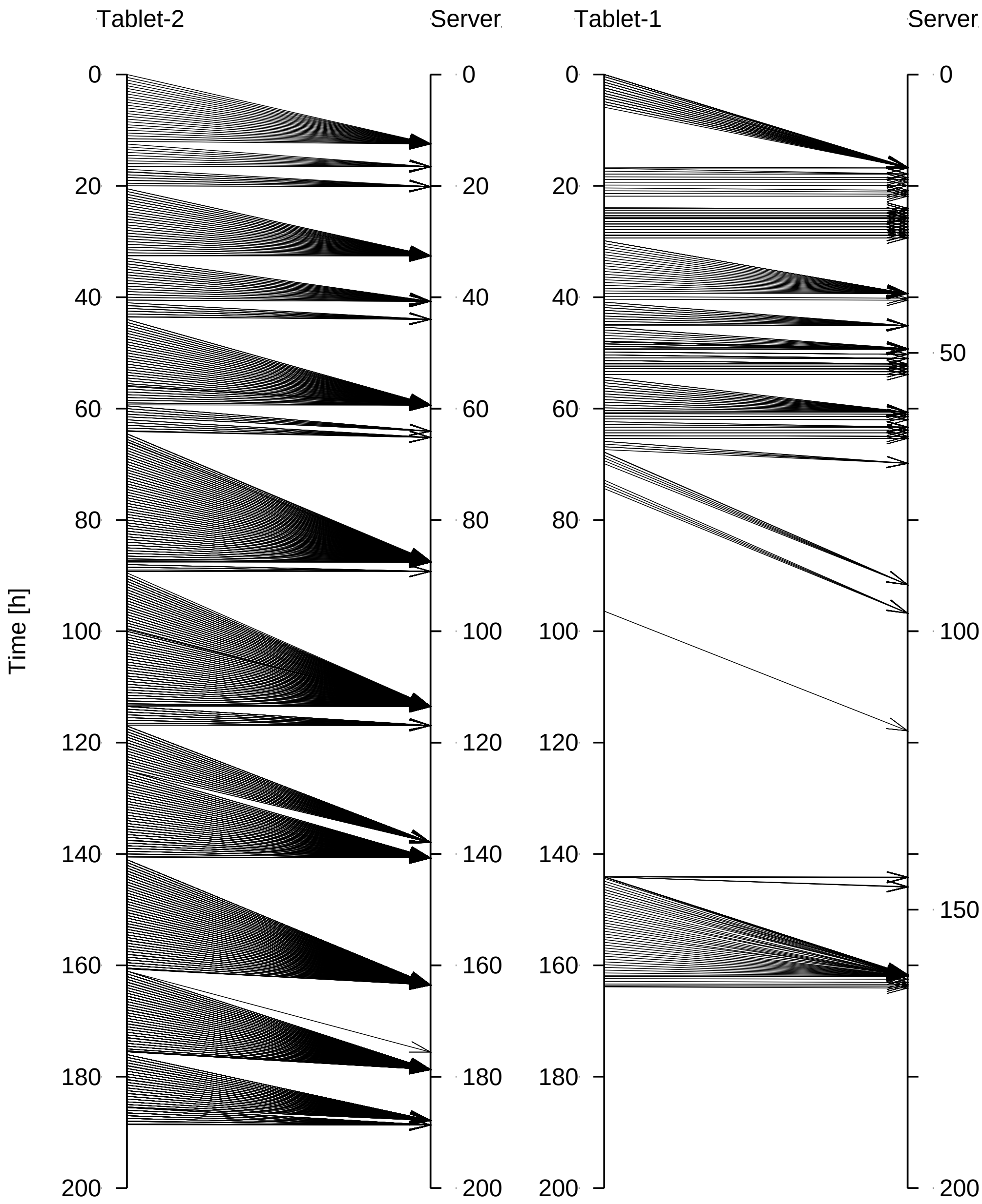

Figure 8 - Message Generation and Reception

To further study the message transmission process we have created a scatter graph with message creation times on the x-axis and (same message) reception times on the y-axis.

Scatter plot of the message creation time and reception time for Tablet 2 (Figure 9) shows the step function structure resulting from message transmission through an intermediary carrier. The diagonal line in the graph represents optimal message transfer where the server receives the message as soon as it is created by the tablet. Since Tablet2 had no direct connectivity with the server required to reach the optimal case, all the data points fall above the diagonal (as they are received later than they are created).

The steps visible in the graph are a result of the pickup truck carrying a buffer full of messages from the tablet to the server. The width of the step corresponds to the amount of time messages were buffered before the pickup arrived. We can see that there are multiple "long" steps of up to 20 hours, which correspond to periods when the pickup did not visit the drill machine, and "short" steps when the pickup visited the drill machine more frequently. Introducing more carriers, for example by enabling mobile phones to act as carriers, will introduce more short steps into the graph and improve the message delivery performance.

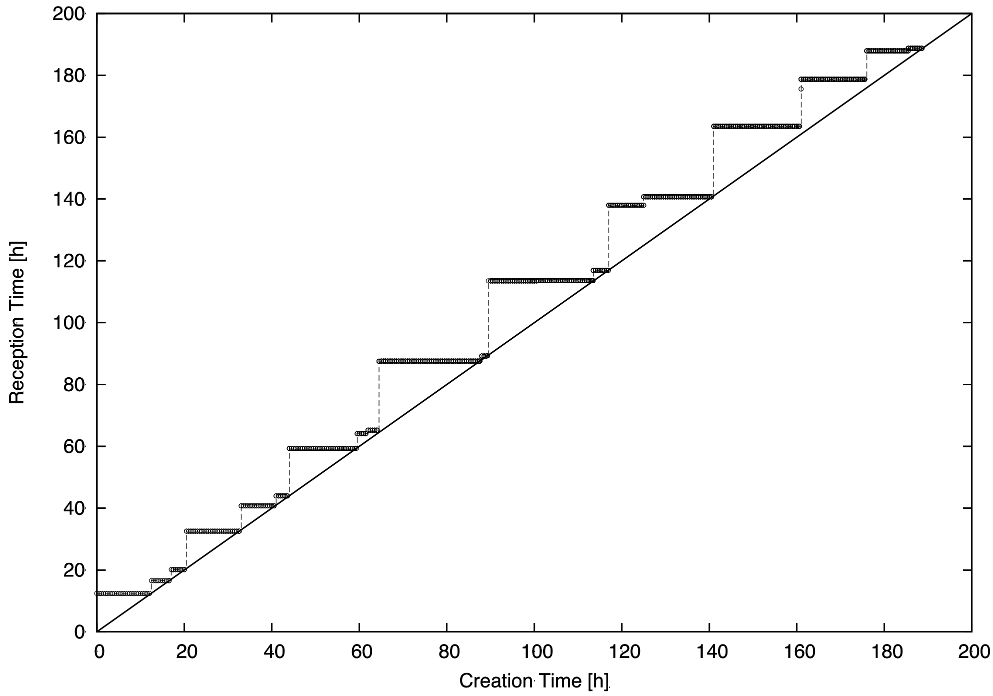

Figure 9 – Tablet2 Message Transmission Scatter Graph

The scatter plot for Tablet1 is shown in Figure 10. This graph also has steps when no contact was available to transmit the data similarly to Tablet2. However, since Tablet1 was in direct contact with the mine network there are periods where the message transmissions follow the diagonal line. This graph is not as complete as the previous due to the device getting powered down for extended periods during the measurements.

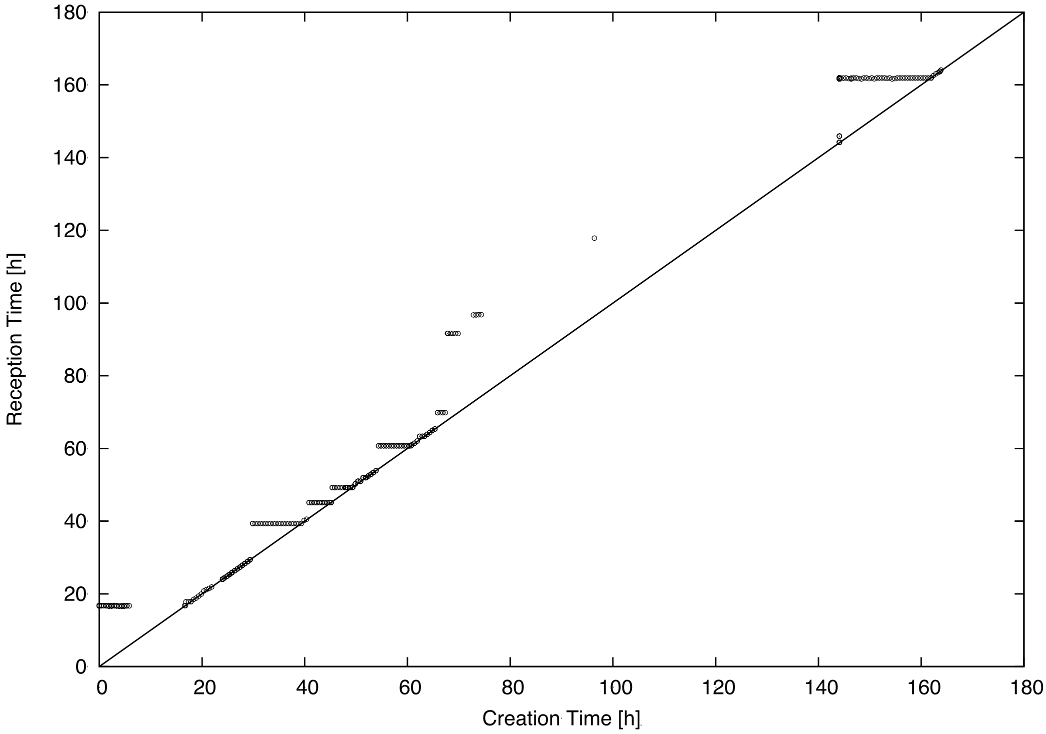

Figure 10 - Tablet1 Message Transmission Scatter Graph

The overall delay behavior is shown in Figure 11. The CDF of Tablet1 is skewed due to the sporadic message generation pattern, but is within expected range. As expected, larger proportion of messages from Tablet1 are delivered with low delays than from Tablet2. This is due to Tablet1 having direct contacts with the mine network during which all generated messages are delivered instantly. However, the performance of the two converge when taking into account messages with longer delays, i.e., those generated while Tablet1 was outside of the mine network coverage.

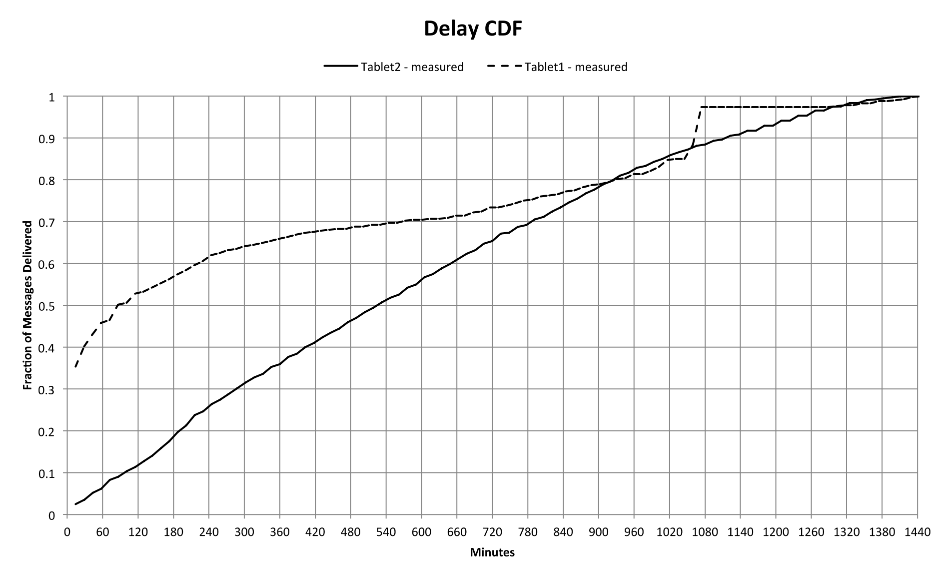

Figure 11 - Delay CDFs

There is an inherent dependency between the inter-contact times and the message delivery delays. The inter-contact time distribution determines how long messages must be buffered on average if no direct transmission link is available. This in turn determines the average message delivery delay. We can show this relationship between the inter-contact time CDF and the message delivery delay CDF for messages created in Tablet2 (Figure 12).

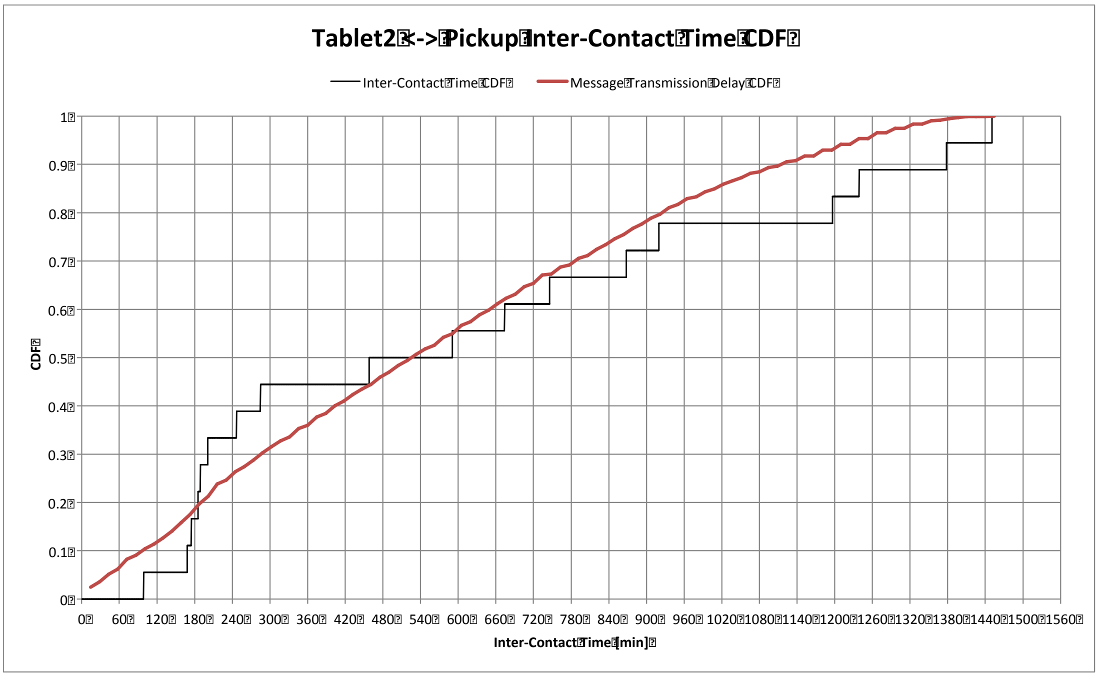

Figure 12 - Inter-Contact Time and Delay CDFs

This shows that the message delivery performance can be estimated from the inter-contact time distribution, which itself can be derived or measured from the movement of the nodes.

3.3 Simulation Study

In order to compare the performance of the network with and without DTN forwarding support, we conducted a simulation study based on the data collected from the Kemi experiment.

We extracted connectivity traces from the DTN router logs generated between 20.4.2012 and 28.4.2012. The connectivity trace was calculated directly from the observed TCP link uptimes and discovery beacons received by the nodes and therefore represent the actual realized connectivity.

The simulated topology is shown in Figure 13, and includes the two tablets operating in the drilling machines (Tablet1 and Tablet2), the pickup truck that functioned as a data carrier (Carrier) and the static node situated in the server room where the generated data was destined (Sink). The connectivity traces represent the capacity functions between the different nodes.

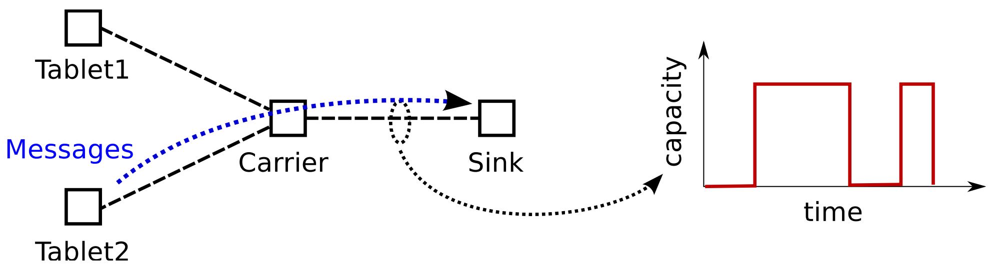

Figure 13 - Simulation setup.

Traffic was generated from Tablet1 and Tablet2 once per hour with each message size between 500KB and 1MB. The link capacities were assumed to be static 500KB/s.

The resulting delay CDF is show in Figure 14.

Tablet1 had a relatively small number of contacts with the intermediary pickup truck compared to the number of contacts with the mine network (16 intermediary contacts, 118 direct network contacts). However, the results show that even these few contact opportunities have a clearly noticeable impact on the delay performance. The delay performance remains the same for the 50% of messages that are delivered within three hours; however, the use of an intermediary DTN forwarder improves the performance for the remaining 50% of the messages. DTN forwarder is already an hour faster to reach 60% delivery rate, and is able to reach 90% delivery rate in 9 hours less than an infrastructure based network. DTN enabled network can deliver all the messages within 14 hours, while infrastructure-only network takes 21 hours to deliver all the messages.

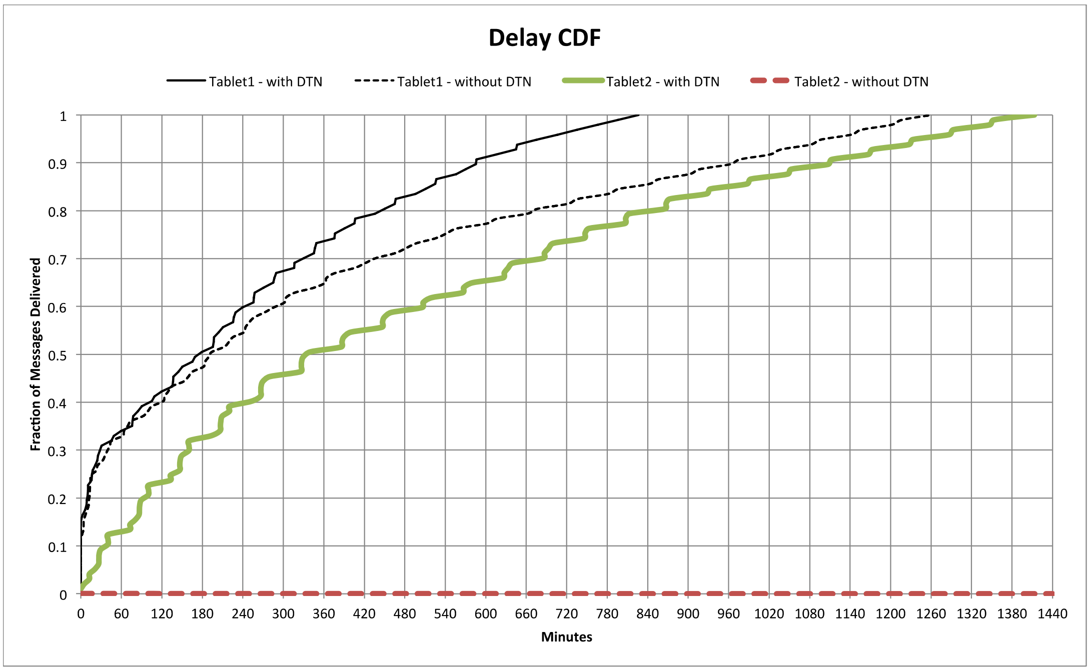

Figure 14 - Message delivery delay CDF.

Tablet2 had no observed contacts with the mine infrastructure network during the test period. This means that no data could be transmitted at all from the drill machine without DTN support. There were 37 observed contacts between Tablet2 and the Carrier pickup truck. This allowed messages to be transmitted to the control room Sink node even though no infrastructure was available. The resulting delay performance is comparable to the delay performance of pure infrastructure network with Tablet1, although on average it takes 2.5 more hours to reach the same delivery rate.

Overall, these results indicate that in a mining scenario, a purely DTN based network, with a small number of nodes, can provide performance comparable to a heavy infrastructure network. Furthermore, even when such infrastructure exists, already a small number of additional DTN forwarding opportunities has a noticeable impact on the overall performance.

4 Conclusion

In this deliverable we presented the test setup used in the Outokumpu Kemi mine between 20.4.2012 and 28.4.2012 to measure the real world performance of the system developed in the project. Our analysis of the collected data shows that the solution can be deployed and run in a production environment, and can transfer data with efficiency comparable to heavy mine infrastructure networks even in a case where all the traffic is carried by intermediary nodes.

We envision future work in improving the physical and algorithmic ‘ruggedness’ of our devices, doing larger scale deployments in environments with less infrastructure, and extending the platforms supported by the routing infrastructure to smaller mobile devices, such as mobile phones. We further believe that the platform is mature enough to support multiple end-user applications, although work still remains in developing operations management and application interfaces to support commercial deployments.

5 References

[1] Philip Ginzboorg; Teemu Kärkkäinen; Aki Ruotsalainen; Mikael Andersson; Jörg Ott, DTN Communication in a Mine, ExtremeCom 2010.

[2] K. Scott, S. Burleigh, Bundle Protocol Specification, RFC 5050, November 2007.