Succinct Data: Extreme Compression for ODK Forms

Making digital field assessment practical during disasters by minimizing data transmission size and costs

Paul Gardner-Stephen, Andrew Bettison, Romana Challans, Jeremy Lakeman, Timothée Malossane and Corey Wallis.
Flinders University,
Adelaide, Australia.

Matthew Lloyd and Ewan Caldicott
New Zealand Red Cross,
Wellington, New Zealand

Dione Gardner-Stephen
Serval Project, Inc.,
Adelaide, Australia.

Abstract

Field communications play a vital role in disaster relief and other humanitarian responses. However, the acute phase of these events is frequently characterized by scarcity of communications capacity, confounding the efforts of responders to develop the situational awareness that would enable them to maximize the effectiveness of their activities.

Access to high-bandwidth global connectivity is often limited to a few sites where existing infrastructure is still functional, or where temporary access has been rapidly deployed. The need to gather data, e.g., standardized field assessment reports, is not limited to these locations. A related problem is that any available communications links are likely to be overloaded. Therefore it is desirable to create a system that can make use of hand-held and low-cost global communications options. One such family of devices is those that provide access to the Iridium constellation's Short Burst Data capability, effectively providing a satellite based SMS service. However making use of such devices is problematic because of the extremely limited bandwidth they offer, and high cost-per-byte.

New Zealand Red Cross and the Serval Project [1 - 6] at Flinders University have responded to this need by creating the Succinct Data concept, wherein the salient information from a field report is automatically extracted, compressed and transmitted via any available means, with the full report following when the collection device, or another collection device it comes into contact with, eventually reaches a high-bandwidth communications link. In this way rich field assessment reports can be collected, with the highest priority information transmitted immediately, such as numbers of persons requiring assistance and basic assessment of dwellings, and with the additional rich information following when possible, such as detailed commentary, images and other media where appropriate.

In this paper we describe the work to date on implementing Succinct Data, and the creation of a new compression scheme that can compress ODK form instances to less than $1/12^{th}$ of their size – less than half the size achievable using gzip.

Keywords—telecommunications; data compression; serval project; succinct data; humanitarian; disaster relief.

I. Introduction

Many humanitarian field activities include the collection of data, that when collated, aids operational organizations in the fulfillment of their missions. For example, knowing the number, distribution and disposition of households in a disaster impacted area allows organizations like New Zealand Red Cross to plan the logistics of responding to the identified needs.

Historically the collection of such data in the field has been on paper-based forms. While the use of paper forms is resilient (depending on no infrastructure beyond the forms and a pencil), it is clear that the use of digital technologies have the potential to improve the process.

Smart phones and tablets can facilitate digital capture and transmissions of data to replace the paper-based forms. Digital transmission makes it possible to automatically aggregate, present, and appropriately share the collected data within and between organizations. This all has the potential to obtain better and more complete data, and to make it available for decision making much sooner, thus enhancing the effectiveness of relief operations, and hopefully, to save lives.

Several prerequisites must be met in order to make effective use of digital data collection in disaster zones:

First, suitable smart phones or tablets must be available, pre-configured and energy sources must be available to power them. These matters can be managed by organizations through their existing logistics processes, and is not dependent on the existence of any infrastructure in the zone of activity.

Second, there must be means for the devices to communicate the information that they collect. This prerequisite is more difficult to meet, as any local bulk digital communications infrastructure, e.g., cellular data or terrestrial internet, is likely to be damaged, over-loaded, or difficult to gain access to during the acute phase of a disaster. In some areas such infrastructure may never have existed. These difficulties essentially rule out dependence on any local terrestrial infrastructure, leaving satellite-based communications as the only viable option, albeit a relatively expensive, slow and low-bandwidth one.

Third, in light of the necessity to operate using satellite services, the data collection software for the devices must limit the volume of data being transmitted as much as possible to minimize both cost and delay, and maximize the impact of information that is communicated.

Fourth, a low-cost and portable satellite terminal is required to enable transmission of collected data as soon as possible. The smallest and cheapest terminals are Iridium Short-Burst Data modules, such as the DeLorme inReach. While such units are relatively cheap (around US$300) and small enough to be carried (small enough to fit in a large pocket), these benefits come at the expense of bandwidth (effectively of the order of a few bytes per second) and economy (data charges of the order of US$10,000 per megabyte). This greatly intensifies the need for the data collection software to minimize the size of transmitted data, and renders XML and other structured data formats effectively unusable, even when compressed. A specific goal of the Succinct Data concept is to be able to encode a realistic field assessment form in a single SMS or inReach text message.

To summarize, the prerequisite not currently met is that field data collection software be able to communicate collected information using extremely compact representations. The remainder of this paper describes our work in creating such a solution, New Zealand Red Cross and the Serval Project's Succinct Data system.

The primary contributions described in this paper are: (1) the development of an extremely effective compression scheme that uses semantic knowledge of the forms being compressed; (2) the implementation of the end-to-end cellular and satellite SMS-based data path for transmission and reception of forms compressed using this scheme, and (3) the integration of these and other existing components into a functional solution.

II. Architecture of Succinct Data

The purpose of Succinct Data is to accept the entry of digital forms in the field, to produce as succinct a representation of the entered data as possible, and to use that succinct representation to communicate the data to a location where it can be aggregated for consumption, as summarized in Figure 1.

The architecture is purposefully framed in terms of data flow. By breaking the system into a series of data creation, transformation and movement steps leads naturally to a modular approach, which in turn makes it easier to co-opt any suitable existing sub-systems, and to easily and efficiently replace individual sub-systems as Succinct Data evolves.

A. Digital Form Layer

For the digital form layer, we have chosen the Open Data Kit (ODK) [7] for form specification and entry, as it provides a flexible platform for the creation of forms (using a sub-set of the open XML-based JavaRosa standard) and a mature and robust application that can be used to complete forms on Android-based devices. For aggregation and presentation of collected data there are a number of options that can interoperate with this open-standard software. Formhub [8] was chosen for the proof-of-concept, as it was already being used by the authors for data collection. The modular approach means that each of these components could be easily substituted for others.

B. Transport Layer

For the transport later, we have implemented SMS, and are in the process of implementing support for the DeLorme inReach Iridium Short-Burst Data terminal.

Supporting SMS has several benefits. First, it provides a low-cost means for testing Succinct Data during development. Second, while SMS is not available during all disaster relief operations where Succinct Data may be used, it is much cheaper than satellite-SMS when it is available, and does not require satellite terminal hardware. Third, it makes it easier for others to build their own Succinct Data deployments for evaluation. Fourth, by using the standard Android intent system to send the SMS messages, we provide an interface that allows substitution with any other advertised short messaging service on Android. This interface is being used to add support for the inReach. We expect this to be available by the time this paper is in print.

Creation of a suitable coding/decoding layer with the desired performance is the primary focus of the remainder of this paper.

C. Succinct Data Coding/Decoding Layer

The coding/decoding layer transforms ODK form instances from their native XML format to an extremely terse representation. A reciprocal facility exists to transform in the reverse direction. For brevity, we will discuss the transformation only in the forward direction.

ODK XML form instances have a relatively simple XML structure, the semantics of which are defined by the form specification, which also exists as an XML document. Figure 2 shows an instance of a simplified field assessment form. Naive compression of encoded form instances such as this is typically able to halve their size, such that even short forms with trivial answers remain too large to be sent in a single SMS or satellite text message.

The form specification defines the type of each field in the form, as well as optionally describing constraints and other relationships in the fields. Figure 3 shows an excerpt of the form specification used to produce the example in Figure 2.

Using the form specification it is possible to improve the compression considerably. First, the names and order of the fields can be fixed, so that they can be completely implied in the compressed data. This achieves considerable savings, by effectively removing the common text from each form instance. In Succinct Data this removal of redundancy is achieved by creating an intermediate data format consisting of key value pairs instead of an XML document structure that we call the stripped data. Figure 4 shows the stripped data for the example of Figure 2.

The stripped data can be deterministically extracted without regard to the types of fields in the form, as it is a reversible general transformation between the XML and stripped data format. The stripped data can typically be compressed to around one fifth of the size of the XML. For some very short and simplistic forms this may be sufficient to allow encoding in a single cellular or satellite SMS message. To further improve compression it is necessary to make use of the semantic information in the XML form specification.

The most useful information is the type of each field, e.g., Boolean, multiple-choice, number, string or geographical position. For example, knowing that a field can take only one of two values, say, “yes” or “no”, allows it to be encoded in a single bit instead of requiring several bytes. This approach can also be used to encode whether each question has been answered at a cost of one bit per question. By employing an arithmetic or range coder [9] it is possible to similarly encode any multiple choice response using the least number of bits possible, which may not be an integer number of bits. For example, a 3-way decision can be encoded using 1.6 bits.

A similar approach can also be applied to numeric inputs, as there exists a finite set of possible values. Some fields also indicate further constraints, such as that the number must be non-negative, less than some maximum, or both. These constraints can be used to reduce the number of possible values, and thus the number of bits required to represent the value.

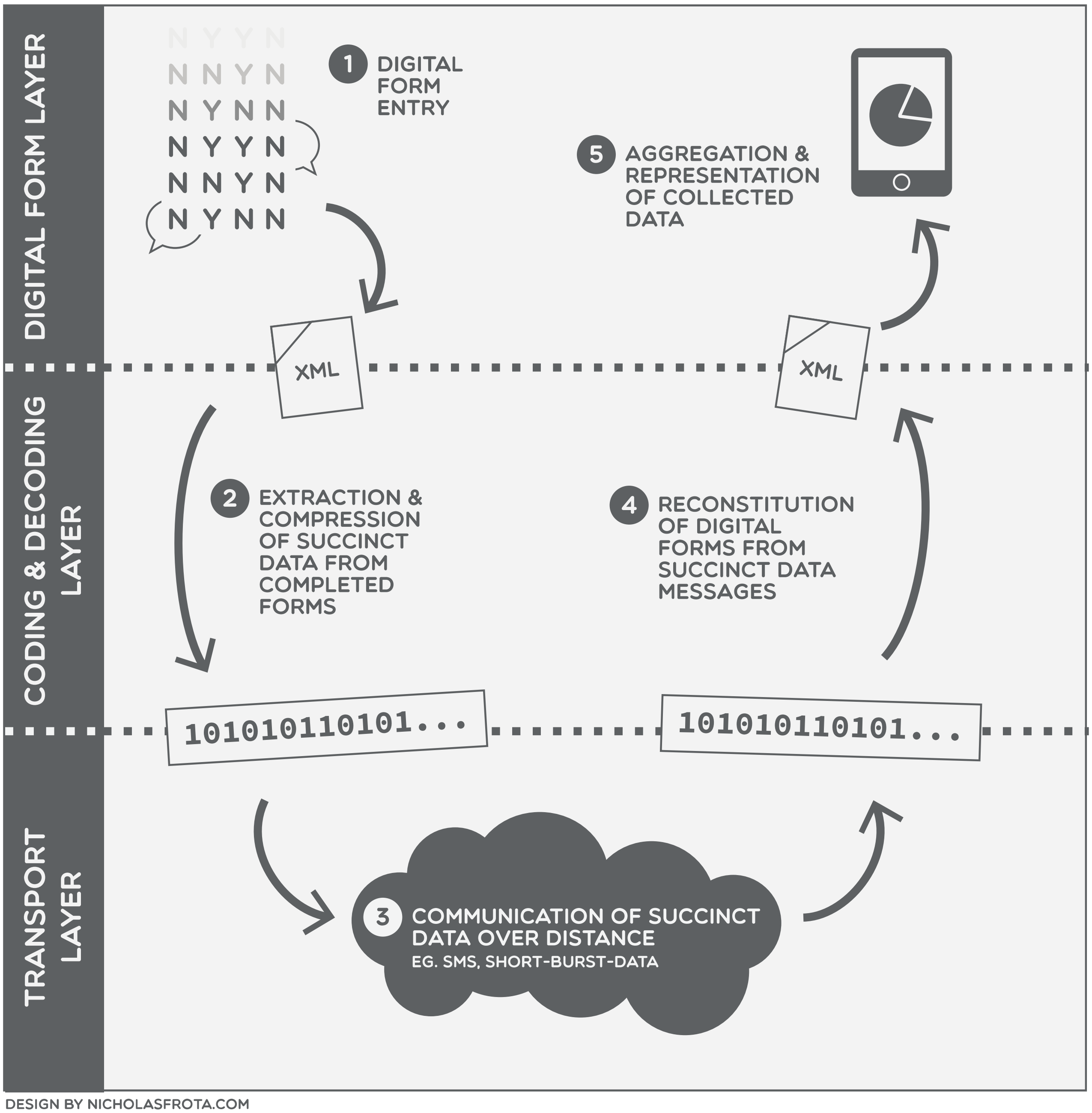

Fig. 1. Succinct Data Architecture. Forms are entered on a mobile device using ODK Collect resulting in an XML document (1). The XML documents are compressed using our open-source software to yield the succinct data (2). The succinct data is then transported using any available messaging facility, e.g., SMS or inReach satellite SMS (3). Our open-source software then is used again, this time to reverse the compression process and obtain an XML document (4) that can be used with the ODK analysis tools.

```xml
<?xml version='1.0' ?>
<nz_redcross_1a_people id="nz-redcross-1a-people" version="3">
<intro_note />
<village_name>Randomplatz</village_name>
<household_head>Smith</household_head>
<household_population>
  <adult_male>1</adult_male>
  <adult_female>1</adult_female>
  <child_01>2</child_01>
  <child_02 /><baby_01 />
  <total_population />
</household_population>
<household_casualties>
  <dead /><injured /><missing /><special_needs />
</household_casualties>
<household>
  <damaged_house /><usable_kitchen /><household_location />
  <household_photo />
</household>
<start>2014-05-07T05:18:59.467-07</start>
<end>2014-05-07T05:19:11.827-07</end>
<deviceid>359372042147975</deviceid>
<meta>
  <instanceID>uuid:5f29c5eb-7a29-4cc4-bf5c-5eb4bda9b4fa</instanceID>
</meta></nz_redcross_1a_people>
```


Fig. 2. Example ODK form instance. Form instances are represented by XML documents that detail the responses to each question. These files are typically of the order of one to a few kilobytes. Using general purpose compression programmes such as gzip or bzip2 they can be roughly halved in size, but remain much too large to fit in a single SMS.

Field types such as time, date or geographical position can be represented as a vector of constrained numeric values, e.g., latitude between -90 and +90 plus longitude between -180 and +180, thus facilitating their efficient encoding. This method allows encoding of such types using around an order of magnitude fewer bytes.

Any field type that is unknown is encoded as a text string. Modest savings can be made with such fields by applying well-known text compression schemes. However, the short length of the fields causes general-purpose text compression systems to perform poorly, necessitating a specialized short-text compression scheme. Such a scheme, SMAC [10], was created by several of the authors in preparation for this use. SMAC supports both ASCII and Unicode characters and typically halves the length of text strings. SMAC also provides access to a range coder that can be easily used for encoding the other field types. Thus while other schemes are possible, SMAC provides all of the necessary machinery for the compression of complete forms.

```
<bind nodeset="/nz_redcross_1a_people/intro_note" readonly="true()"
   type="string"/>
<bind nodeset="/nz_redcross_1a_people/village_name" type="string"/>
<bind nodeset="/nz_redcross_1a_people/household_head"
   type="string"/>
<bind constraint=". >= 0"
   jr:constraintMsg="Negative values not allowed"
   nodeset="/nz_redcross_1a_people/household_population/adult_male"
   type="int"/>
...
<bind nodeset="/nz_redcross_1a_people/household/usable_kitchen"
   type="select1"/>
<bind nodeset="/nz_redcross_1a_people/household/household_location"
   type="geopoint"/>
<bind nodeset="/nz_redcross_1a_people/household/household_photo"
   type="binary"/>
<bind jr:preload="timestamp" jr:preloadParams="start"
   nodeset="/nz_redcross_1a_people/start" type="dateTime"/>
...
<bind jr:preload="property" jr:preloadParams="deviceid"
   nodeset="/nz_redcross_1a_people/deviceid" type="string"/>
...
<select1 ref="/nz_redcross_1a_people/household/usable_kitchen">
  <label>Is the kitchen usable?</label>
  <item><label>Yes</label><value>yes</value></item>
  <item><label>No</label><value>no</value></item>
</select1>
```

Fig. 3. Example ODK form specification. Various field types can be specificied, including simple input values, e.g., for village_name, which accepts a string, constrained input values, e.g., for adult_male, which requires that the value be non-negative, or multiple choice, e.g., for usable_kitchen which effectively describes a boolean. These specifications are interpretted by the succinct data compression layer to encode each response using as little as one bit.

The system should support the use of many different forms, and ensure that each succinct data message is decompressed with respect to the correct form to ensure that nonsensical output is not produced. This is achieved by commencing each succinct data message with a 48-bit form type identifier consisting of the prefix of the MD5 hash of the form. This allows the receiver of a succinct data message to determine the form that was used to produce it, and ensure that it is decompressed appropriately. The hash is long enough to make the probability of a hash collision negligible in realistic use. Any actual hash collision can be dealt with by slightly modifying one of the forms, for example adding a new dummy field or changing the name of an existing field.

Using the field assessment form from the previous examples this method yields succinct data messages as small as 40 bytes for simple forms, the bulk of which is the meta data present in any ODK form (the form identification prefix, start and end time of form entry, device ID and form instance UUID). This corresponds to a compression ratio of around 12:1, which is approximately five times smaller than can be achieved using general-purpose text compression on single messages (Table 1) or even batches of messages (Table 2).

```
village_name=Randomplatz
household_head=Smith
adult_male=1
adult_female=1
child_01=2
start=2014-05-07T05:18:59.467-07
end=2014-05-07T05:19:11.827-07
deviceid=359372042147975
instanceID=uuid:5f29c5eb-7a29-4cc4-bf5c-5eb4bda9b4fa
```

Fig. 4. Example stripped data. The stripped data consists of only the field names and values extracted from an XML ODK form instance document, in this case the document shown in Figure 2.

D. *Discussion of compression performance*

Considering now the expected effectiveness of this compression scheme in general, we take two approaches. First, we reason about the expected performance based on the semantics of the forms being compressed. Then second, we present the results for 676 instances of a short weed control survey used by the authors together with the local Natural Resources Management board.

As described above, the Succinct Data compression scheme leverages semantic information extracted from the ODK form specification to reduce the entropy of the documents being compressed. This is a deterministic process, allowing the expected size of compressed instances of forms to be calculated a priori. Succinct Data allows each question to be answered or not, represented by a single bit.

Multiple-choice, Boolean, integer, fixed-point, date and geographic points are all encoded close to the entropy of these fields by applying arithmetic coding to the semantically decomposed fields. For example, Boolean values are encoded using a single bit, while integer fields are encoded using a range coder with equal probability for each value. Fixed-point decimal values are encoded similarly to integer values. The minimum and maximum values of fields are extracted from their ODK form specification. Thus the compressed size of a form can be reduced by more tightly constraining the acceptable range of inputs for a given field.

Dates and geographic points are processed by parsing their textual representations, and then encoding their components similarly to the case of integer and fixed-point values. For example, date-time fields formatted like 2014-05-07T05:18:59.467-07 are converted to seconds since 1970 UTC. Thus dates between 1970 and 2038 can be represented in 32 bits. In a future release it will be possible to further constrain dates to obtain further savings. We also anticipate applying this to dependent times within a form, where it is known that one time-stamp must follow another, allowing the range of valid times to be further constrained, and consequentially reducing the number of bits required to encode.

Geographical coordinates are also decomposed, this time into latitude and longitude components. Here the precision is controllable, with 32-bits being required to encode a location to approximately 1m accuracy, or using less bits if a coarser location is allowed.

UUIDs also receive special treatment as they appear in all forms, being represented in 128 bits, which is equal to their internal entropy. An option is provided to truncate UUIDs where compressed forms are just over an SMS message size quantum.

Consideration was given to using range or arithmetic coding to weight the possible values for each field away from equal probability to reduce compressed message size on average. While relatively easy to implement, this was not done as it was seen as counterproductive in that it would cause inadequate compression of forms containing reports of uncommon or unexpected conditions – which may be critical information for responders. The quantized capacity of SMS-type transports is another factor in support of our decision. Provided that the expected size of a given form falls within a quantum, rather than just exceeding one, there is little motivation to obtain further savings. The predictability of compressed message size supports automatic evaluation of such boundary conditions, although we are yet to implement such automation.

The exception to the predictability of compressed message size is for text fields. In that case the SMAC compression scheme is used to reduce these fields by approximately 50% on average. Indeed, this is the sole area where the Succinct Data scheme is currently unable to encode a given form close to its underlying entropy, yet even in this situation the compression is the best in its class, as described in [10].

To compare the effectiveness of our compression scheme with the existing art, we processed 676 weed control form instances using Succinct Data and also with bzip2 and gzip. These forms include a UUID, start and end times, several multiple choice questions, a GPS location, the filename of a photograph, and a device identifier of which Succinct Data has no intimate knowledge and is forced to represent as a text string.

For bzip2 and gzip we compressed all form instance XML files together, giving the maximum possible advantage to those tools. Succinct Data was employed in its normal mode, including the addition of the six-byte form identification header, which alone requires 4,056 bytes. Such an overhead would be required if using gzip or bzip2, but is not imposed here.

Despite these handicaps that advantage gzip and bzip2, Succinct Data convincingly outperforms both as shown in Table 2, with gzip requiring twice the number of bytes. Even bzip2 is unable to match the performance of Succinct Data, despite the nature of the Burrows-Wheeler transform [11] that allows it to compress collections of similar documents so well also makes it implausible to be used to incrementally compress forms in this way.

If the six-byte form identification header were imposed on gzip2 [12] and bzip [13], or conversely, if longer forms were considered that better amortize the cost of the form identification header, the advantage of Succinct Data becomes even greater, suggesting that Succinct Data will can readily provide two to three times the information content per SMS message compared with existing compression schemes.

III. Implementation

To make the system practical, we have created open-source tools [14 – 15] that automatically extract the semantic information about fields from a form specification, and then use that to compress the stripped data from completed forms to produce the succinct data. The tools also provide the reverse path, decompressing succinct data to stripped data, and then reproducing the XML form using a simple template driven approach.

These tools have been integrated into an open-source Android application that allows a user to select a form to complete, launches ODK Collect to allow completion of the form, and then captures the completed form. The completed form is then compressed using the methods described above, and dispatched using the Android SMS intent, which allows dispatch by cellular SMS. A customized instance of FormSMS receives and decompresses the succinct data via SMS, and collates it for analysis. The complete form is also dispatched via the Serval Mesh Rhizome service for horizontal transport allowing any included rich media – which is automatically excluded from the succinct data – to be received in due course.

IV. Conclusions

In this paper we have described our progress towards realizing Succinct Data as a practical and cost-effective means for collecting field data during disaster and similar situations. Our objective was to allow the reversible encoding of a realistic field assessment form in a single SMS message. We have achieved that objective with room to spare, having demonstrated the compression of forms in less than 100 bytes. Moreover, we have created a general purpose tools that will allow the methods described in this paper to be easily applied to a wide range of forms created using the ODK framework. Thus, not only have we proven the possibility of Succinct Data, but made considerable progress to making it possible for operational organizations to make use. We expect that it could be ready for evaluation by early adopters before the end of the year.

V. Future Directions

Ongoing work is currently focused on implementing the inReach satellite-SMS interface, which will complete the core functionality of Succinct Data and allow New Zealand Red Cross to begin evaluation for potential use in their operations, including as a replacement for satellite phones in various pacific nations.

Further work is also planned on improving compression in several ways. First, implementing variable precision for various data types. For example, rounding the geographic position to the nearest 10m may be acceptable, and can save several bits. Similarly one might elect to round timestamps to the nearest minute or hour. Second, the range of values for fields may be further constrained. For example, dates may be assumed to be within a given calendar year, and geographical positions within a given gradicule. Third, we plan to improve the performance of our short-text compressor over time.

TABLE I. Example Compression Performance of Succinct Data for a Sample Form

Format	Size		
	Size (bytes)	***Percentage of size versus XML***	***Number of SMS messages required[a]***
ODK XML	743	100%	5
General-Purpose Compression (gzip -9)	401	53%	3
Stripped Data	218	29%	2
Succinct Data[b]	61	8.2%	1 (<1/2 filled)

a. Assumes SMS messages consist of up to 160 characters covering printable ASCII characters.

b. Includes six byte form identifier.

TABLE II. Comparative Compression Performance of Succinct Data Versus Gzip and Bzip2 for 676 Weed Control Survey Form Instances

Method	Size		
	Size (bytes)	***Avg. bytes per form***	***Compressed size compared with original XML***
Succinct Data	27,254	40.32	7.2%
gzip -9	54,115	80.05	14.4%
bzip2 -9	43,605	64.50	11.6%

References

[1] P. Gardner-Stephen (2011). "The Serval Project: Practical Wireless Ad-Hoc Mobile Telecommunications." Internet: http://developer.servalproject.org/files/CWN_Chapter_Serval.pdf [21 May 2014].

[2] P. Gardner-Stephen, J. Lakeman, R. Challans and A. Bettison (2013). "The rationale behind the serval network layer for resilient communications." *Journal of Computer Science,* 9(12) pp. 1680-1685. [http://dx.doi.org/10.3844/jcssp.2013.1680.1685]

[3] P. Gardner-Stephen, J. Lakeman, R. Challans, C. Wallis, A. Stulman and Y. Haddad (2012). "MeshMS: Ad Hoc Data Transfer within Mesh Networks." *International Journal of Communications, Network and System Sciences,* 5(8) pp. 496-504. [http://dx.doi.org/10.4236/ijcns.2012.58060]

[4] P. Gardner-Stephen (2011). "Sustaining Telecommunications Capability and Capacity during

Acute Phase of Disasters and Disaster Responses." *Prehospital and Disaster Medicine.* Wisconsin, USA: Cambridge University Press, pp. s101-s102. [http://dx.doi.org/10.1017/S1049023X11003207]

[5] P. Gardner-Stephen (2011). "The Serval Project: Creating a Robust Infrastructure-Independent Communications Safety Net," presented at The Engineering and Physical Sciences in Medicine and the Australian Biomedical Engineering Conference. Australia: Exploring New Territory: Innovative Solutions in Medicine and Health Physics. Darwin. Aug 2011, pp. 84-85.

[6] P. Gardner-Stephen, S. Palaniswamy. (2011). "Serval Mesh Software-WiFi Multi Model Management," presented at the 1st International Conference on Wireless Technologies for Humanitarian Relief in ACWR 11, New York, pp. 71-77. [http://dx.doi.org/10.1145/2185216.2185245]

[7] Y. Anaokwa, C. Hartung, W. Brunette, A. Lerer, G. Boriello. (2009). "Open Source Data Collection in the Developing World." *IEEE Computer* 42(10) pp 97-99. [http://dx.doi.org/10.1109/MC.2009.328]

[8] "Formhub" Internet: https://formhub.org, [21 May 2014].

[9] I. Witten, R. Neal and J. Cleary (1987) "Arithmetic Coding for Data Compression." *Communications of the ACM* 30(6) pp 520 – 540 [http://doi.acm.org/10.1145/214762.214771]

[10] P. Gardner-Stephen, R. Challans, A. Bettison., C. Wallis, J. Hampton and J. Lakeman. (2013). "Improving Compression of Short Messages." *International Journal of Communications, Network and System Sciences,* 6(12) pp. 497-504. [http://dx.doi.org/10.4236/ijcns.2013.612053]

[11] M. Burrows and D. Wheeler. (1994) "A block-sorting lossless data compression algorithm." HP Labs Technical Reports, SRC-RR-124 [http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-124.pdf]

[12] Gailly, J. and Adler, M. (2003) "gzip." Internet: http://www.gzip.org [4 July 2014].

[13] Seward, J. (2014) "bzip2." Internet: http://www.bzip.org/ [4 July 2014].

[14] P. Gardner-Stephen, R. Challans, A. Bettison., C. Wallis, J. Hampton and J. Lakeman. (2014) "Survey acquisiton management tools for Android." Internet: https://github.com/servalproject/survey-acquisition-management [27 May 2014].

[15] P. Gardner-Stephen, R. Challans, A. Bettison., C. Wallis, J. Hampton and J. Lakeman (2014) "Short strings compression library." Internet: https://github.com/servalproject/smac [27 May 2014].