Objetivo:

Obter conhecimentos básicos sobre programação socket para desenvolver softwares clientes.

Atividade 1

Estude, compile, corrija (se necessário) e teste o programa hostbyaddr.c. Modifique o programa de modo a poder calcular o tempo necessário à busca de informação sobre uma máquina. Compare o tempo necessário para procurar máquinas com endereços válidos e inválidos. Repita o teste para máquinas de sua sub-rede e para máquinas de outras sub-redes. Explique as diferenças encontradas, se houver.

Uso: ./hostbyaddr_mod Endereco_IP

hostbyaddr

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>

int
main(int argc, const char **argv){

   u_long addr;
   struct hostent *hp;
   char **p;

   if (argc != 2) {
      (void) printf("Uso: %s Endereco_IP\n", argv[0]);
      exit (1);
   }
   if ((int)(addr = inet_addr(argv[1])) == -1) {
      (void) printf("Forma do Endereco_IP: a.b.c.d\n");
      exit (2);
   }
```

```
    hp = gethostbyaddr((char *)&addr, sizeof (addr), AF_INET);
    if (hp == NULL) {
        (void) printf("Não foram encontradas informações sobre o hospedeiro
%s\n", argv[1]);
        exit (3);
    }

    for (p = hp->h_addr_list; *p != 0; p++) {
        struct in_addr in;
        char **q;
        (void) memcpy(&in.s_addr, *p, sizeof (in.s_addr));
        (void) printf("%s\t%s", inet_ntoa(in), hp->h_name);
        for (q = hp->h_aliases; *q != 0; q++)
            (void) printf(" %s", *q);
        (void) putchar('\n');
    }

    exit (0);
}
```

hostbyaddr modificado

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/time.h>               /* for gettimeofday() */
#include <unistd.h>                         /* for gettimeofday()  */
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>

int main(int argc, const char **argv)
```

```
{

   u_long addr;

   struct hostent *hp;
          struct timeval t_inicial, t_final;
          float t_gethostbyaddr;

   char **p;



   if (argc != 2) {

      (void) printf("Uso: %s Endereco_IP\n", argv[0]);

      exit (1);

   }



          // inet_addr transforma uma string de ip (x.x.x.x) em um número
binário

   if ((int)(addr = inet_addr(argv[1])) == -1) {

      (void) printf("Forma do Endereco_IP: a.b.c.d\n");

      exit (2);

   }


          gettimeofday( &t_inicial, NULL );
          // gethostbyaddr(const void *addr, int len, int type);

   hp = gethostbyaddr((char *)&addr, sizeof (addr), AF_INET);
          gettimeofday( &t_final, NULL );

          t_gethostbyaddr = (float)(t_final.tv_sec  - t_inicial.tv_sec);
          t_gethostbyaddr += (t_final.tv_usec -
t_inicial.tv_usec)/(float)1000000;
          printf("Tempo gethostbyaddr(): %.10f sec\n", t_gethostbyaddr);
          fflush(stdout);


   if (hp == NULL) {

      (void) printf("Não foram encontradas informações sobre o hospedeiro
%s\n", argv[1]);

      exit (3);

   }


          /*        struct hostent
                    char    *h_name;        // official name of host
                    char    **h_aliases;    // alias list
                    int     h_addrtype;     // host address type
                    int     h_length;       // length of address
```

```
                char    **h_addr_list;  // list of addresses

        struct in_addr
                unsigned long s_addr;
        */


    for (p = hp->h_addr_list; *p != 0; p++) {

        struct in_addr in;

        char **q;


        (void) memcpy(&in.s_addr, *p, sizeof (in.s_addr));
                (void) printf("%s\t%s", inet_ntoa(in), hp->h_name);


        for (q = hp->h_aliases; *q != 0; q++)

            (void) printf(" %s", *q);

        (void) putchar('\n');

    }


    exit (0);

}
```

O código acima é a modificação do original de modo que possa calcular o tempo gasto para buscar a informação sobre uma máquina na rede.

Acima foi implementado o gettimeofday inicial e final, ao final foi feito o calculo e impresso o resultado.

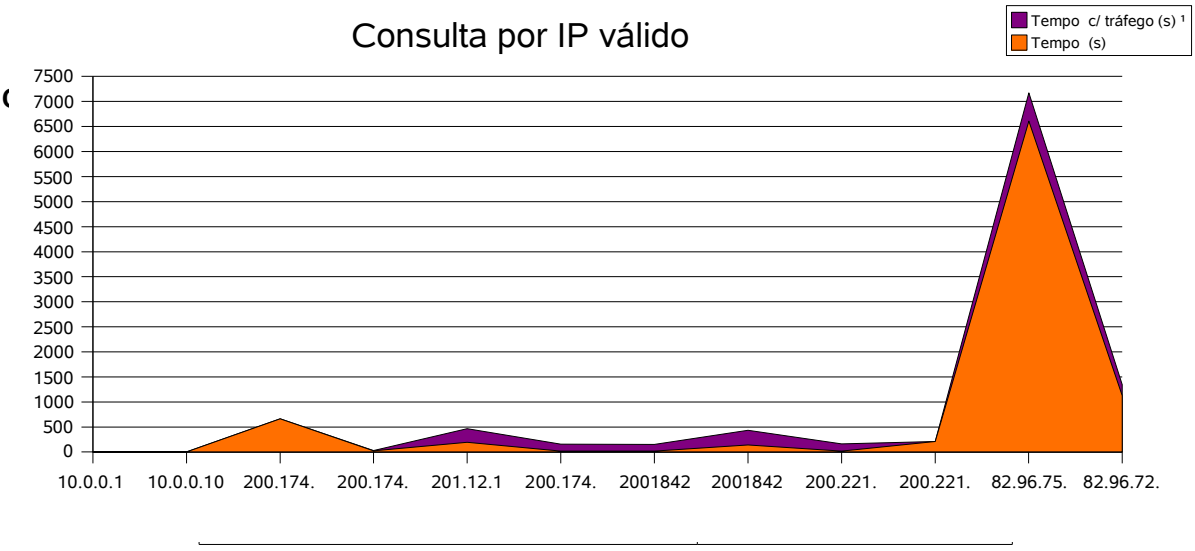
Veja abaixo os dados coletados para consulta em máquinas a rede:

Consulta por IP's válidos:

IP	hostname	Tempo (s)	Tempo c/ tráfego (s) [1]
10.0.0.1 (ip local)	cesarkallas.opensrc cesarkallas	**0.0009660000**	0.0033519999
10.0.0.10 (gateway)	opensrc opensrc	**0.0010380000**	0.0012350000
200.174.144.14 (dns1)	dns1.cps.virtua.com.br	**0.6652169824**	0.0014940000
200.174.144.15 (dns2)	dns2.cps.virtua.com.br	**0.0241839997**	0.0046370002
201.12.128.1	c90c8001.cps.virtua.com.br	**0.1946869940**	0.2693189979
200.174.144.8	1448.cps.virtua.com.br	**0.0213740002**	0.1383080035
200.184.238.149	200-184-238-149.intelignet.com.br	**0.0202709995**	0.1342159957
200.184.254.182	intelig-se1-0-ixrrnp102.intelignet.com.br	**0.1421169937**	0.2952699959
200.221.2.45	home.uol.com.br	**0.0183530003**	0.1459269971
200.221.2.4	domredir.bol.com.br	**0.2094170004**	200.221.2.4

IP	hostname	Tempo (s)	Tempo c/ tráfego (s) [1]
82.96.75.3	rackcheck.com	**6.6051712036**	0.5656419992
82.96.72.1	gw.unitedservers.de	**1.1281609535**	0.2182389945

[1] Tempo medido com o aumento de tráfego na interface local de rede. O aumento foi atráves de uma copia de um grande arquivo entre a estação e o gateway.



Análise dos resultados da atividade 1:

Os resultados obtidos para endereços válidos mostram que quanto menor o caminho percorrido para chegar o host, menor o tempo.

O tempo para a consulta não é sempre o mesmo, independente se é uma rede interna ou se a consulta é feita via internet, isso porque o tráfego de dados dos dispositivos que compõe o caminho até chegar o host influencia. Nos tempos medidos na atividade 1 podemos notar que ao elevar o tráfego de dados na interface local de redes o tempo aumenta.

Outro fator que pode influenciar no tempo de consulta é a distância que o sinal percorre até chegar o host, se o host ficar em outro país, continente, o tempo será bem maior, pois terá que passar por diversos dispositivos (roteadores) até chegar no host, isso é demonstrado nos últimos dois resultados dos endereços válidos.

Com relação aos endereços inválidos, nota-se que o tempo aumenta pelo mesmo fator dos endereços válidos, tráfego de dados e localização fisica (caminhos percorridos), quanto mais distante o host, maior o tempo.

A tabela de endereços inválidos demonstra também que os tempos são maiores quando a consulta é feita para endereços fora da faixa de ip da interface local de rede .

Atividade 2

Construa a *gethostbyname* e imprime a retornada. Compare o uma usando e . Repita o de sub-rede e de outras sub-redes. Explique as encontradas, se houver.

Uso: ./gethostbyname hostname

gethostbyname

```
#include <stdio.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>
#include <sys/time.h>               /* for gettimeofday() */
#include <unistd.h>                         /* for gettimeofday()  */

#include <netinet/in.h>

#include <arpa/inet.h>

#include <netdb.h>



int main(int argc, const char **argv)
{

   u_long addr;

   struct hostent *host;
          struct timeval t_inicial, t_final;
          float t_gethostbyname;

   char **p;



   if (argc != 2) {

      (void) printf("Uso: %s hostname\n", argv[0]);

      exit (1);

   }


          gettimeofday( &t_inicial, NULL );
          // * gethostbyname(const char *name);

   host = gethostbyname(argv[1]);
          gettimeofday( &t_final, NULL );

          t_gethostbyname = (float)(t_final.tv_sec  - t_inicial.tv_sec);
          t_gethostbyname += (t_final.tv_usec -
t_inicial.tv_usec)/(float)1000000;
          printf("Tempo gethostbyname(): %.10f sec\n", t_gethostbyname);
          fflush(stdout);


   if (host == NULL) {

      (void) printf("Não foram encontradas informações sobre o hospedeiro
%s\n", argv[1]);

      exit (3);
```

```
}

        /*      struct hostent
                char    *h_name;        // official name of host
                char    **h_aliases;    // alias list
                int     h_addrtype;     // host address type
                int     h_length;       // length of address
                char    **h_addr_list;  // list of addresses

        struct in_addr
                unsigned long s_addr;

        */

    for (p = host->h_addr_list; *p != 0; p++) {

        struct in_addr in;

        char **q;


        (void) memcpy(&in.s_addr, *p, sizeof (in.s_addr));
                (void) printf("%s\t%s", inet_ntoa(in), host->h_name);


        for (q = host->h_aliases; *q != 0; q++)

            (void) printf(" %s", *q);

        (void) putchar('\n');

    }


    exit (0);

}
```

O código acima exibe o tempo gasto para consultar informações de rede de um determinado host, identificado por nome.

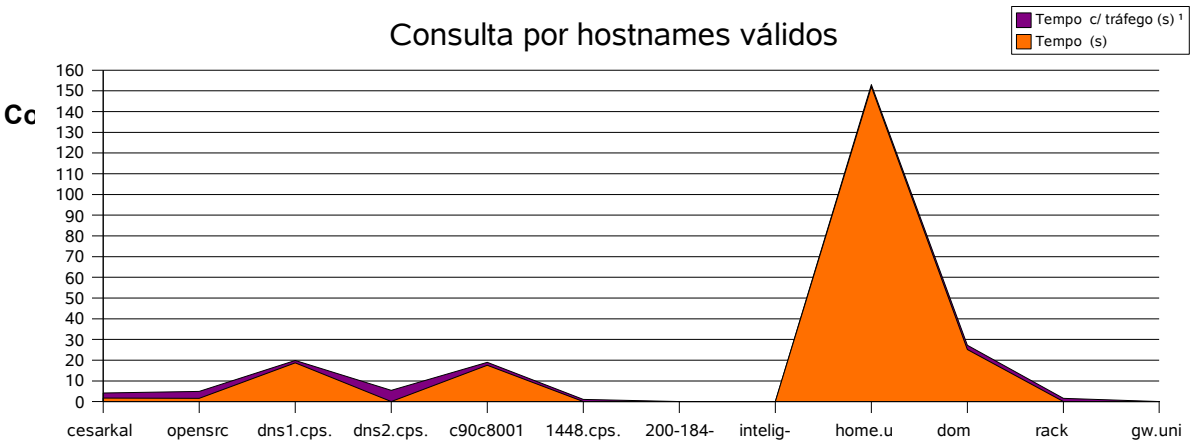
Acima foi implementado o gettimeofday inicial e final, ao final foi feito o calculo e impresso o resultado.

A função gethostbyname(const char *name) está sendo usada no lugar da função gethostbyaddr().

Consulta por hostnames válidos:

IP	hostname	Tempo (s)	Tempo c/ tráfego (s) [1]
10.0.0.1 (ip local)	cesarkallas.opensrc cesarkallas	0.0017830000	0.0023390001
10.0.0.10 (gateway)	opensrc opensrc	0.0015760000	0.0033880000
200.174.144.14 (dns1)	dns1.cps.virtua.com.br	0.0187470000	0.0011150000
200.174.144.15 (dns2)	dns2.cps.virtua.com.br	1.8913689852	0.0054950002
201.12.128.1	c90c8001.cps.virtua.com.br	0.0175219998	0.0014790000
200.174.144.8	1448.cps.virtua.com.br	0.1860249937	0.0011130000
200.184.238.149	200-184-238-149.intelignet.com.br	-	-

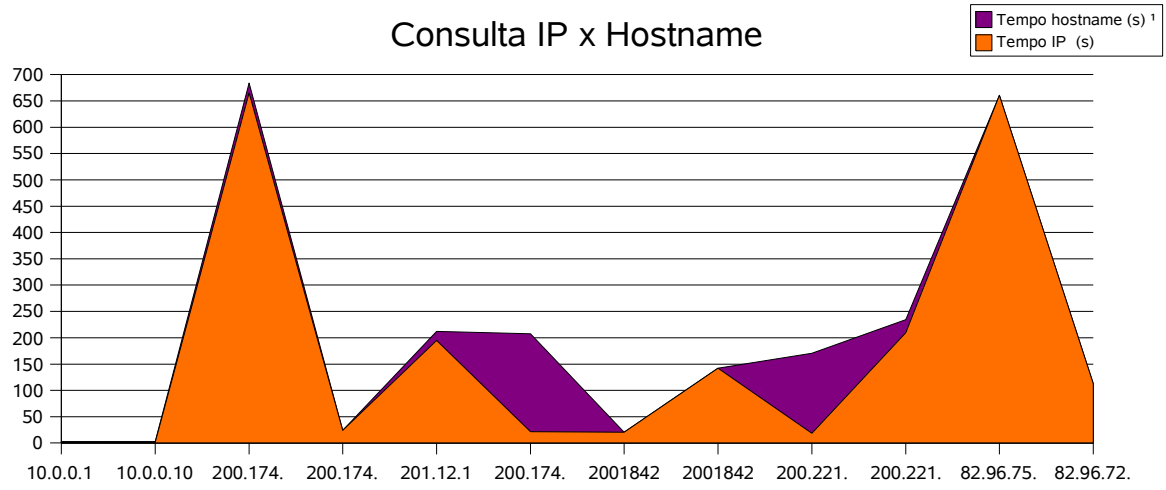
IP	hostname	Tempo (s)	Tempo c/ tráfego (s) [1]
-00.184.254.182	intelig-se1-0-ixrrnp102.intelignet.com.br	-	-
200.221.2.45	home.uol.com.br	0.1519789994	0.0010280000
200.221.2.4	domredir.bol.com.br	0.0251109991	0.0021350000
82.96.75.3	rackcheck.com	4.8386330605	0.0015430000
82.96.72.1	gw.unitedservers.de	-	-



Cc

Analise dos resultados da atividade 2:

O tempo gasto é dependente de muitos fatores, teoricamente o tempo gasto é maior na consulta por hostname, pois o nome precisa ser resolvido e posteriormente consultado.



Conclusão

A atividade mostra claramente que a medição do tempo é muito relativa, dependendo de vários fatores da rede como tráfego e distância física .
Outro fator importante a ser citado é que, na rede interna que foi testado o código, se retirar do arquivo de hosts do linux o nome de um host, torna-se impossível buscar informações por nome da máquina, apenas por IP. Isso de dá pelo fato do que o ambiente testado não possui um servidor de resolução de nome e nem um arquivo pré-configurado com os nomes já estabelecidos.

Ambiente de testes e Sessão

- Processor: Mobile AMD Athlon(tm) XP2500+
- Linux cesarkallas 2.6.11.4-21.8-default #1 Tue Jul 19 12:42:37 UTC 2005 i686 athlon i386 GNU/Linux
- SUSE Linux 9.3
- National Semiconductor Corporation DP83815 (MacPhyter) Ethernet Controller 10/100Mb
- Rede local 10Mbps / Hub encore 10Mbps
- NET Virtua Internet – Cable Internet - 300Kbps