

# Community Seismic Network: A Dense Array to Sense Earthquake Strong Motion

by Robert W. Clayton, Thomas Heaton, Monica Kohler, Mani Chandy, Richard Guy, and Julian Bunn

## INTRODUCTION

The Community Seismic Network (CSN) is currently a 500-element strong-motion network located in the Los Angeles area of California (see Fig. 1). The sensors in the network are low-cost microelectromechanical (MEM) accelerometers that are capable of recording on scale up to accelerations of  $\pm 2g$ . The primary product of the network is a set of measurements of ground shaking in the seconds following a major earthquake. An example of this is shown in Figure 2. The shaking information will be contributed to U.S. Geological Survey products such as ShakeMap (Wald *et al.*, 1999) and ShakeCast (Wald *et al.*, 2006), with the goal of providing first responders a proxy for damage that can guide efforts immediately following the event. The basic premise is the strong ground-motion shaking varies on a subkilometer scale, which will require a dense network to meaningfully measure the shaking. Evidence for this comes from earthquakes recorded by dense oil company surveys in the Los Angeles area (Clayton *et al.*, 2011).

The CSN is presently concentrated in the northern Los Angeles region, but the plan is to expand the network over the entire populated regions of the Los Angeles basin, at which point it will become a network of thousands of sensors. To handle this size of network, we have developed a cloud-based processing system that dynamically expands in processing power during an earthquake to handle the increased load. We also use a distributed system in which single-board computers (SBCs) attached to the sensor carry out the basic detection and feature extraction, and the results are sent immediately to the cloud. This eliminates delays due to transport of the waveform data, and increases the likelihood that the event detection information will be sent to the region before the potential collapse of the communication infrastructure. The use of the cloud is also important for robustness of the system in that the central processing system is outside the reporting area and is distributed across multiple locations.

To expand the system across the entire Los Angeles basin, we have developed a plan to deploy sensors in schools. As a first step, we have started to instrument the 1200 campuses of the Los Angeles Unified School District (LAUSD) (see Fig. 3). We have already deployed sensors at 100 of these campuses as a test of the plan, and these sensors have been running successfully for several months. The next step will be to instrument the campuses of the other school districts in the Los Angeles basin,

which, when complete, will provide an excellent backbone for the network. Private volunteers will then deploy sensors in the network, resulting in increased density, as has already been achieved in Pasadena.

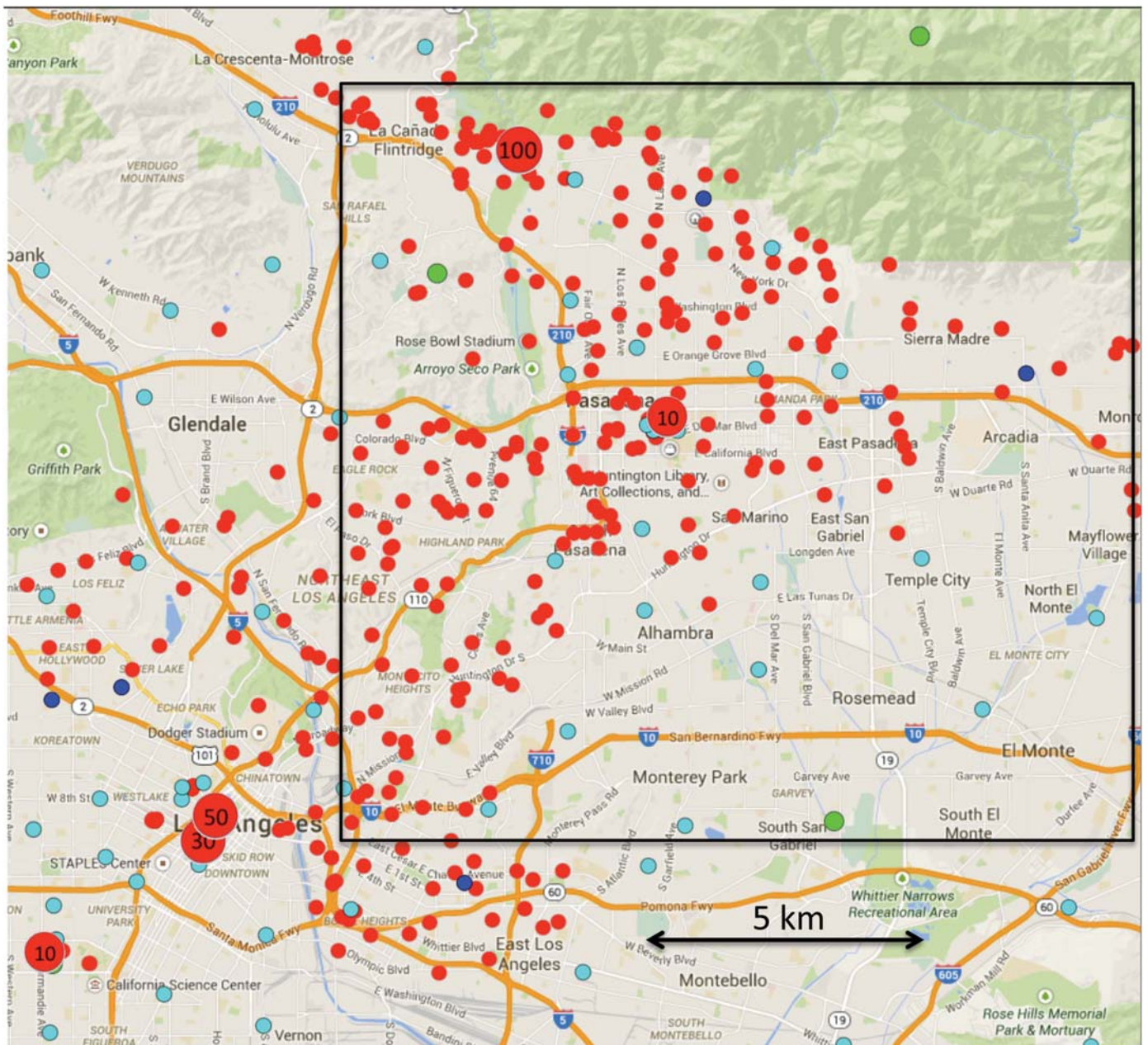
CSN started five years ago and is now developed into a functioning seismic network. The original system was described in Clayton *et al.* (2011), and here we describe the evolution of that system. The plan is to integrate the operation of the CSN with the Southern California Seismic Network (SCSN), which is the authoritative network for reporting earthquakes in southern California. The SCSN maintains an array of high-quality broadband and strong-motion sensors in southern California, with an average station spacing of 10 km in the Los Angeles area. The CSN event detection and attribute software is being modified to conform to the method used by SCSN, so that the pick and amplitude information can be fed directly into that system. The CSN is also instrumenting buildings with the goal of providing detailed measurements of the structure before, during, and after an earthquake, so as to monitor the state of health of the structure.

## LOW-COST SENSORS SYSTEMS

The use of low-cost sensors as shown in Figure 4 is key to allowing the dense network to be built. CSN is currently using the Phidget 1041, which is a 3-axis MEM accelerometer that has been modified by the manufacturer to have an on-scale range of  $\pm 2g$ . It has a sensitivity of approximately 70  $\mu g$ , which means it can detect a magnitude 3 event anywhere in the Los Angeles basin. The sensitivity is not as good as a standard seismometer but does provide reasonable waveforms and picks for close events as low as magnitude 2–3, and excellent recordings for larger events.

The performance of this sensor and other Class-C devices was recently compared in a study by Evans *et al.* (2014), and the Phidget sensor was found to be tied for top of this class. In Figure 5, a side-by-side comparison of the Phidget and a Class-A accelerometer (an EpiSensor) is shown for the La Habra earthquake (29 March 2014  $M_w$  5.1) recorded in the PASC vault, which is at a distance of 30 km from the epicenter. The two sets of recordings compare very well, both in terms of amplitude and waveform.

The sensor connects to the host computer via a USB port. In the original design of our sensor package, the sensor was



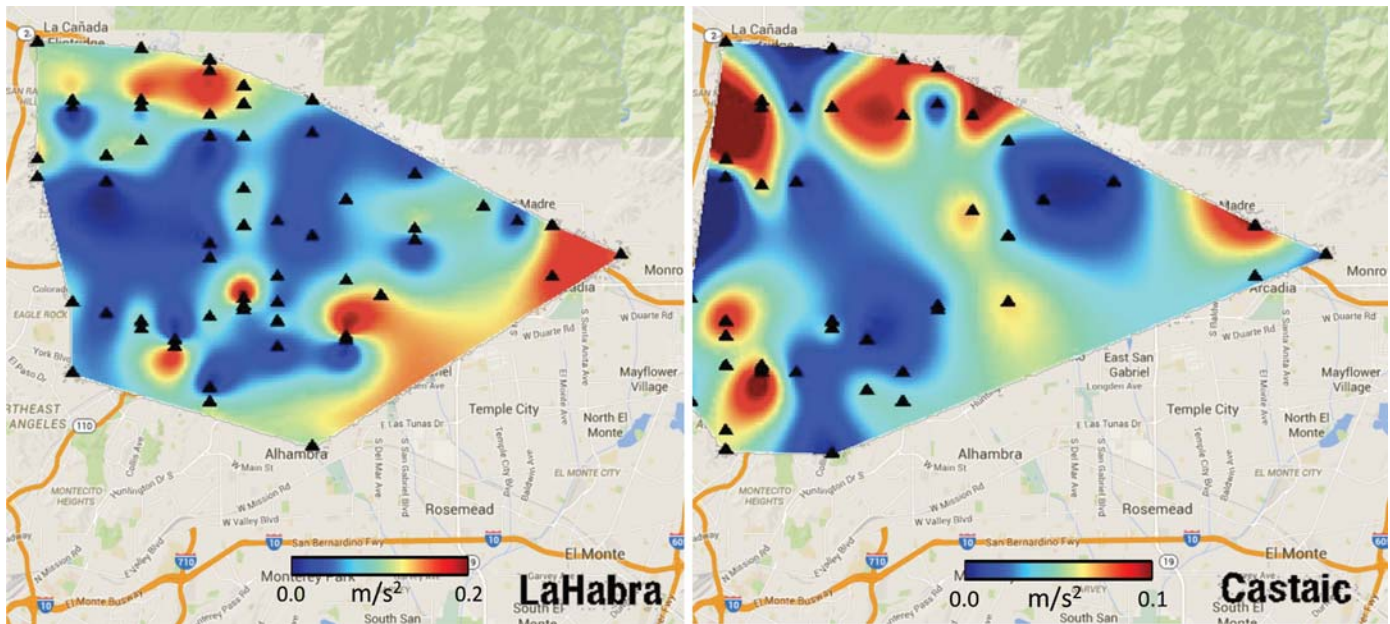
▲ **Figure 1.** Map of networks in the Pasadena area. Red dots are Community Seismic Network (CSN) stations, dark blue are Netquakes stations, light blue are strong-motion stations, and green are Southern California Seismic Network (SCSN) stations. The large red dots with numbers denote buildings with multiple CSN sensors. The average nearest neighbor spacing of the CSN sensors is 0.8 km. The spacing of all other strong-motion networks in this area is 4 km. The box denotes the region covered in Figure 2.

connected directly to a general-purpose host computer (such as a Windows or Mac desktop), which ran the data handling and communication software. However, the change in computing paradigm away from 24/7 desk-side computers toward laptops and tablets has made this model less useful. Instead, we now use a small-dedicated SBC for each sensor. We use a SheevaPlug SBC, which has an ARM-9 1-GHz processor and 512 Mbytes of memory. We have found that this device is reliable and has sufficient processing power to analyze the sensor data in real time. The entire system is housed in a plastic box (see Fig. 5),

which also contains a small battery backup, which allows the system to continue recording a major event should power be lost. The data are written to nonvolatile memory and the system is able to store approximately 4 days of data. Some of the sensors also have SD cards that provide much longer data backup.

The Phidget accelerometer samples data at 250 samples per second and our software decimates the data to 50 samples per second and stores it. The sensor client software periodically polls a network time protocol (NTP) server to maintain time





▲ **Figure 2.** Peak ground acceleration: The peak ground acceleration for the La Habra earthquake (left panel) and the Castaic earthquake (right panel). The La Habra earthquake (29 March 2014) was magnitude 5.1 at a distance of 30 km from the center of the plot. The Castaic event (4 January 2015) was magnitude 4.1 at a distance of 70 km.

synchronization that is accurate to within a few milliseconds. This clock is used to insert timestamps into the data stream and to correct slight irregularities in the digitization rate. Although NTP time is not as accurate as Global Positioning System time, it is universally available, even in buildings and other closed spaces where the bulk of our sensors are deployed.

The client has two different modes for sending data. In the first mode, event-driven data such as pick times and amplitudes are sent as soon as sufficient signal has been received to make the measurements. Currently, sufficient signal is set at 1 s, with updates following until the signal has diminished. The length of the window will trade off with the expected uptime of the Internet during a major earthquake. This mode avoids waveform transport latency that can occur if the picks are made at a central site. In the second mode, the waveform data for small chunks of time (10 min) is sent as soon as it has been accumulated. To reduce bandwidth, this step can be omitted or limited to only windows in which unusually large accelerations have been detected, but so far bandwidth has not been an issue and consequently all stations currently send all of the waveform data.

In addition to data communication, the client sends regular state-of-health messages and checks with the cloud for software and parameter updates. Updates consist of either simple parameter changes or new versions of the client code itself. In the latter case, the client downloads the new version of the code, and after checking that the new code has a valid cryptographic signature, installs the new version and restarts. For security, all communications between the sensor client and the cloud are initiated by the client itself. We have also found it useful to have the client open an Secure Shell connection to a fixed computer for maintenance purposes.

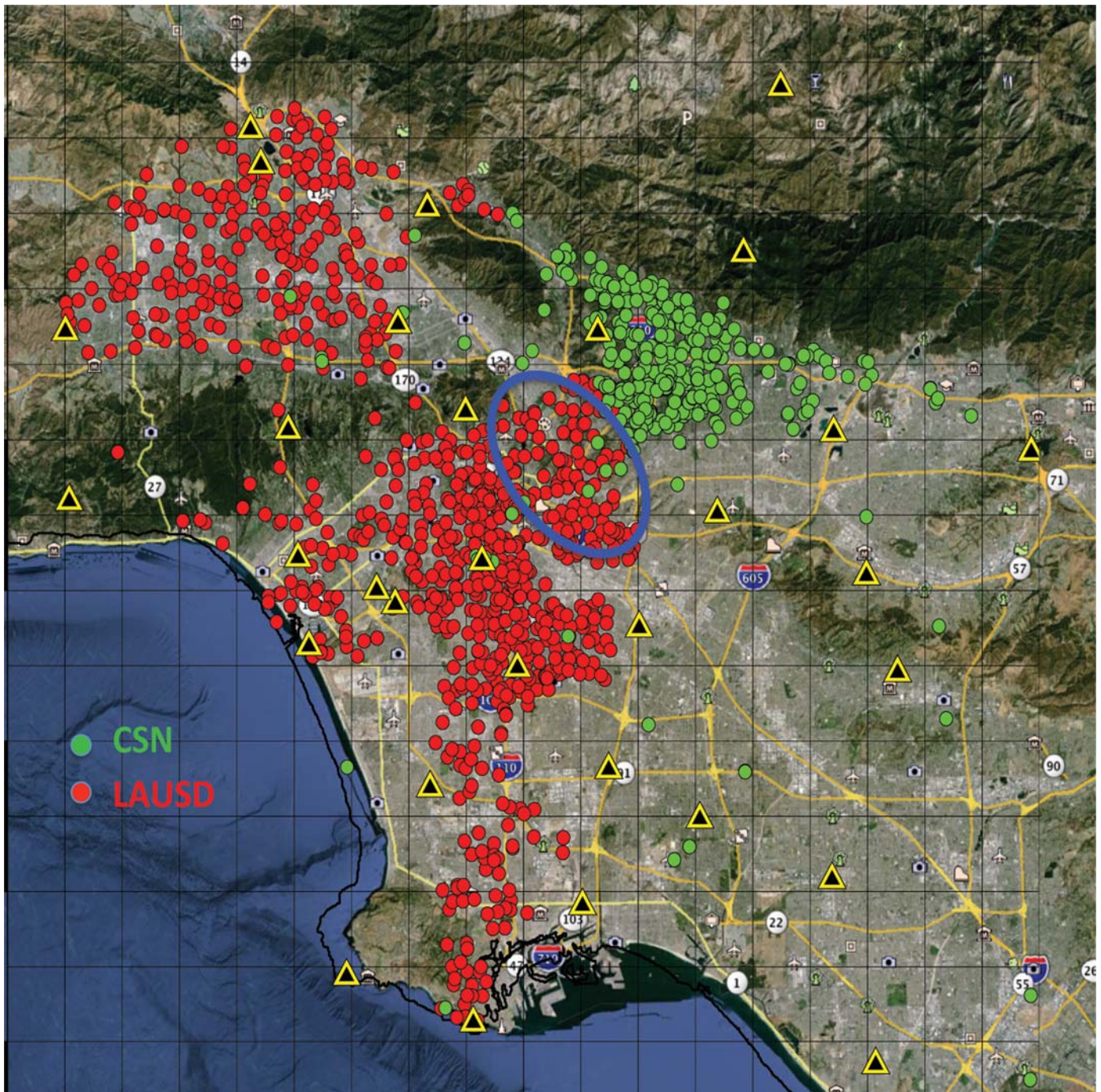
The current client software was originally written in C for efficiency on home computers. Now that most of the sensors (and all future) are stand-alone, the software is being changed to Python to make changes easier and facilitate the development of more sophisticated algorithms.

## CLOUD AND DISTRIBUTED COMPUTING

The computational model that is used by CSN is one of distributed parameter estimation followed by an analysis in the cloud. The measurement of parameters such as pick times, maximum amplitude, apparent frequency, and signal-to-noise levels are made by the sensor systems themselves in near-real time. This information and updates to it are sent directly to the CSN system that is implemented in the Google App Engine cloud. This means that this critical data are sent out of the region before the seismic events have propagated over the Los Angeles basin. This is more robust than sending the data to a site within the reporting region. This aspect of using the cloud will become increasingly important as traditional communication modes (microwave, cellular network, radio) move to the Internet for at least part of their path. The waveform data are sent regularly in small chunks (currently 10 min in length) to the cloud or to a traditional archiving facility.

The detection and ground-motion parameter extraction by the sensor system has a couple of advantages. First, the latency caused by network transport of waveform data is avoided—that is, the parameters are sent as soon as they are determined. Second, the calculations can use the full resolution of the data (the MEM devices are often sampled at higher rates than the waveforms are recorded at). With the sensor currently used in the CSN system the MEM device is sampled at





▲ **Figure 3.** Expansion plans: Over the next 3 years, we plan to expand the network into Los Angeles by placing sensors at the 1200 campuses of the Los Angeles Unified School District (LAUSD), shown by the red dots. The circles show the 100 campuses we have already done. The yellow triangles show the SCSN network. The green dots are the current extent of the CSN.

250 samples per second, but the waveforms are only sent at 50 samples per second. The sensors can also be operated in a couple of different modes to reduce the transmitted waveform data volume—they can be configured to only send waveform data when they are triggered, or they can send data only when requested. This flexibility also allows us to get limited windows of high-sample-rate data if needed. The send-on-request mode

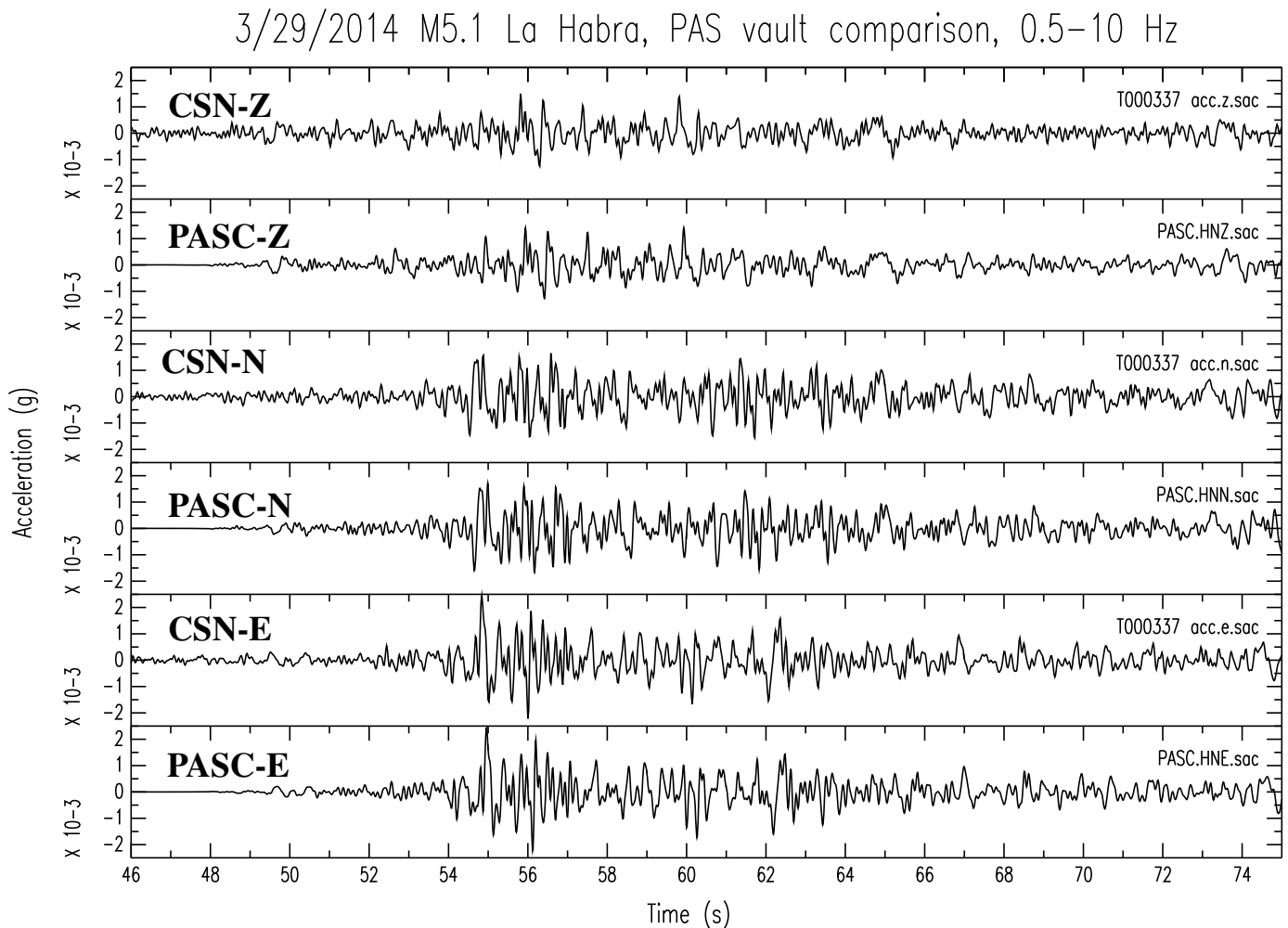
is the procedure we use with cell-phone sensors (described in a later section of this article).

The Cloud software (using Google's App Engine) represents several person years of design and programming effort. The system is highly scalable and robust, and has been shown to cope with very high message rates from hundreds of sensors. The current version running in production uses the Python

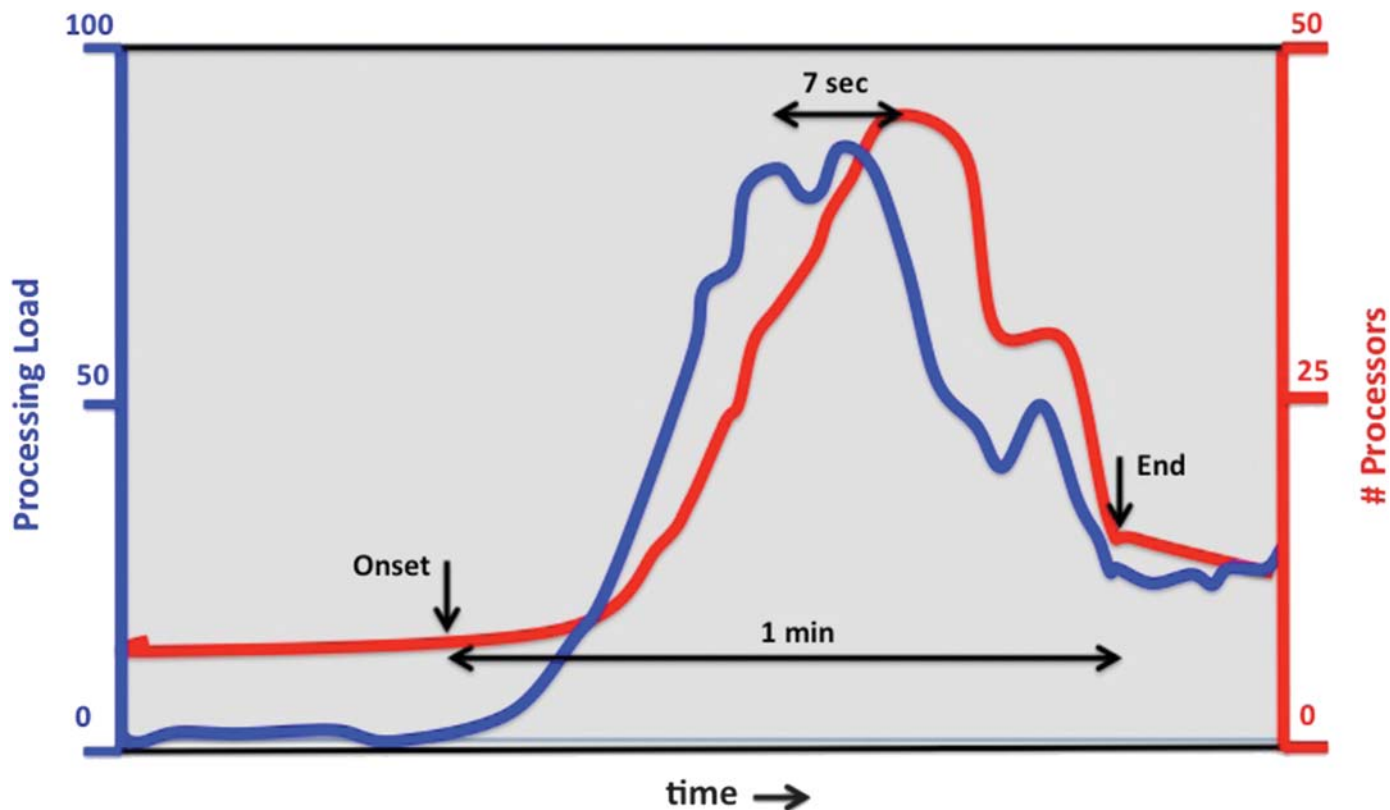




▲ **Figure 4.** Sensor package: The box contains a Phidget Class-C microelectromechanical accelerometer, and a single-board ARM computer with 512 Mbytes of memory, and a battery backup. The package connects to the Internet by wire and is powered by line power.



▲ **Figure 5.** Comparison of Phidget (Class-C) accelerometer with an EpiSensor (Class-A) accelerometer. The recordings of the  $M_w$  5.1 La Habra earthquake are made on the pier of the Pasadena station PASC. The comparison shows the recordings are very similar in both amplitude and waveform.



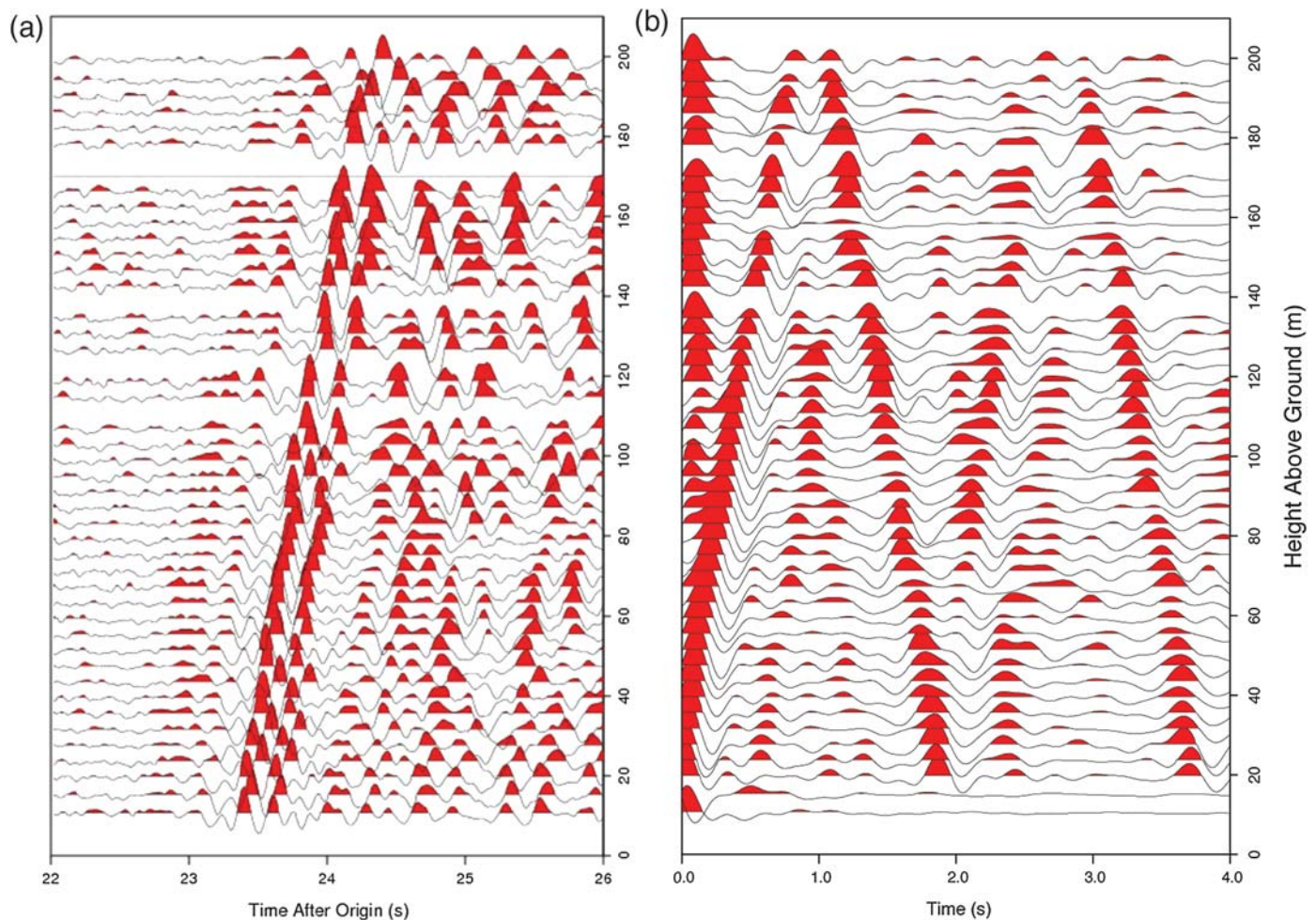
▲ **Figure 6.** Dynamic computational resources of the cloud. The blue curve shows the processing load as measured by the number of picks per second for a large teleseismic event (Sea of Okhotsk, 14 August 2012). The red curve shows the number of processors used in analyzing the event, which varies from 5 at the onset to 45 at the peak. The processor curve lags the load curve by 7 s due to the latency of starting processes. The duration of the enhanced load due to the event is over in about 1 min.

binding to App Engine; a previous version used the Java binding. Experience with the former prompted a complete rewrite of the code in Python to solve several technical issues that were experienced. Most notable among these was significant latency in creating new Java Virtual Machines to handle the increasing bursty loads that occurred during a significant seismic event. Under normal, quiet conditions, the App Engine processes around 20 client requests per second using around eight virtual machine instances (two of which are resident, i.e., always available). During significant events, the request rate from clients has been seen to climb to several thousand per second, for which tens of instances are automatically provisioned to serve the load. With the Python binding, typical request latency is subsecond. An example of the dynamic performance of the cloud-computing environment is shown in Figure 6. The Python binding also facilitates implementing the client on other SBCs such as the Raspberry PI.

The CSN Cloud architecture supports clients that initially connect over the Internet, and Register, a process that catalogs details about the client, such as its location and types of attached sensors, and creates a secure token that is subsequently exchanged between the client and the Cloud to validate messages. All Cloud–client communication is effected by the use of standard hypertext transfer protocol exchanges on port 80, and

all communication is initiated by the client as already described (there is no push from the Cloud to the clients). The HTTP protocol avoids problems with firewalls, and the no-push adds some security to the system (an important issue for all systems that use open networks such as the Internet). The primary concern here is not data theft, but rather deliberate interventions that spoof an earthquake or disable the system. Once registered, clients send periodic heartbeat messages that indicate to the cloud system that they are alive and operating correctly. These messages may also contain offers of waveform data from the client, to which the Cloud may respond with a URL for the client to upload that data. In the case where the client makes a detection, it sends a pick message to the Cloud containing information such as the pick's time and acceleration amplitudes. These messages are aggregated by the Cloud software, which employs geospatial algorithms to associate the picks, and determine whether an event is occurring, its location, and decide if an alert should be generated (Liu, 2013). A more complete description of the communication and the use of the Cloud is given in Olson (2014).

The Cloud software provides a collection of web-based and remote access tools that allow users and administrators to inspect and change parts of the system. For examples, there is a client editor that the owner of a client can use to edit details



▲ **Figure 7.** Earthquake and ambient noise correlations in a 52-story building. (a) The S wave from a M 4.1 earthquake, 70 km from the building. (b) The results for correlating ambient noise using the lowest floor station as a reference. Both show the shear-wave travel at approximately 200 m/s, with reflections from the top and bottom of the building. Panel (a) also shows additional reflections and a high-speed wave that is likely traveling in the steel frame elements of the building.

of the client, there is an administrator map that shows a global view of the entire system, and there are access points that allow administrators to download from the Cloud, and archive, sensor data readings from all clients. In addition, the software supports namespaces, which makes deploying new networks easy. For example, it supports both the southern California-based CSN sensors in one namespace, as well as a set of sensors in Gandhinagar, India, in another namespace, as well as other hazard networks.

It should be emphasized that the Cloud architecture was designed as a general-purpose system, and not specific to any particular type of sensor or client. It could be used equally well for many other types of geospatial event detection, taking data from many other types of sensors, such as environmental detectors (temperature, pressure, various gases and particulate matter, and radiation).

To test whether the cloud system can handle a much larger number of sensors, we included all of the SCSN sensors (an increase of 400 stations) by placing a computer on the wave-bus of SCSN and porting the CSN client code to it. The CSN

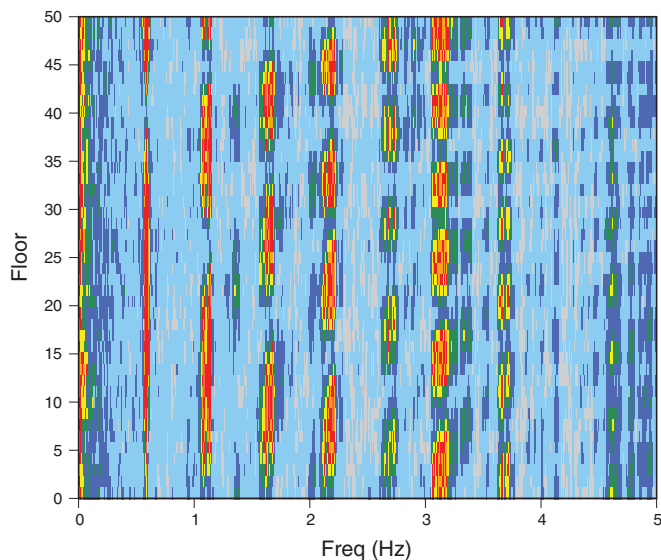
was able to handle this increased traffic easily during earthquakes that shook the entire Los Angeles area in 2014. The test was also useful to demonstrate a possible migration path for traditional networks to the cloud.

The Advanced National Seismic Network ShakeCast system has recently been ported to the cloud (Lin *et al.*, 2014) for some of the same reasons described above. This means that it may be possible for the networks and the product delivery system to communicate directly in the cloud without a traditional computer system involved.

## INTEGRATED FREE-FIELD AND BUILDING MEASUREMENTS

We have placed a number of sensor packages in several buildings in order to make detailed observations of the building motions during an earthquake. We instrumented a 15-story building in downtown Los Angeles with 30 sensors, a 52-story building with 50 sensors, and several other buildings with 2–10



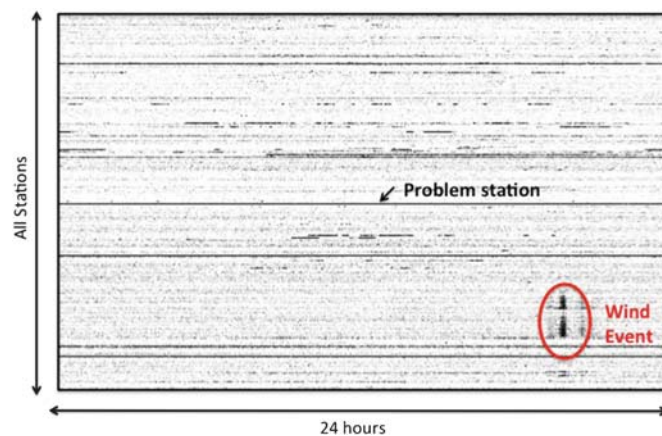


▲ **Figure 8.** Spectra from a 52-story building from the Castaic earthquake. Shown are the floor-by-floor amplitude spectra of the east component of acceleration due to the 4.1 Castaic earthquake (see Fig. 7a for the time-domain plot). Twelve eigenmodes of the building are evident.

sensors. We are in the process of installing an additional 50 sensors in the 52-story building to measure torsional deformation. The density of this instrumentation will be an important source of data for evaluating whether earthquake damage can be detected in structures using sensors. Our standalone sensor packages (orange boxes) are eminently suitable for this purpose because companies are generally reluctant to install third-party software on their own computers.

An example of recordings made using the CSN array in a building is shown in Figure 7. Traveling waves from a 4.1 earthquake some 70 km away can be seen propagating up and down the structure. In addition, waves generated by correlating ambient noise are also shown propagating in the structure (Prieto *et al.*, 2010). This example demonstrates that our NTP-based timing mechanism is sufficiently accurate. Figure 8 shows the spectra recorded from a 4.1 earthquake at a distance of 70 km, and demonstrates that the sensors are able to resolve several vibrational modes of the building.

The dense measurements that we make in buildings are unusual in this field. Although there are some buildings that have reasonably dense sensor arrays (Factor Building at University of California Los Angeles [Kohler *et al.*, 2007]; Atwood Building in Anchorage [Celebi, 2006]), the three-component sensors are not located on every floor. We believe that the sensor density we have deployed will be useful in determining the state of health of a building immediately following an earthquake, and for making an assessment of the overall damage to the building stock of Los Angeles following an earthquake (Kohler *et al.*, 2013, 2014). We also believe that there is merit in having a seamless integration of building and free-field sites. This will be the case in CSN as the building and free-field



▲ **Figure 9.** Display of pick counts for all stations for a 24-hr period. The display shows the level and activity on the network, with darker shading indicating a high number of picks per minute. The horizontal lines indicate malfunctioning sensor systems (usually an intermittent USB connection). The wind event is a 15-min burst of wind that affected the high-rise buildings in downtown Los Angeles. The plot shows the entire network (>500 stations) and the 24 hr beginning at midnight 25 April 2015.

sensors are part of the same network and are recorded and archived in the same fashion.

## MACHINE LEARNING

One of the goals of CSN was to employ some of the ideas and techniques of machine learning to the maintenance and operation of the network. One idea is to weight the quality of a particular sensor measurement by the history of measurements the sensor has produced. At the moment, we are only using simple versions of this technique to detect when sensors are reporting an exceptional number of picks. This is usually caused by the poor placement of the sensor, or a faulty connection between the sensor and computer. In such cases, the sensor is down weighted and marked for repair. An example of one of the monitoring displays is shown in Figure 9. Our plan is to further develop this to allow the sensors to be tuned to the environment in which they are placed, for example, by using sensor-specific thresholds. Other factors such as recent earthquake history and event location could eventually play a role in the machine learning. The hope is that machine-learning techniques will allow the network to automatically take account of underperforming sensors and changing earthquake activity.

## INCLUDING OTHER SENSORS AND NETWORKS

As already noted, the interface between the sensor package and cloud is sufficiently general to allow other devices to be connected into the system. The accelerometers in cell phones are such an example and are discussed in the section below. Other types of seismic sensors can also be included in the system. If they are capable of handling the communications protocol we



have defined then they can be included directly, and if not, our system could likely be easily adapted to accommodate them.

We have also tested multisensor packages that in addition to an accelerometer contain environmental sensors for temperature, pressure, various gases, and radiation. The same communication protocol works and nonseismic data can be simply sent to the same or a different cloud. It is thought that sensors packaged in this fashion would be appealing to private households. Postearthquake measurements of gas concentrations and temperature carried out by a single sensor platform can help detect multiple hazards (Mou, 2013).

## CELL PHONES

From the outset of the CSN project there has been the intriguing idea that the accelerometers in a cell phone could also be used to provide a dense map of acceleration. They present challenges in that the sensors in even the most sophisticated cell phones is at least four times less sensitive than the stationary sensors used in CSN, and they are routinely subject to anthropological motions (e.g., being dropped) that are stronger than the signal we wish them to measure. They are also constantly moving around, which makes static station databases of little use. Nevertheless, they remain an attractive potential resource because there are millions of these phones in constant use in Los Angeles. We have investigated the use of cell phones (Faulkner, 2014; Faulkner *et al.*, 2014) and have designed the CSN system to accommodate them. We have produced an app for Android phones (CSN-Droid) that separates anthropologic motion from earthquake motion and sends the extracted information to the CSN cloud. At the moment we are not using the data in the analysis, but the potential remains a possibility.

## MICROZONATION MAPS

The density of stations is particularly well suited to producing microzonation maps from smaller events in similar fashion as shown in Figure 2. With a dense array this can be done on a spatial scale that can capture the subkilometer variations that are evident in Long Beach (Clayton *et al.*, 2011). The simplest way to create a microzonation map would be to average the peak acceleration maps that are produced for a number of local events, after they have been corrected for geometric spreading and source mechanism (where possible).

## LOCAL EARLY WARNING

We have also been investigating the use of CSN in providing local earthquake early warning (EEW). Local warning differs from network warning, which is the methodology being adopted for the major EEW effort in the western United States, in that it is for users who are within the unable-to-warn zone of network warning. This zone usually extends 5–10 km away from the epicenter and exists primarily due to the delay imposed by the propagation of *P* waves to a minimum set of stations. With local warning, a notice is given if the acceler-

ation exceeds a trigger level at three nearby stations. Only a maximum of a few seconds' warning is possible with this method, and hence its application will be restricted to automatic systems that can react that quickly. The advantage CSN has in this possible product is the density of stations means there are very likely to be three stations in close proximity for the warning to be issued in a timely fashion.

## FOREIGN DEPLOYMENTS

There has also been interest in the CSN technology for networks in foreign countries, particularly ones that cannot afford a conventional seismic network. The low-cost sensors along with the cloud-based central processing make this system economical and simple to set up. As far as the cloud component of the system is concerned, a new network simply becomes a new namespace in the existing system. We are running such a system in conjunction with IIT Gandhinagar in India. The concept could move to another level with an array of clouds to integrate several arrays such as separate regional networks or separate building deployments, but we have not tried this.

## ARRAY OF ARRAYS

One methodology that has emerged in the past few years is the use of small very dense arrays to image phenomena such as microseismicity, tremor, and the dynamics of earthquake rupture. The dense arrays allow the use of advanced beam-forming methods to focus on particular directions, frequencies, and velocities, and an array of such arrays allows the images to be mapped on rupture surfaces. It has been used successfully to map tremor in subduction systems, and is proposed as a method to study rupture physics on strike-slip faults. The CSN sensors are a good candidate for building such arrays because of their low cost and ease of deployment. We have almost completed such an array on the Jet Propulsion Laboratory (JPL) campus (near Caltech) in which 50 sensors will be placed in the free-field locations and 50 in buildings, all in an area that is  $\frac{1}{2}$  km square. We are looking for two more similar sites that will allow the arrays to focus on targets such as the San Andreas and other faults. In the case of the JPL array, it is instrumented at a density that we hope will be equaled in the entire Los Angeles basin in the future, and hence can be used to judge the benefits of this miniature city. ☒

## ACKNOWLEDGMENTS

We thank the Betty and Gordon Moore Foundation and the California Institute of Technology for funding the development of the Community Seismic Network.

## REFERENCES

- Celebi, M. (2006). Recorded earthquake responses from the integrated seismic monitoring network of the Atwood building, Anchorage, Alaska, *Earthq. Spectra* **22**, no. 4, 847–864.

- Clayton, R., T. Heaton, M. Chandy, A. Krause, M. Kohler, J. Bunn, M. Olson, M. Faulkner, M. Cheng, L. Strand, R. Chandy, D. Obenshain, A. Liu, and M. Aivazis (2011). Community seismic network, *Ann. Geophys.* **54**, no. 6, doi: [10.4401/ag-5269](https://doi.org/10.4401/ag-5269).
- Evans, J., R. Allen, A. Chung, E. Cochran, R. Guy, M. Hellweg, and J. Lawrence (2014). Performance of several low-cost accelerometers, *Seismol. Res. Lett.* **85**, 1, 147–158, doi: [10.1785/0220130091](https://doi.org/10.1785/0220130091).
- Faulkner, M. N. (2014). Community sense and response systems, *Ph.D. Dissertation*, California Institute of Technology, <http://resolver.caltech.edu/CaltechTHESIS:04152014-111007328> (last accessed July 2015).
- Faulkner, M., R. Clayton, T. Heaton, K. M. Chandy, M. Kohler, J. Bunn, R. Guy, A. Liu, M. Olson, M.-H. Cheng, and A. Kraus (2014). Community sense and response systems: Your phone as quake detector, *CACM* **57**, no. 7, doi: [10.1145/2622628.2622633](https://doi.org/10.1145/2622628.2622633).
- Kohler, M., T. Heaton, and S. Bradford (2007). Propagating waves in the steel, moment-frame factor building recorded during earthquakes, *Bull. Seismol. Soc. Am.* **97**, no. 4, 1334–1345.
- Kohler, M. D., T. H. Heaton, and M. H. Cheng (2013). The community seismic network and quake-catcher network: Enabling structural health monitoring through instrumentation by community participants, *Proceedings of the SPIE Smart Structures/Non-destructive Evaluation Conference*, San Diego, California, 10–14 March.
- Kohler, M. D., T. H. Heaton, M. H. Cheng, and P. Singh (2014). Structural health monitoring through dense instrumentation by community participants: The community seismic network and quake-catcher network, *10th U.S. National Conference on Earthquake Engineering (NCEEE)*, Anchorage, Alaska, 21–25 July.
- Lin, K.-W., D. Wald, and L. Turner (2014). ShakeCast User Guide (Draft), *U.S. Geol. Surv. Document (ShakeCast\_Manual\_Link)*.
- Liu, A. H.-W. (2013). Sensor networks for geospatial event detection: Theory and applications, *Ph.D. Dissertation*, California Institute of Technology, <http://resolver.caltech.edu/CaltechTHESIS:06062013-224746692>.
- Mou, J. (2013). Situation awareness application, *Master's Thesis*, California Institute of Technology, <http://resolver.caltech.edu/CaltechTHESIS:06272013-211013400>.
- Olson, M. J. (2014). Cloud computing services for seismic networks. *Ph.D. Dissertation*, California Institute of Technology, <http://resolver.caltech.edu/CaltechTHESIS:08242013-182604077>.
- Prieto, G. A., J. F. Lawrence, A. I. Chung, and M. D. Kohler (2010). Predicting earthquake response of civil structures from ambient noise, *Bull. Seismol. Soc. Am.* **100**, 2322–2328, doi: [10.1785/0120090285](https://doi.org/10.1785/0120090285).
- Wald, D. J., K. Lin, B. Worden, and L. Turner (2006). ShakeCast: Facilitating the use of ShakeMap for post-earthquake decision-making and response within Caltrans and other critical lifeline communities, *Proc. of the Fifth National Seismic Conference on Bridges & Highways*, San Francisco, California, September 2006.
- Wald, D. J., V. Quitoriano, T. H. Heaton, H. Kanamori, C. W. Scrivner, and B. C. Worden (1999). TriNet “ShakeMaps”: Rapid generation of peak ground-motion and intensity maps for earthquakes in southern California, *Earthq. Spectra* **15**, no. 3, 537–556.

*Robert W. Clayton*  
*Richard Guy*  
*Seismological Laboratory*  
*Caltech, MC 252-21*  
*Pasadena, California 91125 U.S.A.*  
*clay@gps.caltech.edu*

*Thomas Heaton*  
*Monica Kohler*  
*Civil and Mechanical Engineering*  
*Caltech, MC 104-44*  
*Pasadena, California 91125 U.S.A.*

*Mani Chandy*  
*Computer Science*  
*Caltech, MC 305-16*  
*Pasadena, California 91125 U.S.A.*

*Julian Bunn*  
*Center for Data-Driven Discovery*  
*Caltech, MC 158-79*  
*Pasadena, California 91125 U.S.A.*

Published Online 5 August 2015