Articol

Teoria grafurilor aplicate în rețele de senzori wireless

Bătrînuţ Silviu

Ingineria Sistemelor, Facultatea de Automatică şi Calculatoare, Universitatea Politehnică Timişoara, Timişoara ,România

[batranutsilviu@yahoo.com](mailto:batranutsilviu@yahoo.com)

Publicat: 11.04.2019

**Rezumat:**

Această lucrare prezintă şi explică cum se aplică teoria grafurilor, împreună cu unii algoritmi în reţelele de senzori wirless. Cum va arată viitorul? Cum se va îmbunătăţi viaţa omului în viitorul apropiat? Răspunsul parţial la aceste întrebări îl veţi afla în acest articol. Vom studia împreună grafurile, tipurile acestora, algoritmii ce pot să se aplice pe acestea. Iar pe final vom afla şi cum pot influenţa tehnologia, serviciile și producția.

**Cuvinte cheie:**

Graf, Arbore, Algoritm, Wireless, Senzor, Internet of Things

1. **Introducere**

În ultimii ani s-a dezvoltat foarte mult domeniul Internet of Things și cel al rețelelor de senzori wireless. Viitorul tehnologiei pare să țină cont de aceste domenii. Utiltatea lor este variată, de la monitorizarea calității aerului până la uz militar. În această lucrare se va discuta importanța acestor rețele.

Pentru început vom aborda teoria tehnică a grafurilor și a tehnologiei Wi-Fi, apoi vom discuta despre rețelele de senzori wireless și cum se aplică teoriile anterioare în aceste rețele, iar în final vom trage niște concluzii.

1. **Aspecte teoretice**
   1. **Teora grafurilor**

Un graf este o structură de date care conține muchii, noduri și poate fi orientat sau ne-orientat. Această structură este menită să reprezinte un sistem, în cazul nostru va fi un sistem de senzori wireless. Grafurile ne-orientate nu țin cont de sensul transmiterii informației, dar cei orientați țin seama.

Un arbore este un graf ne-orientat care nu conține cicluri și în care, oricare 2 noduri sunt conectate de un singur drum.

Alte noțiuni implicate în teoria grafurilor sunt conexitatea și componentele conexe. Un graf este conex dacă există cel puțin un drum între oricare două noduri distincte. O componentă conexă este acea componentă, formată din unu sau mai multe noduri care nu au nici-o conexiune cu o altă componentă.

O noțiune importantă în teoria grafurilor este cea de biconexitate. Un graf este biconex dacă eliminându-se oricare nod, acesta va rămâne conex. Caracteristica aceasta este vitală pentru integritatea unui graf. Dacă unul din noduri nu funcționează corespunzător și graful este biconex, atunci sistemul nu va avea de suferit. Un graf este biconex dacă este conex și nu conține nici un nod de articulație. Un astfel de nod reprezintă o vulnerabilitate în sistem, dacă acest nod eșuează atunci rețeaua va fi ruptă în 2. Pentru a determina dacă un nod este articulație vom parcurge următorii pași:

* Vom parcurge toate nodurile și pe fiecare îl vom elimina din graf
* Verificăm dacă graful rămâne conex
* Adăugăm nodul înapoi în graf

Pentru grafurile orientate există și noțiunea de graf tare-conex. Această caracteristică este îndeplinită atunci când există un drum între oricare 2 noduri ale grafului în ambele sensuri. Pentru determinarea acestei caracteristici se poate aplica algoritmul lui Kosaraju:

* Parcurge graful în adâncime și reține nodurile într-o stivă
* Transpune matricea de adiacență
* Scoate pas cu pas noduri din stivă, până aceasta rămâne goală și parcurge iar prin adâncime considerând ca sursă nodul scos anterior. În acest mod se vor afișa componentele tare conexe

Fiind dat un graf conex și neorientat, un arbore parțial al acelui graf este un subgraf care este arbore și conectează toate nodurile. Un graf poate avea mai mulți arbori parțiali. Un arbore parțial cu cost minim este acel arbore parțial care are costul cel mai mic, comparativ cu ceilalți arbori parțiali. Costul unui arbore parțial este dat de suma costurilor fiecărei muchii din acel arbore. Cu ajutorul algoritmului lui Kruskal în varianta Greedy, se poate determina acest gen de arbore. Pașii pentru acest algoritm îi avem mai jos:

* Sortează toate muchiile în ordine descendentă a costurilor lor
* Alege cea muchia cu costul minim. Verifică dacă formează un ciclu cu arborele parțial format până acum. Dacă nu, atunci reține această muchie într-o structură, altfel nu fă nimic
* Repetă pasul 2 până când sunt N-1 muchii în arborele parțial, unde N este numărul total de noduri din graf

Pentru a lucra cu grafurile( parcurgeri sau determinarea de drumuri) va fi nevoie de aplicarea unor algoritmi consacrați. Pentru început vom considera parcurgerile. Acestea sunt parcurgerea în adâncime( Depth-First Search) și parcurgerea în lățime( Breadth-First Search), ele sunt menite să parcurgă arbori.

Căutarea în adâncime se realizează urmărind primul nod disponibil și se parcurge următorul nod disponibil, fiu cu cel anterior, până se ajunge la o frunză( un nod fără copii) sau până se ajunge la un nod țintă. În următoarea fază, se parcurg recursiv nodurile cele mai apropiate care nu au fost vizitate.

Căutarea în lățime se realizează parcurgând toate nodurile copil ale unui nod părinte. Apoi explorează nodurile copil ale copiilor și așa mai departe.

O funcție foarte folosită este cea de determinare a celui mai scurt drum. Pentru a afla acest lucru se pot aplica algoritmii lui Dijkstra sau Floyd-Warshall. Aceștia pot genera cel mai scurt drum dintre oricare 2 noduri din graf. Pentru Dijkstra, se crează o structura unde se reține drumul minim din arborele parțial. Se inițializează structura distanțelor dintre noduri cu o valoare maximă. Sursa va avea valoarea 0. Cât timp structura inițială nu include toate nodurile, se alege un nod care nu este în structură și care are distanța minimă, se include în structură și apoi se actualizează distanțele. Iar pentru Floyd-Warshall se iau toate combinațiile de 3 noduri și se verifica dacă muchiile parțiale fac parte din drumul determinat de cele mai depărtate noduri, dacă da atunci se actualizează distanțele.

Pentru o acoperire întreagă a unui graf se pot aplica algoritmi de determinare a ciclurilor sau drumurilor euleriene sau hamiltoniene. Un drum hamiltonian se definește printr-un drum care vizitează toate nodurile unui graf. Un ciclu hamiltonian este un drum hamiltonian care are o conexiune directă între primul și ultimul nod. Pe cealaltă parte, un drum eulerian este acel drum care vizitează toate muchiile, iar el se poate numi un ciclu eulerian doar dacă se poate ajunge direct de la primul la ultimul nod. Algoritmii aplicați pentru determinarea celor 2 drumuri sunt utili, în special, în determinarea acoperirii sau a nivelului de optimizare ale unui graf. Transmiterea informației prin cât mai multe muchii sau noduri scade eficiența sistemului și crește timpul de transmisie. Un graf este eulerian dacă este conex și dacă are exact 0 noduri cu grad impar, dacă acesta are 2 noduri cu grad impar atunci acesta este semi-eulerian. Gradul unui nod este dat de numărul de conexiuni cu alte noduri. Pentru determinarea unui drum hamiltonian avem următorul algoritm, implementat în maniera backtracking:

* Crează un vector ce reprezintă drumul și adaugă nodul sursă
* Parcurge nodurile rămase și la fiecare verifică dacă este adiacent cu anteriorul din vector și nu este deja adăugat în drumul căutat
* Dacă găsim asemenea nod atunci va fi adăugat în vector
* Dacă nu găsim niciunul atunci graful nu este hamiltonian
  1. **Tehnologia Wi-Fi**

Wi-Fi( Wireless Fidelity) este o tehnologie de transmisie a datelor prin undele radio. Aceasta are o lungime de undă rezervată pentru a evita interferențele. Fiecare aparat care face parte dintr-o rețea Wi-Fi trebuie să aibă un adaptor, care transformă informațiile primite prin radio în informații stocate sub formă de biți pentru a putea fi prelucrate. Odată recepționate de un dispozitiv central( router, switch), datele pot fi transmise mai departe prin cabluri de Ethernet.

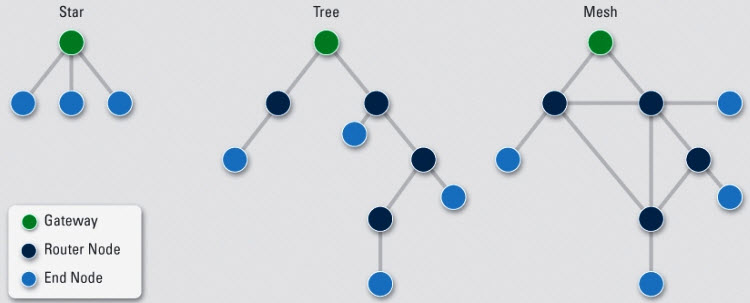
1. **Rețele de senzori wireless**

Pentru început, vom analiza bazele acestui fel de rețea, mai exact senzorii. Un senzor este un aparat, în general electric, care este capabil de a detecta mărimi fizice din mediul înconjurător și să le transforme într-un semnal electric pentru a fi procesate mai departe. Exemple de tipuri de senzori ar fi: de temperatură, de umiditate, acustici, de proximitate, giroscop, accelerometru, etc…

Un nod reprezintă un senzor împreună cu un adaptor de Wi-Fi și un microcontroller pentru a putea stoca și rula un algoritm de comandă și o sursă de energie( în majoritatea cazurilor, o baterie), iar legăturile wireless sunt reprezentate de către muchii. O rețea de senzori wireless constă într-o rețea de senzori( minim 2) care pot comunica informațiile determinate anterior prin intermediul legăturilor fără fir. Reprezentările fizice ale nodurilor diferă în dimensiune, de la câțiva mm2 până la câțiva cm2 ,desigur fiind de preferat dimensiunile mici pentru a influența mediul cât mai puțin. În funcție de poziționările relative ale nodurilor, o rețea poate avea mai multe topologii:

* Stea
* Arbore
* Plasă

Inițial, protocoalele de comunicare în rețea erau simple și necesitau doar comunicarea unidirecțională. Odată cu dezvoltarea rețelelor a apărut și nevoia de comunicare bidirecțională, astfel au apărut topologiile de tip Stea și Plasă.



Iar ca și mediul unde își desfășoară activitatea, acestea pot fi găsite în mediul terestru, subteran, subacvatic, etc.

Pentru o comunicare eficientă între noduri se folosesc seturi de specificații aprobate de IEEE( Institutul inginerilor electrotehniști și electroniști) prin standarde ca și IEEE 802.15.4. Un asemenea protocol este ZigBee.

Utilitatea unei astfel de rețele poate fi găsită în următoarele domenii:

* Uz militar sau supravegherea unei granițe
* Supravegherea persoanelor cu probleme de sănătate
* Monitorizarea condițiilor mediului înconjurător
* Procese industriale
* Agricultură
* Home intelligence

Un aspect de luat în considerare la proiectarea unei rețele de acest gen este securitatea. Comunicarea realizându-se între multe noduri și cu un protocol cunoscut face ca rețeaua să fie vulnerabilă la atacuri sau interferențe.

Cunoscând toate aceste aspecte teoretice putem trece la utilitatea lor practică în cadrul unei rețele de senzori wireless.

1. **Aplicarea teoriei grafurilor în rețele de senzori wireless**

Teoria grafurilor poate fi aplicată pe aceste rețele alături de diferiți algoritmi. Acestea pot reprezenta cu ușurință senzorii și legăturile dintre ei. Structurile de tip graf, sau arborii, alături de folosirea unor algoritmi, ca și căutarea unui nod sau aflarea celui mai scurt drum dintre 2 noduri, facilitează foarte mult comunicarea dintr-o rețea de senzori.

Algoritmii aplicați sunt utili în:

* Localizarea unui nod
* Eficiența energetică
* Acoperire
* Conectivitate
* Controlul traficului
* Detecția de evenimente
* Detecția de erori

Algoritmii prezentați în capitolele anterioare au utilitate în cazurile de mai sus. Localizarea poate fi dată de o simplă parcurgere, iar conectivitatea de către algoritmii de conexitate, tare-conexitate sau biconexitate. Un aspect important este cel al controlului traficului. Această problemă se poate rezolva prin poziționarea unor noduri intermediare de tip router sau switch, care pot asimila informațiile de la câteva noduri și transmite mai departe unde se dorește. O altă metodă este folosirea unui algoritm de rutare a mesajelor în funcție de congestia drumurilor, cum ar fi controlul după feedback, controlul după prioritate sau controlul după echilibru.

În final, pentru eficiența energetică există o multitudine de soluții, spre exemplu, utilizarea unor instrumente low-power, alimentarea dintr-o sursă alternativă din mediul înconjurător, precum energia solară, dar și aplicarea unui algoritm de verificare a nodurilor consumatoare mari de energie și limitarea lor sau dezactivarea/utilizare unui mod low-power a nodurilor care nu au activitate.

1. **Concluzii**

În concluzie, rețelele de senzori wireless sunt unelte ale viitorului, privind din orice perspectivă. Utilitatea lor este variată, iar investitorii le-au prevăzut potențialul. Un mare număr de rețele sunt folosite la această oră, iar exploatarea lor doar va crește. La baza acestora stau algoritmii, fără de care rețelele nu ar avea nici un folos. Acești algoritmi controlează tot, de la localizarea unui nod până la detecția erorilor nodurilor. Performanțele lor devin mai bune, iar optimizările fac ca produsul final să fie unul de calitate și de încredere.

1. **Bibliografie**

<https://www.elprocus.com/introduction-to-wireless-sensor-networks-types-and-applications/>

<https://sci-hub.se/10.1002/9780470396360.ch7>

<https://www.silabs.com/documents/public/white-papers/evolution-of-wireless-sensor-networks.pdf>

<http://www.ijsce.org/wp-content/uploads/papers/v2i2/B0648042212.pdf>

[https://sci-hub.se/https://www.sciencedirect.com/science/article/pii/S156625351830277X](https://sci-hub.se/https:/www.sciencedirect.com/science/article/pii/S156625351830277X)

<https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/>

[https://sci-hub.tw/https://www.sciencedirect.com/science/article/pii/S1474667017499551](https://sci-hub.tw/https:/www.sciencedirect.com/science/article/pii/S1474667017499551)

<https://sci-hub.tw/https://www.sciencedirect.com/science/article/abs/pii/S1389128614001418>