Distributed Sensor Network for meteorological observations and numerical weather Prediction Calculations

Ádám Vas
University of Debrecen, Faculty of Science and Technology,
Department of Meteorology
Debrecen, Hungary
vas.adam@inbox.com

Ádám Fazekas
University of Debrecen, Faculty of Informatics,
Department of Informatics Systems and Networks
Debrecen, Hungary
fazekas.adam@inf.unideb.hu

Gábor Nagy, László Tóth
SciTech Műszer Kft.
Debrecen, Hungary
nagyg2000@t-online.hu, laszlo.toth@scitechmuszer.com

***Abstract*—The prediction of weather generally means the solution of differential equations on the base of the measured initial conditions where the data of close and distant neighboring points are used for the calculations. It requires the maintenance of expensive weather stations and supercomputers. However, if weather stations are not only capable of measuring but can also communicate with each other, then these smart sensors can also be applied to run forecasting calculations. This applies the highest possible level of parallelization without the collection of measured data into one place. Furthermore, if more nodes are involved, the result becomes more accurate, but the computing power required from one node does not increase. Our Distributed Sensor Network for meteorological sensing and numerical weather Prediction Calculations (DSN-PC) can be applied in several different areas where sensing and numerical calculations, even the solution of differential equations, are needed.**

***Keywords*— distributed sensor network; meteorological sensing; numerical weather prediction**

I. INTRODUCTION

Meteorology is strongly influenced by scientific and technological developments, which is especially true in the case of weather forecasting. The improvements of thermodynamics helped to recognize that the solution of equations of hydro- and thermodynamics as well as continuity with five initial variables (pressure (mbar), temperature (°K), density (kg/m^3), humidity (%), velocity (m/s)) are feasible to be used for successful weather forecasting calculations [1],[2].

However, the fast collection of weather data reports from wide areas became possible only after the invention of electric telegraph in 1835. Because of the lack of known physical and mathematical models as well as proper computing instruments, forecasting was based on a visual pattern recognition method where a previous similar weather situation and the subsequent one were used to predict the upcoming one. Apparently, finding a perfect analogue for an event in the past had low probability which made this technique difficult to use.

The first weather prediction based on physical equations and numerical calculation was made by Richardson [3] in the first decades of the 20th century. The complexity of numerical weather prediction models started to increase significantly when the Electronic Numerical Integrator and Computer (ENIAC) was announced and the modeling work by J. Charney, R. Fjørtoft and J. von Neumann (CFvN) [4] started. As computers grew in power and capacity, it became possible to run more complex mathematical models involving denser initial data in space and time [2].

Since the number of measurement points and observed parameters nowadays are relatively large and continuously grow, as well as the complexity of the models, the weather forecasting calculations require more and more expensive supercomputers, although modern weather sensors are already equipped with a digital central unit (microprocessor (μP), microcontroller (μC), Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA)), which represents a significant computation power currently utilized "only" for filtering the measured values and sending them to a central computer.

The central computer could be eliminated or greatly simplified with our method of forecast calculations where the nodes are connected to each other through the Internet; therefore, it is not needed to collect data and make the calculation at one location. These nodes can communicate with each other and make numerical finite difference or finite element calculations based on their close and distant neighbors' data. This grid itself forms the sensor network and a distributed computing system, which can be used for solving differential equations without using a central computer, so it can function as a multiprocessor system with extremely high number of computational units, making the highest level of parallelization of the calculations possible.

In such a system by increasing the number of nodes for getting better covered field and therefore more accurate calculations the computing power required by one node does not increase, since they use their close and distant neighbors' data only (Fig. 1). The nodes can also be placed on the board of moving vehicles i.e. ships and airplanes for monitoring even the vertical parameters.

Our work of developing DSN-PC started five years ago [5],[6],[7],[8],[9],[10], and here we present the latest results.

II. THE ENIAC FORECAST AND ITS IMPLEMENTATION ON DSN-PC

Earlier it was shown that our μC based networked sensor system is able to do both meteorological sensing and making calculations simultaneously. These calculations involved determining isotherm lines, temperature gradient vectors and solving the differential equation of heat transfer by applying finite difference method (FDM) based on communication between nodes [9]. For demonstrating the functionality and capability of our method and networked system also for simple weather prediction calculations, the well-known first successful 24 hour weather prediction algorithm for large-scale motions in barotropic atmosphere, developed by CFvN and run on the ENIAC, was applied. Here the CFvN solution of finite difference linearized barotropic vorticity equations is also shown. Although it was used for calculating predictions for the 500 mbar geopotential heights and its input data set differs from that of our DSN-PC stations can provide (ground level pressure (mbar), temperature (°C) and relative humidity (%)) at the present, by applying CFvN calculations we could show the capability of our system of realizing such kind of calculations. We are working on development of new and more sophisticated weather prediction algorithms for DSN-PC.

In CFvN calculations the spherical Earth was mapped by polar stereographic projection onto a rectangular plane (Fig. 2) with sides of L_x and L_y, then a rectangular grid was created and defined by the coordinates of

$$x = \frac{L_x}{p} i, \quad y = \frac{L_y}{q} j \tag{1}$$

$$i = 0, ..., p; \quad j = 0, ..., q$$

where

$$\frac{L_x}{p} = \frac{L_y}{q} = \Delta s \tag{2}$$

and Δs=736 km at the North Pole (494 km at 20°N), $\Delta\lambda$=8° of longitude at φ=45°N.

During the following calculations, according to CFvN, the next notations are used:

$$r = \frac{\cos\varphi}{1+\sin\varphi} \tag{3}$$

$$m = -2\frac{dr}{d\varphi} = \frac{2}{1+\sin\varphi} = 1 + r^2 \tag{4}$$

$$f = 2\Omega\sin\varphi = 2\Omega\frac{1-r^2}{1+r^2} \tag{5}$$

$$h = \frac{gm^2}{f} \tag{6}$$

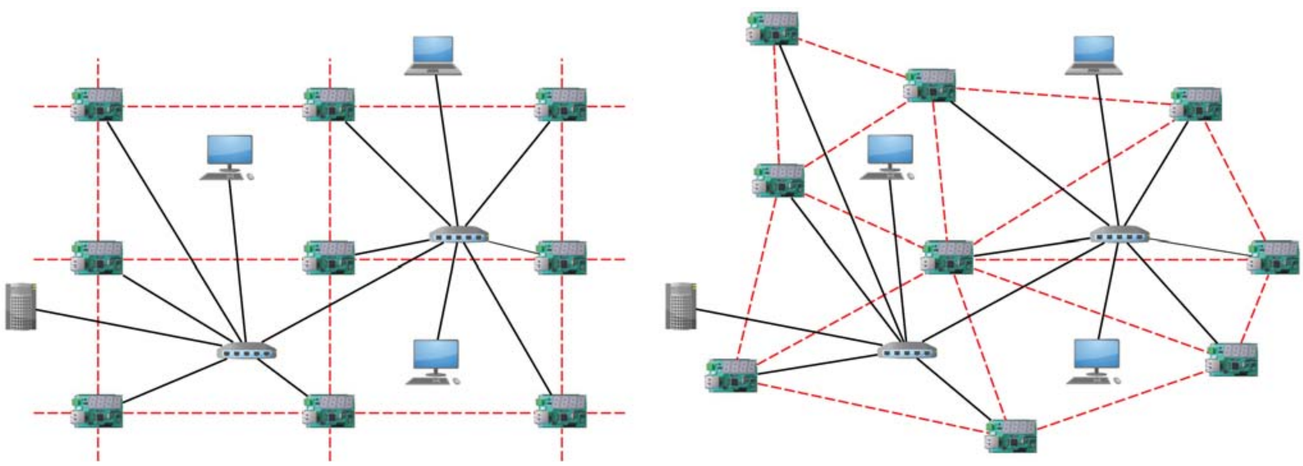

Fig. 1. The conception of DSN-PC sensor network, which forms the necessary grid for numerical calculations by using the nodes of the networked sensor stations not only for measurements and collection of data but for making numerical calculations in a distributed way. The networked weather stations (nodes) are able to communicate with their close and distant neighbors through the Internet and solve weather prediction calculations. In the present work we applied FDM on a rectangular grid (Left) but the calculations can also be done by Finite Elements Method (FEM) on irregular grid (Right).

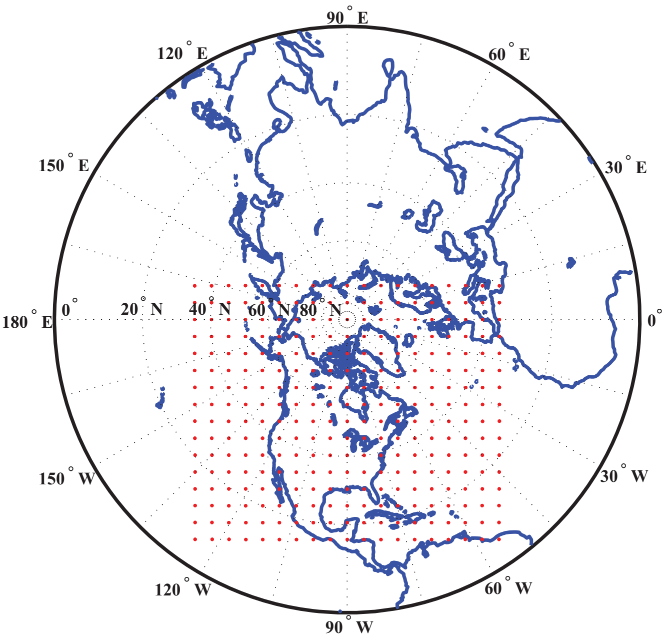

Fig. 2. Stereographic projection of the northern hemisphere and the 16×19 rectangular grid structure for CFvN calculations.

where

- φ is the geographical latitude.
- r is the distance from the pole on the stereographic projection map. The radius of the equator on the map was chosen as the unit of distance.
- m is the magnification factor of the polar stereographic projection.
- f is the Coriolis parameter.
- Ω is the angular velocity of Earth's rotation ($\Omega=2\pi(24\cdot 60\cdot 60)^{-1}$ rad/s).
- g is the gravitational constant ($g=9.80665$ m/s^2).

The finite difference vorticity equations of CFvN that have been implemented in our system are the following:

$$\eta_{ij} = h_{ij}\xi_{ij} + f_{ij} \tag{7}$$

$$\frac{\partial \xi_{ij}}{\partial t} = J_{ij}(\eta, z) \tag{8}$$

$$\Delta_{ij}\left(\frac{\partial z}{\partial t}\right) = \frac{\partial \xi_{ij}}{\partial t} \tag{9}$$

where

- $$\xi_{ij} = \Delta z_{ij} \tag{10}$$

  This equation is used for the calculation of ξ_{ij} in the first step of the iteration while in the following steps (24) is used.

- z is the geopotential height of the 500 mbar level. The initial data were taken from the 500 mbar analyses of the U.S. Weather Bureau [4]. In our case the interpolated data for the initial values of z at the grid points were taken from [11].
- η is the absolute vorticity.
- the h_{ij} and f_{ij} parameters are independent of t and have to be calculated only once for each node.
- the Poisson equation:

$$\Delta_{ij}\left(\frac{\partial z}{\partial t}\right) = \frac{1}{\Delta s^2}\left[\left(\frac{\partial z}{\partial t}\right)_{i+1,j} + \left(\frac{\partial z}{\partial t}\right)_{i-1,j} + \left(\frac{\partial z}{\partial t}\right)_{i,j+1} + \left(\frac{\partial z}{\partial t}\right)_{i,j-1} - 4\left(\frac{\partial z}{\partial t}\right)_{ij}\right] \tag{11}$$

- the Jacobi operator:

$$J_{ij}(\eta, z) = \frac{1}{4(\Delta s)^2}\left[\left(\eta_{i+1,j} - \eta_{i-1,j}\right)\left(z_{i,j+1} - z_{i,j-1}\right) - \left(\eta_{i,j+1} - \eta_{i,j-1}\right)\left(z_{i+1,j} - z_{i-1,j}\right)\right] \tag{12}$$

The boundary conditions for $\left(\frac{\partial z}{\partial t}\right)$ are:

$$\left(\frac{\partial z}{\partial t}\right)_{0j} = \left(\frac{\partial z}{\partial t}\right)_{pj} = \left(\frac{\partial z}{\partial t}\right)_{i0} = \left(\frac{\partial z}{\partial t}\right)_{iq} = 0 \tag{13}$$

$$i = 0, \ldots, p; \quad j = 0, \ldots, q$$

which means the geopotential height is considered to be constant at the boundaries during the term of prediction. However, the boundary values of $\left(\frac{\partial \xi}{\partial t}\right)_{ij}$ are different in cases when the fluid is entering or leaving the investigated area. In the first case $\left(\frac{\partial \xi}{\partial t}\right)_{ij} = 0$ and in the second case $\left(\frac{\partial \xi}{\partial t}\right)_{ij}$ can be calculated by linear extrapolation from the interior value of η and z:

$$i = 0: z_{0,j+1} - z_{0,j-1}\begin{cases} \geq 0: \left(\frac{\partial \xi}{\partial t}\right)_{0j} = 2\left(\frac{\partial \xi}{\partial t}\right)_{1j} - \left(\frac{\partial \xi}{\partial t}\right)_{2j} \\ < 0: \left(\frac{\partial \xi}{\partial t}\right)_{0j} = 0 \end{cases} \tag{14}$$

$$i = p : z_{p,j-1} - z_{p,j+1} \begin{cases} \geq 0 : \left(\frac{\partial \xi}{\partial t}\right)_{pj} = 2\left(\frac{\partial \xi}{\partial t}\right)_{p-1,j} - \left(\frac{\partial \xi}{\partial t}\right)_{p-2,j} \\ < 0 : \left(\frac{\partial \xi}{\partial t}\right)_{pj} = 0 \end{cases} \tag{15}$$

$$\mathrm{j = 0 : z_{i-1,0} - z_{i+1,0}} \begin{cases} \geq 0 : \left(\frac{\partial \xi}{\partial \mathrm{t}}\right)_{\mathrm{i0}} = 2\left(\frac{\partial \xi}{\partial \mathrm{t}}\right)_{\mathrm{i1}} - \left(\frac{\partial \xi}{\partial \mathrm{t}}\right)_{\mathrm{i2}} \\ < 0 : \left(\frac{\partial \xi}{\partial \mathrm{t}}\right)_{\mathrm{i0}} = 0 \end{cases} \tag{16}$$

$$\mathrm{j = q : z_{i+1,q} - z_{i-1,q}} \begin{cases} \geq 0 : \left(\frac{\partial \xi}{\partial \mathrm{t}}\right)_{\mathrm{iq}} = 2\left(\frac{\partial \xi}{\partial \mathrm{t}}\right)_{\mathrm{i,q-1}} - \left(\frac{\partial \xi}{\partial \mathrm{t}}\right)_{\mathrm{i,q-2}} \\ < 0 : \left(\frac{\partial \xi}{\partial \mathrm{t}}\right)_{\mathrm{iq}} = 0 \end{cases} \tag{17}$$

$$i = 0, ..., p; \quad j = 0, ..., q$$

The corner points are not taken into consideration.

In a worldwide network of DSN-PC the boundary condition may not necessarily be defined in the above way since the nodes of the network will provide them automatically through their directly measured and calculated data.

The explicit solution for $\left(\frac{\partial z}{\partial t}\right)_{ij}$ shown in [4] that was more suitable for the ENIAC calculation procedure, and is given by four one-dimensional Fourier transforms:

$$\left(\frac{\partial z}{\partial t}\right)_{ij} = -\frac{(\Delta s)^2}{pq} \sum_{l=1}^{p-1} \sum_{m=1}^{q-1} \sum_{r=1}^{p-1} \sum_{s=1}^{q-1} \left(\sin^2 \frac{\pi l}{2p} + \sin^2 \frac{\pi m}{2q}\right)^{-1} \left(\frac{\partial \xi}{\partial t}\right)_{rs} \sin\frac{\pi l r}{p} \sin\frac{\pi m s}{q} \sin\frac{\pi l i}{p} \sin\frac{\pi m j}{q} \tag{18}$$

However, in our system this method would require from each DSN-PC node to collect data from all the other nodes of the investigated area, which would be difficult and would require large memory and processor capacities. Therefore, in our system an iterative solution, calculated in a distributed way by the µC based nodes, was applied, that consists of the next steps:

$$z_{ij} \xrightarrow{(10)} \xi_{ij} \xrightarrow{(3)-(7)} \eta_{ij} \xrightarrow{(12)} J_{ij}(\eta, z) \xrightarrow[\text{for the whole field}]{\text{n times iteration of (8),(9),(11)}} \left(\frac{\partial z}{\partial t}\right)_{ij} \tag{19}$$

At the first step on the border nodes ξ_{ij} are extrapolated similarly to (14)-(17):

$$i = 0: \quad \xi_{0j} = 2\xi_{1j} - \xi_{2j} \tag{20}$$

$$i = p: \quad \xi_{pj} = 2\xi_{p-1,j} - \xi_{p-2,j} \tag{21}$$

$$j = 0: \quad \xi_{i0} = 2\xi_{i1} - \xi_{i2} \tag{22}$$

$$j = q: \quad \xi_{iq} = 2\xi_{i,q-1} - \xi_{i,q-2} \tag{23}$$

During this procedure the nodes are communicating with their neighbors, exchanging data and solving the differential equation by applying finite difference scheme (19). Since the iterative solution of the finite difference scheme can be explained as the propagation of disturbances in the field of the investigated area, the data change and the finite difference iteration parts of the calculation for the different nodes in our case can happen not only in a regular way, but even randomly.

When $\left(\frac{\partial z}{\partial t}\right)_{ij}$ and $\left(\frac{\partial \xi}{\partial t}\right)_{ij}$ are determined from (11) and (12), the time extrapolation can be done using the next equations [4]:

$$\xi_{ij}(t_0 + \Delta t) = \xi_{ij}(t_0) + \Delta t \left(\frac{\partial \xi}{\partial t}\right)_{ij}(t_0)$$

$$\xi_{ij}(t + \Delta t) = \xi_{ij}(t - \Delta t) + 2\Delta t \left(\frac{\partial \xi}{\partial t}\right)_{ij}(t) \tag{24}$$

$$z_{ij}(t_0 + \Delta t) = z_{ij}(t_0) + \Delta t \left(\frac{\partial z}{\partial t}\right)_{ij}(t_0)$$

$$z_{ij}(t + \Delta t) = z_{ij}(t - \Delta t) + 2\Delta t \left(\frac{\partial z}{\partial t}\right)_{ij}(t) \tag{25}$$

These calculations have to be repeated 24 times for the whole forecast period and the solution can be found iteratively by solving for $\left(\frac{\partial z}{\partial t}\right)_{ij}$ and extrapolating the motion forward in time with the suitable boundary conditions. The flowchart of the algorithm is shown in Fig. 3. Initial steps including the calculation of the geographical parameters are omitted for better readability.

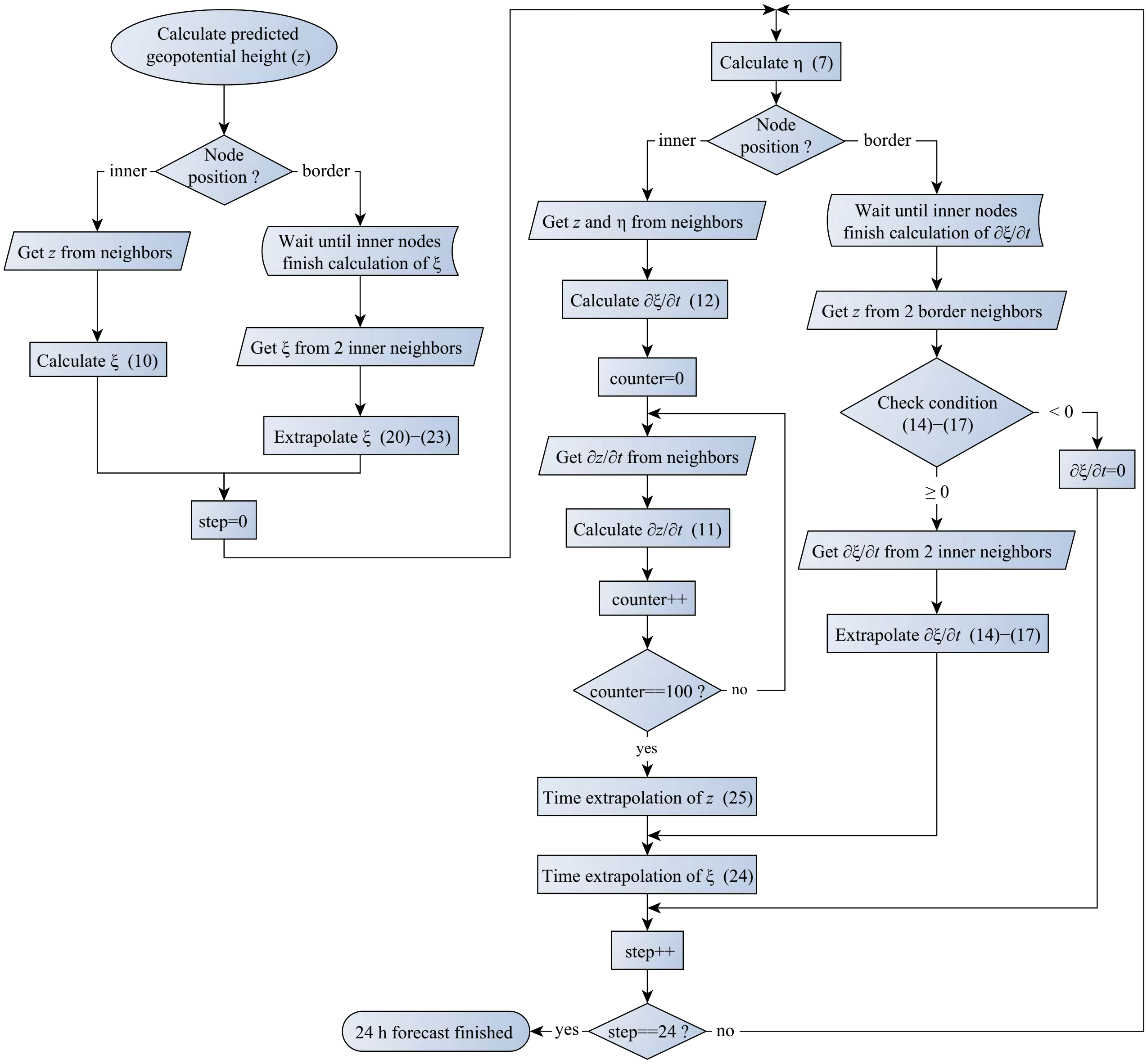

Fig. 3. The algorithm of the calculations running on the real and simulated boards.

III. DESCRIPTION OF THE PRESENT PROTOTYPE DSN-PC SYSTEM

A. Hardware design

The present DSN-PC system has three main parts. Firstly, the µC based hardware (Fig. 4), which is equipped with temperature (°C), humidity (%) and pressure (mbar) sensors, 10Base-T Ethernet connection and the necessary power supply circuit. Secondly, a software running on the µC responsible for controlling the measurements, network connections and making the necessary calculations. Finally, since at present stage only three physical test boards are used, a software simulated test environment was necessary to be built for adding virtual nodes to the system. More details of the board and the used software components can be found in [9].

It is important to highlight that the present instruments are not intended to be traceable or form calibrated nodes for real weather forecasting, but rather demonstration devices capable of measuring basic parameters with a known error and executing the necessary calculations. The reliable sensor data filtering and calibration is out of the scope of our present experiments, however, for a next level of field tests, it is needed to address this issue to make the results comparable with currently used weather stations.

In the test environment we established a simple Ethernet network in which 3 µC based nodes and a PC communicate through an Ethernet switch (Fig. 5). The other nodes are simulated on the PC [9].

Fig. 4. Our prototype DSN-PC station [9].

B. Software on the DSN-PC panel

For embedded software development we used MPLAB X, an integrated development environment from Microchip and the TCP/IP Stack library, which includes the source code of many network communication protocols.

Based on that library, the most important parts of our software are the UDP servers and clients responsible for the communication between the stations over the network. UDP is currently preferred to TCP because the PIC18F67J60 μCs only have 4 kB of data memory, and UDP requires significantly less memory.

Previously three calculation methods have been implemented such as temperature gradients, isotherms and the solution of heat transfer equation by FDM [9]. Since we do not have our own forecast algorithm for DSN-PC yet, but we wanted to show its capability making not only simple calculations and solving the simple heat transfer equation but solving more difficult differential equations related to weather prediction calculations; the well-known first successful method of CFvN was implemented.

As soon as the final system is available, an efficient way of firmware (FW) updates is going to be essential, especially in the early phase of the development. The FW update mechanism needs a boot loader, which is able to download a new version of FW if available and apply the update. It is important to guarantee that a corrupted communication cannot prevent the boot loader from getting back to normal operation. Microchip's Library for Applications contains a demo application called Internet Bootloader which fits our requirements. Based on that demo a customized version was made, which is integrated with a symmetric encryption method based on XTEA cipher for increased security. The boot loader application is based on a TFTP server, listening for incoming connection on UDP port 69. In the case of an incoming connection the server waits for incoming data from the client and downloads the new FW. For easier usability we are working on a TFTP client based version, so it will be able to automatically check for a new version on an update server, download the new FW and reprogram the μC.

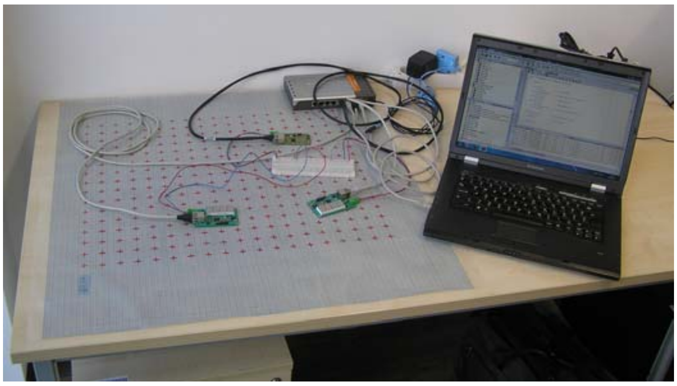

Fig. 5. The test environment with 3 real weather stations connected to each other and to the simulated environment through an Ethernet switch. For better understanding, the 16×19 rectangular grid (Fig. 2) is represented on grid paper.

IV. RESULTS

The initial observed data of the 500 mbar geopotential height is represented on the map of Fig. 6, and the map of the 24 hour later observed data is shown in Fig. 7. The necessary data in both cases are the same as they were in the original CFvN calculations and are taken from [11]. The original solution on the ENIAC [4] as well as P. Lynch's demonstrations on MATLAB and mobile phone [11],[12],[13] all apply (18). However, in our DSN-PC system, the application of the iterative method (19) is more suitable, since for one node it is sufficient to know the necessary data of their close and distant neighbors, and do not need to collect and store the whole data set from all the other nodes of the investigated area. The result of the 24 hour geopotential height CFvN prediction calculation by our DSN-PC system, which applies (19) can be seen in Fig. 8.

It agrees well with the previous results (Fig. 9) [4],[11],[12],[13], and the difference between those and ours are negligible (Fig. 10), which shows that our method and a DSN-PC-like system may have to be applied for making more sophisticated weather prediction calculations in the future.

V. CONCLUSION

On the base of our conception we succeeded to design and develop instruments and architecture, as well as to build the prototype of a meteorological Network for Sensing and numerical weather Prediction Calculations (DSN-PC). As it was shown earlier, the present system is capable of sensing temperature, pressure and humidity besides calculating isotherms, temperature gradients and solving simple numerical differential equations like heat transfer by the method of finite differences [9]. The system is working through the Internet, where the nodes receive the measured parameters from their close and distant neighbors and make the necessary calculations. Here the conception and structure of DSN-PC was described, furthermore the implementation of the well-known finite-difference vorticity equations of CFvN on our system is shown to demonstrate its capability of making simple weather prediction calculations in a test environment.

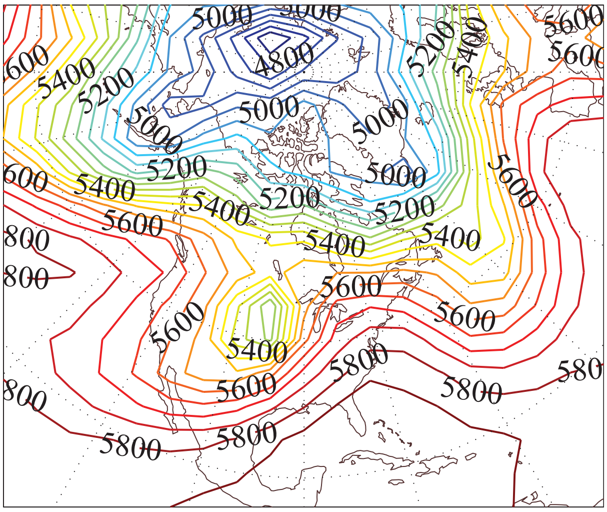

Fig.6. The initial observed 500 mbar geopotential height weather map as the input of CFvN calculations [4],[11],[12].

Although the present system consists of only 3 stations and the other nodes of the applied virtual field are simulated, it was possible to investigate several different hardware configurations, the applied algorithms and software as well as to get sufficient experiences for further developments of our DSN-PC system. Some of the further possibilities are:

- developing modern suitable weather prediction algorithms (i.e. method of finite elements with irregular net) for DSN-PC.
- applying it in short-term regional weather forecast or urban climate and heat island analysis.
- increasing the order of calculations.
- involving vertical data in the calculations.

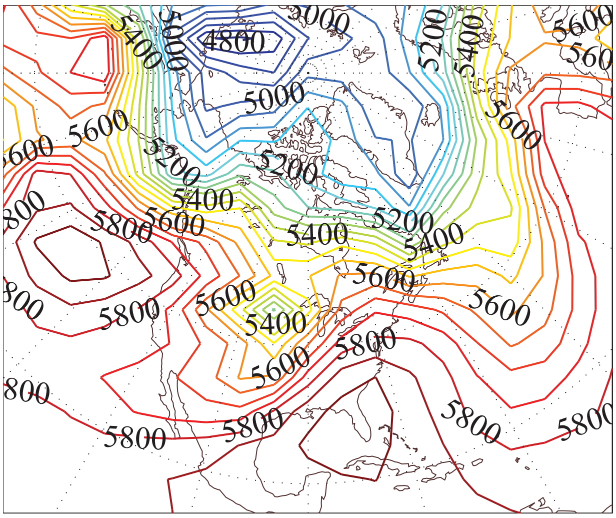

Fig. 8. The result of 24 hour 500 mbar geopotential height CFvN weather prediction calculation by using the iterative solution (19) and our prototype DSN-PC system in the test environment.

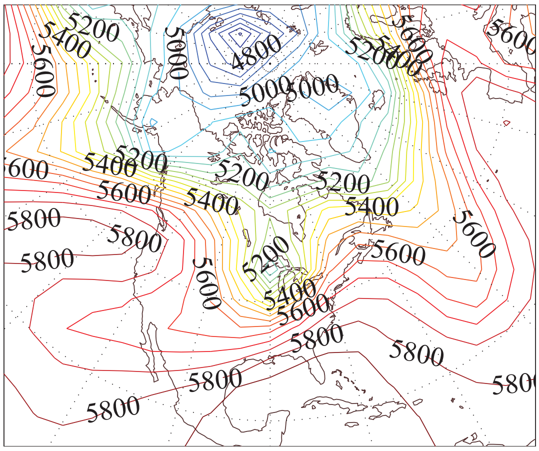

Fig. 7. The final observed 500 mbar geopotential height weather map (24 hours later) [4],[11],[12].

- developing new stations with extended sensor capabilities, increased measurement accuracy and improved networking functions such as applying GPS receiver and long range radiowave or GSM connection.
- at the present a central computer is needed to help the nodes to find their neighbors when a new node is introduced to the system, that could be eliminated by applying a neighbor discovery algorithm running on the stations.
- using the system not only for weather sensing and prediction but also for other FDM/FEM systems.
- applying forecast calculation methods on mobile phones equipped with i.e. air pressure sensors.

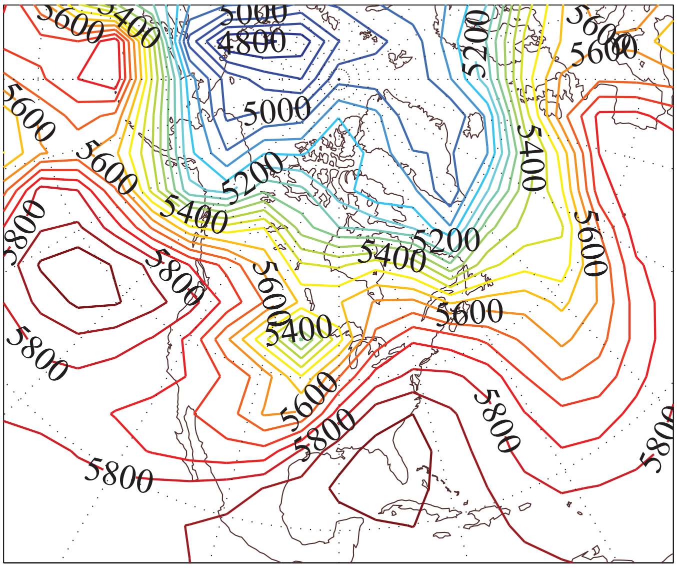

Fig. 9. The result of 24 hour 500 mbar geopotential height CFvN weather prediction calculation by using (18) and P. Lynch's open source MATLAB code [11],[12].

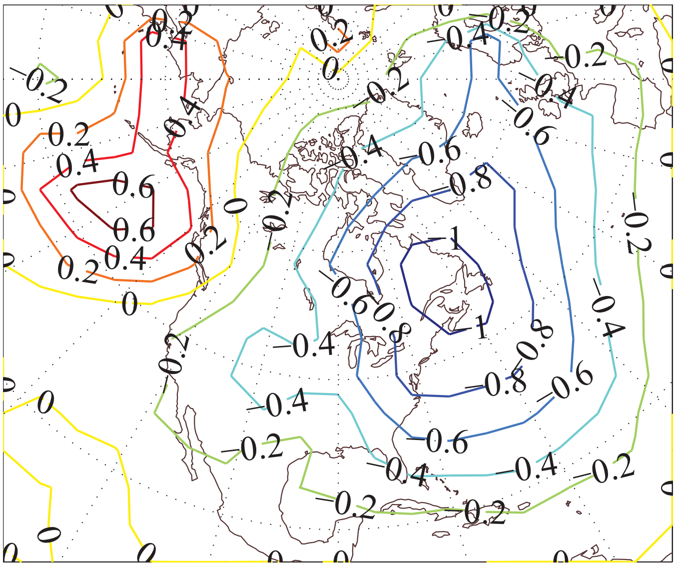

Fig. 10. The differences between the results of 24 hour 500 mbar geopotential height CFvN weather prediction calculation by using (18) on a single computer [11],[12] and the iterative solution of (19) by our prototype DSN-PC system in a test environment.

The DSN-PC practically simulates the interactions of the real world where the stations are placed in the nodes of a regular or irregular net, stretched physically on the surface of the Earth for sensing, communicating with each other and making numerical prediction calculations. It can quasi simulate reality and make predictions, since the velocity of the spreading of disturbances during the regular or randomly made steps of numerical calculations through the network is faster than it is in the case of the real world; that makes the system being capable of prediction calculations.

Since there are a lot of possibilities for further developments in many fields (meteorology, electronics, embedded hardware/software systems, networking, mathematics, physics, …) of interest related to DSN-PC and it can be applied in different research and university programs, we make the

- circuit diagram of our present DSN-PC station
- the embedded software of measurements, Ethernet communication, existing calculations etc.
- the Java software of DSN-PC simulation environment

freely available on http://www.scitechmuszer.com for non-commercial developments and university programs; furthermore, we are open for future cooperation.

Acknowledgment

The authors wish to thank SciTech Műszer Kft. for providing the possibility to use their facilities and for supporting our work financially.

References

[1] F.G. Shuman, “Weather prediction,” U.S. Department of Commerce National Oceanic and Atmospheric Administration, National Weather Service, Washington, Office Note 198, 1979.

[2] P. Lynch, “The origins of computer weather prediction and climate modeling”, J. Computational Physics, vol. 227, no. 7, pp. 3431-3444, 2008.

[3] L.F. Richardson, Weather prediction by numerical process, Cambridge: Cambridge University Press, 1922.

[4] J.G. Charney, R. Fjørtoft and J. von Neumann, “Numerical integration of the barotropic vorticity equation," Tellus, vol. 2, pp. 237-254, 1950.

[5] B. Simon and G. Pásztor, “Mikrokontrolleren alapuló időjárás előrejelző hálózat,” B.S. thesis, Faculty of Informatics, University of Debrecen, Debrecen, Hungary, 2010.

[6] B. Kulcsár and D. Virág, “Mikrokontrolleren alapuló időjárás előrejelző hálózat,” B.S. thesis, Faculty of Informatics, University of Debrecen, Debrecen, Hungary, 2011.

[7] B. Ferenc, “Rendszerfrissítő alkalamzás mikrokontrolleren alapuló időjárás-előrejelző rendszerhez,” B.S. thesis, Faculty of Informatics, University of Debrecen, Debrecen, Hungary, 2012.

[8] G. Csizmadia, “Hálózati alkalmazás fejlesztése meteorológiai rendszerhez,” B.S. thesis, Faculty of Informatics, University of Debrecen, Debrecen, Hungary, 2012.

[9] Á. Vas, Á. Fazekas, B. Lehotai, G. Nagy and L. Tóth, “Microcontroller-based network for meteorological sensing and weather forecast calculations,” Carpathian Journal of Electronic and Computer Engineering, vol. 5, pp. 139-142, 2012.

[10] Á. Vas, “Mikrokontroller-alapú időjárás-előrejelző hálózat építése,” M.S. thesis, Faculty of Informatics, University of Debrecen, Debrecen, Hungary, 2013.

[11] P. Lynch. (2008, February). The ENIAC Integrations [Online]. Available: http://mathsci.ucd.ie/~plynch/eniac

[12] P. Lynch, “The ENIAC forecasts: A re-creation,” Bull. Amer. Meteor. Soc., vol. 89, no. 1, pp. 45-55, 2008.

[13] P. Lynch and O. Lynch, “Forecasts by PHONIAC,” Weather, vol. 63, no. 11, pp. 324-326, 2008.