

# Dynamický stav modelu OMNeT++ pomocí SNMP

Jakub Smejkal

Ing. Vladimír Veselý

Ústav informačních systémů - Fakulta informačních technologií - Vysoké učení technické v Brně

Složitost sítí a technologií obecně je stále větší. Proto také vzrůstají nároky na jejich údržbu a problematiku řešení problémů obecně. Z této motivace byl vytvořen nástroj OMNeT++, který představuje prostředek pro tvorbu diskretní simulace. Schopnosti tohoto simulátoru rozšiřuje projekt INET a na něj navazující ANSA, který je vyvíjen na fakultě VUT FIT. Pro korektní průběh simulace je nutné vložit na startu procesu co nejpodrobnější data o simulovaném systému. Na tento aspekt cílí zde popisovaná diplomová práce. Výsledný nástroj je zamýšlen využít k analýze počítačové sítě VUT.

Jelikož záznamy informací o okolí dosahují pouze k nejbližším sousedům je nutné se mezi uzly sítě přesouvat a tím tedy realizovat podstatu procházení grafu. Protože vstupní bod je pouze jeden a naším cílem je analyzovat celý graf, tak se jedná o variantu hledání cest z jednoho uzlu do všech ostatních. Časová složitost prohledávání do šířky je  $O(m+n)$  kde  $m$  značí počet hran grafu a  $n$  počet uzlů. Níže uvedu podobu vytvořené modifikované varianty algoritmu prohledávání do šířky.

## Protokol CDP

Tento zástupce síťových technologií patří do třídy protokolů specializujících se na vyhledávání prvků v topologii a zprostředkování informací o nich. Hlavní činností je tedy odesílání a přijímání informací jak o lokálním zařízení, tak o sousedních. Každý prvek podporující tuto technologii pak provádí sběr dat a ukládá je do MIB, ta je následně přístupná přes SNMP. Protokol je proprietární a není otevřeným standardem. V provozu ho tedy uvidíme pouze na výrobcích Cisco a několika dalších, které mají uzavřené licence s touto společností. Mezi ně patří i firma Hewlett-Packard. Právě z prvků od tohoto výrobce je majoritně složená síť VUT, proto jej lze využít i v této aplikaci. Pro svou činnost protokolu slouží vícesměrová adresa, která je ve tvaru 01-00-0c-cc-cc-cc. Jedná se tedy o technologii pracující na druhé vrstvě. Další charakteristikou je nezávislost na konkrétním síťovém protokolu, proto je použití CDP možné uplatnit na poměrně širokou škálu technologií. Přenos jednotlivých informací je realizován strukturou TLV, která obsahuje atributy. Zkratka představuje trojici typ, délka, hodnota (z anglického Type Length Value).

Version	TTL	Checksum	Type	Length	Value
(1 byte)	(1 byte)	(2 bytes)	(2 bytes)	(2 bytes)	(variable)

## Protokol LLDP

LLDP je oproti CDP průmyslovým standardem označován jako 802.1AB. Princip funkce je velmi podobný variantě od Cisca, tedy opět dochází k odesílání a kolekci informací o jednotlivých zařízeních. Tyto informace jsou uloženy v MIB podle RFC2922. Jedná se také o linkový protokol. Komunikace probíhá pomocí vícesměrové adresy, která má formát 01-80-c2-00-00-00. Zprávy zasílané pomocí LLDP jsou složeny ze sekvence TLV položek. Otevřenost standardu tohoto protokolu pak řeší problém obtížnější správy sítě složené ze zařízení od více výrobců.

Chassis ID TLV	Port ID TLV	TTL TLV	Optional TLVs	End TLV

## Rekonstrukce topologie

Výše zmíněné protokoly lze tedy využít při rekonstrukci topologie. Dále potřebujeme ale také algoritmus na její procházení. Jako předloha pro princip algoritmu bylo použito prohledávání do šířky. Tato varianta se nabízí z druhu činnosti programu, kdy jako vstup slouží prvek v cílové topologii. Od tohoto bodu se pak nástroj dotazuje na okolní zařízení pomocí SNMP technologie.

**Algorithm 2** Modifikované prohledávání do šířky

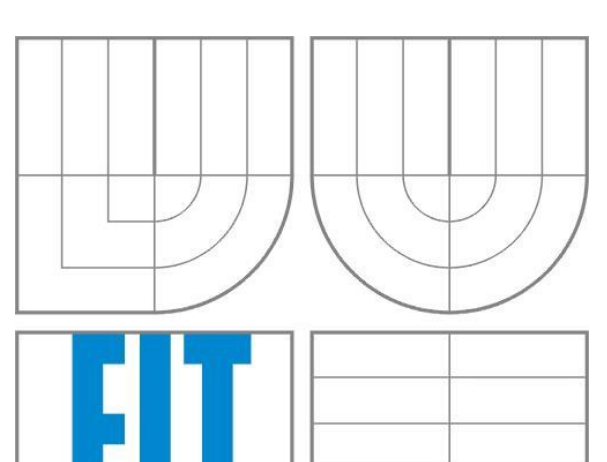
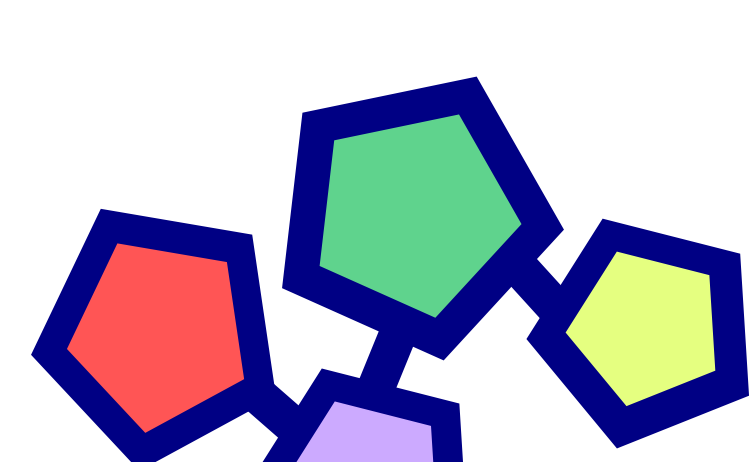
```
function MODBFS(root)
  Adj ← getUID(root)
  Enqueue(Q, root)
  while Q ≠ ∅ do
    u ← Dequeue(Q)
    Adj ← u
    neighbors ← getNeighbors(u)
    for ∀v ∈ neighbors do
      neighbor ← getUID(v)
      if color[neighbor] = Black ∧ neighbor ≠ root then
        Adj[u] ← neighbor
        continue
      end if
      if neighbor ≠ root then
        Enqueue(Q, neighbor)
      end if
      Adj[u] ← neighbor
    end for
    color[u] ← Black
  end while
end function
```

## Dynamická data

V rámci diplomové práce probíhá i načítání dynamických dat z prvků. Aktuálně se soustředíme na směrovací, ARP, MAC tabulku. Kvůli nutnosti data vhodně předat simulátoru vznikla následující XML struktura, která je naplňována při daty získanými při prohledávání jednotlivých prvků v topologii.

```
< Devices >
< Router id = "11.0.0.1" >
  < routingTable >
    < routeEntry Destination = "" Interface = "" Metric = "" NextHop = "" Protocol = "" / >
  </ routingTable >
  < ARPTable >
    < ARPEntry NetAddress = "" PhyAddress = "" Port = "" Type = "" / >
  </ ARPTable >
</ Router >
< Switch id = "11.0.0.2" >
  < MACTable >
    < MACEntry Address = "" Port = "" Status = "" / >
  </ MACTable >
  < ARPTable >
    < ARPEntry NetAddress = "" PhyAddress = "" Port = "" Type = "" / >
  </ ARPTable >
</ Switch >
</ Devices >
```

Jakub Smejkal (2MSK), FIT VUT v Brně (2012), pod vedením Ing. Vladimíra Veselého (UIFS) v rámci projektu ANSA výzkumné skupiny Nes@FIT  
Tato Práce vznikla za podpory projektu MŠMT CZ.1.07/2.3.00/09.0067 TeamIT - Budování konkurenceschopných výzkumných týmu pro IT



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ