

# Operační systémy Windows kap. 7

## Konfigurace sítě část 3

### Úvod

K zobrazení informací o síti můžete použít řadu příkazů. Tyto nástroje mohou být užitečné i při řešení problémů se sítí.

### Příkaz ping

Příkaz `ping` odešle pakety ICMP na určitou IP adresu v síti a poté vám sdělí, jak dlouho trvalo přenesení dat a získání odpovědi. Je to praktický nástroj, který můžete použít k rychlému testování různých bodů sítě.

#### Poznámka

ICMP (anglicky Internet Control Message Protocol) je v informatice jeden z nejdůležitějších protokolů počítačových sítí založených na rodině protokolů TCP/IP (tedy protokolu, který používá Internet). Protokol ICMP používají operační systémy v síti pro odesílání služebních informací, například chybových zpráv pro oznámení, že požadovaná služba není dostupná nebo že potřebný počítač nebo router není dosažitelný.

ICMP se svým účelem liší od TCP a UDP protokolů tím, že obvykle není používán síťovými aplikacemi přímo, nýbrž je vygenerován na základě nějaké události. Výjimkou je např. nástroj `ping`, který posílá ICMP zprávy „Echo Request“ (a očekává příjem zprávy „Echo Reply“), aby určil, zda je cílový počítač dosažitelný a jak dlouho paketům trvá, než se dostanou k cíli a zpět (tj. měří latenci).

Termín `ping` pochází ze sonarové technologie, která vysílá zvukové impulzy a poté čeká na zpětnou ozvěnu. V počítačové síti je ve většině operačních systémů zabudován nástroj `ping`, který funguje podobně. Zadáte příkaz `ping` spolu s konkrétní adresou URL nebo IP adresou. Počítač odešle několik informačních paketů na dané zařízení a pak čeká na odpověď. Po obdržení odpovědi vám nástroj `ping` zobrazí, jak dlouho trvalo odeslání jednotlivých paketů - nebo vám sdělí, že nepřišla žádná odpověď.

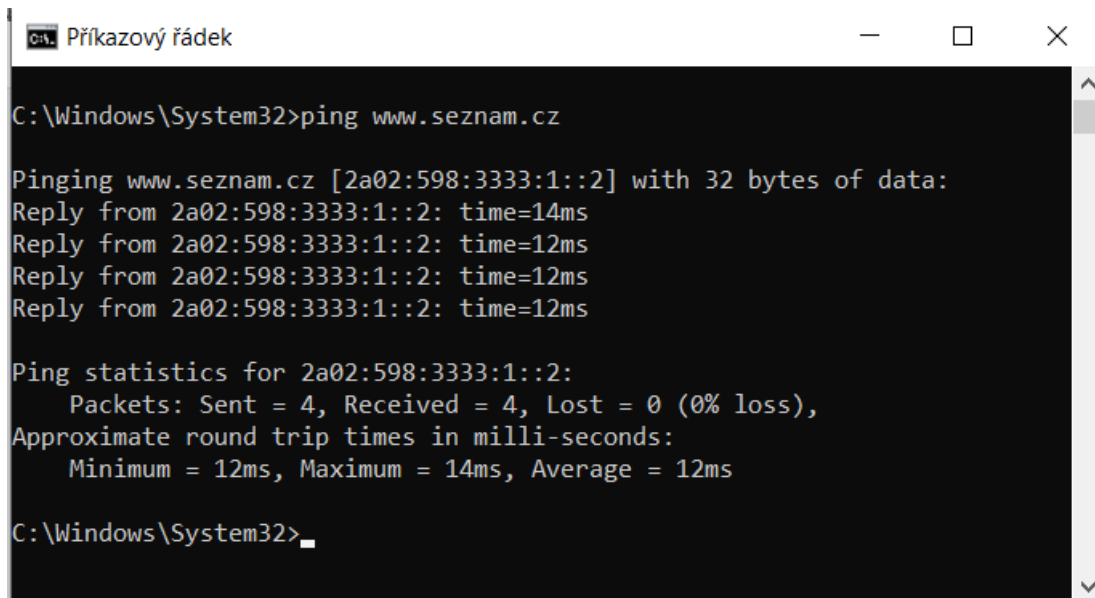
Příkazem `ping` lze zjistit, zda je jiný počítač dosažitelný. Pokud příkaz `ping` dokáže odeslat síťový balíček na jiný počítač a obdrží odpověď, pak byste měli být schopni se k tomuto počítači připojit.

Můžete otestovat, zda se váš počítač může spojit s jiným zařízením - například se směrovačem - v místní síti, nebo zda se může spojit se zařízením na internetu. To vám může pomoci určit, zda je problém se sítí někde v místní síti, nebo někde mimo ni. Doba, za kterou se k vám pakety vrátí, vám pomůže zjistit, zda je připojení pomalé nebo zda dochází ke ztrátě paketů.

V našem příkladu použijeme příkazový řádek systému Windows. Příkaz `ping` však můžete použít i v prostředí Windows PowerShell nebo v aplikaci Terminál v systému MacOS či v libovolné distribuci Linuxu.

#### Použití příkazu ping

V promptu zadejte `ping` spolu s adresou URL nebo IP adresou, kterou chcete pingnout, a stiskněte klávesu Enter. Na obrázku níže pingujeme na adresu `www.seznam.cz` a dostáváme normální odpověď:



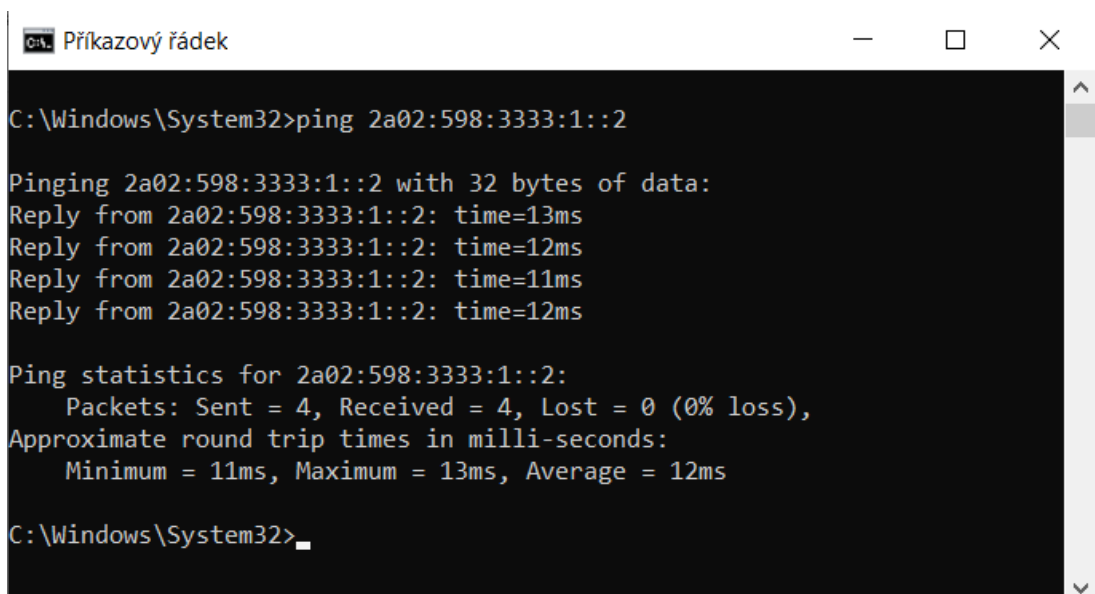
```
C:\Windows\System32>ping www.seznam.cz

Pinging www.seznam.cz [2a02:598:3333:1::2] with 32 bytes of data:
Reply from 2a02:598:3333:1::2: time=14ms
Reply from 2a02:598:3333:1::2: time=12ms
Reply from 2a02:598:3333:1::2: time=12ms
Reply from 2a02:598:3333:1::2: time=12ms

Ping statistics for 2a02:598:3333:1::2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 14ms, Average = 12ms

C:\Windows\System32>
```

Na dalším obrázku pingujeme na adresu IP serveru [www.seznam.cz](http://www.seznam.cz) a dostáváme normální odpověď:



```
C:\Windows\System32>ping 2a02:598:3333:1::2

Pinging 2a02:598:3333:1::2 with 32 bytes of data:
Reply from 2a02:598:3333:1::2: time=13ms
Reply from 2a02:598:3333:1::2: time=12ms
Reply from 2a02:598:3333:1::2: time=11ms
Reply from 2a02:598:3333:1::2: time=12ms

Ping statistics for 2a02:598:3333:1::2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 11ms, Maximum = 13ms, Average = 12ms

C:\Windows\System32>
```

Ve výchozím nastavení příkaz `ping` zašle čtyři síťové packety. Chcete-li nastavit zasílání packetů donekonečna, použijte variantu `-t` viz obrázek níže:

```
C:\Windows\System32>ping 2a02:598:3333:1::2 -t

Pinging 2a02:598:3333:1::2 with 32 bytes of data:
Reply from 2a02:598:3333:1::2: time=14ms
Reply from 2a02:598:3333:1::2: time=13ms
Reply from 2a02:598:3333:1::2: time=12ms
Reply from 2a02:598:3333:1::2: time=13ms
Reply from 2a02:598:3333:1::2: time=13ms
Reply from 2a02:598:3333:1::2: time=11ms
Reply from 2a02:598:3333:1::2: time=12ms
Reply from 2a02:598:3333:1::2: time=13ms
Reply from 2a02:598:3333:1::2: time=13ms
Reply from 2a02:598:3333:1::2: time=13ms
Reply from 2a02:598:3333:1::2: time=13ms
Reply from 2a02:598:3333:1::2: time=12ms
Reply from 2a02:598:3333:1::2: time=12ms
```

Pro ukončení zasílání nekonečné sady packetů stiskněte kombinaci **Ctrl+C**.

Příkaz `ping` umožňuje nastavit variantu síťového protokolu IP (verze 4 nebo 6). Ve výchozím nastavení se použije protokol Ipv6. Chcete-li nastavit zasílání packetů na název `www.seznam.cz` po síťovém protokolu Ipv4 použijte parametr **-4**, viz obr. níže:

```
C:\Windows\System32>ping www.seznam.cz -4

Pinging www.seznam.cz [77.75.75.176] with 32 bytes of data:
Reply from 77.75.75.176: bytes=32 time=11ms TTL=56
Reply from 77.75.75.176: bytes=32 time=16ms TTL=56
Reply from 77.75.75.176: bytes=32 time=12ms TTL=56
Reply from 77.75.75.176: bytes=32 time=12ms TTL=56

Ping statistics for 77.75.75.176:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 11ms, Maximum = 16ms, Average = 12ms

C:\Windows\System32>
```

Chcete-li nastavit zasílání packetů na název `www.seznam.cz` po síťovém protokolu Ipv6 použijte parametr **-6**, viz obr. níže:

```
Příkazový řádek

C:\Windows\System32>ping www.seznam.cz -6

Pinging www.seznam.cz [2a02:598:3333:1::1] with 32 bytes of data:
Reply from 2a02:598:3333:1::1: time=14ms
Reply from 2a02:598:3333:1::1: time=14ms
Reply from 2a02:598:3333:1::1: time=13ms
Reply from 2a02:598:3333:1::1: time=12ms

Ping statistics for 2a02:598:3333:1::1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 12ms, Maximum = 14ms, Average = 13ms

C:\Windows\System32>
```

Parametry lze kombinovat. Chcete-li např. pingovat trvale na [www.seznam.cz](http://www.seznam.cz) po IPv4 zadejte `ping www.seznam.cz -4 -t`:

```
Příkazový řádek - ping www.seznam.cz -4 -t

C:\Windows\System32>ping www.seznam.cz -4 -t

Pinging www.seznam.cz [77.75.74.172] with 32 bytes of data:
Reply from 77.75.74.172: bytes=32 time=15ms TTL=56
Reply from 77.75.74.172: bytes=32 time=12ms TTL=56
Reply from 77.75.74.172: bytes=32 time=12ms TTL=56
Reply from 77.75.74.172: bytes=32 time=12ms TTL=56
Reply from 77.75.74.172: bytes=32 time=12ms TTL=56
Reply from 77.75.74.172: bytes=32 time=12ms TTL=56
Reply from 77.75.74.172: bytes=32 time=12ms TTL=56
Reply from 77.75.74.172: bytes=32 time=12ms TTL=56
Reply from 77.75.74.172: bytes=32 time=13ms TTL=56
Reply from 77.75.74.172: bytes=32 time=11ms TTL=56
```

Odezva (odpověď) síťového zařízení není vždy normální. Na obrázku níže je vidět, jak to může vypadat, když se řada paketů ztratí:

```
C:\Users>ping 192.168.0.1 -n 25

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time=3ms TTL=64
Reply from 192.168.0.1: bytes=32 time=222ms TTL=64
Reply from 192.168.0.1: bytes=32 time=1ms TTL=64
Reply from 192.168.0.1: bytes=32 time=2ms TTL=64
Reply from 192.168.0.1: bytes=32 time=2ms TTL=64
Request timed out.
Request timed out.
Request timed out.
Reply from 192.168.0.1: bytes=32 time=492ms TTL=64
Reply from 192.168.0.1: bytes=32 time=1395ms TTL=64
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 192.168.0.1: bytes=32 time=151ms TTL=64
Reply from 192.168.0.1: bytes=32 time=121ms TTL=64
Reply from 192.168.0.1: bytes=32 time=80ms TTL=64
Reply from 192.168.0.1: bytes=32 time=15ms TTL=64
Reply from 192.168.0.1: bytes=32 time=468ms TTL=64
Reply from 192.168.0.1: bytes=32 time=719ms TTL=64
Reply from 192.168.0.1: bytes=32 time=1154ms TTL=64
Reply from 192.168.0.1: bytes=32 time=424ms TTL=64
Reply from 192.168.0.1: bytes=32 time=2ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 25, Received = 17, Lost = 8 (32% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1395ms, Average = 308ms
```

Jak je vidět na obrázku, naše komunikace je dost špatná. Z 25 odeslaných paketů jich bylo přijato 17 a osm bylo ztraceno, což je 32 % ztrát paketů. I průměrná hodnota RTT (308 ms) v průměru dokazuje, že tato konkrétní např. bezdrátová komunikace je na tom špatně.

## Analýza výsledků

V testu jsme dosáhli 32% ztráty paketů na bezdrátové síti. Bezdrátové sítě jsou velmi zranitelné vůči faktorům, které poškozují pakety putující vzduchem, jako je například vysokofrekvenční rušení, pokrytí signálem nebo slabý signál.

Pokud dochází ke ztrátě paketů, ale neznáte zdroj, zkuste tyto testy provést v různou denní dobu, na různých místech a s různými zařízeními.

### Poznámka

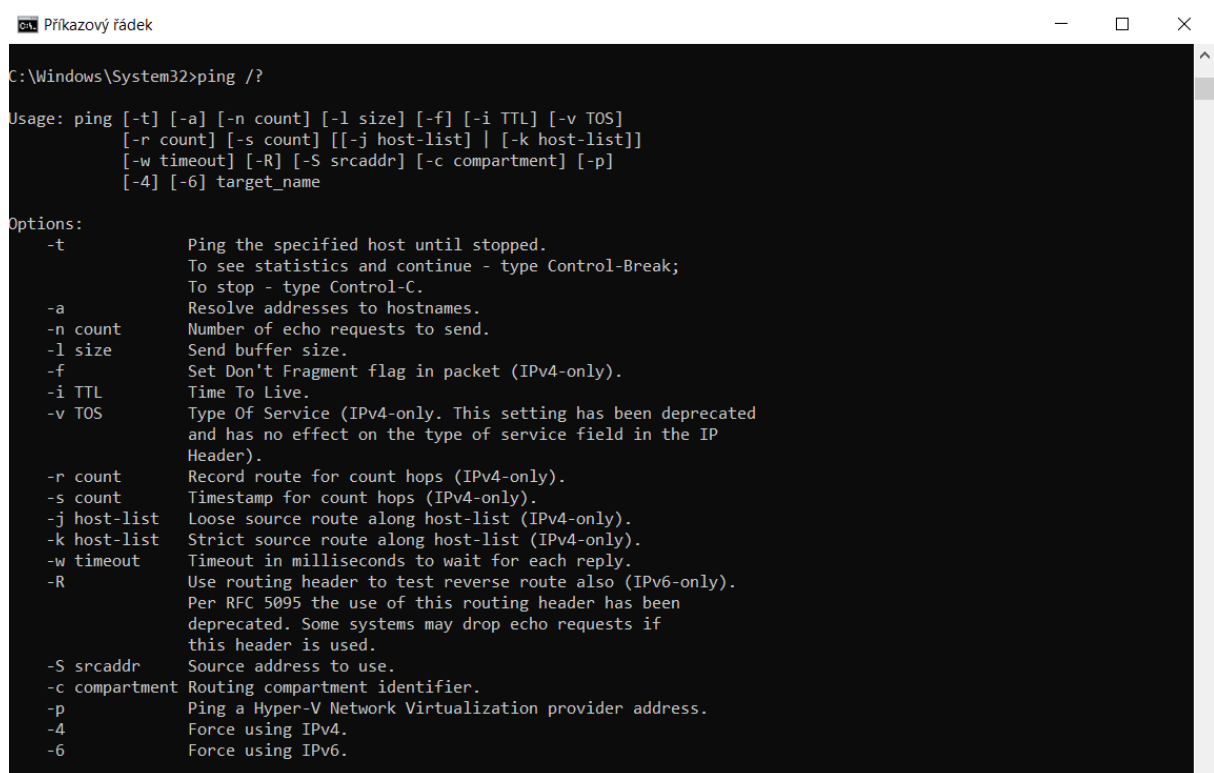
Výsledky následujících testů vám pomohou určit možné příčiny ztráty paketů:

- Pokud notebook přemístíte a zaznamenáte různé % ztráty paketů, může se jednat o problém s bezdrátovým připojením nebo kabeláží (pokud se připojujete k jinému portu Ethernet).
- Pokud dochází k různým ztrátám paketů v různou denní dobu, je možné, že dochází k přetížení sítě.
- Pokud dochází k různé ztrátě paketů u více zařízení, může se jednat o problém s aktualizacemi, médii (špatné kabely nebo bezdrátové připojení) nebo dokonce o poškozený hardware.
- Pokud výsledky ukazují ztrátu paketů při odesílání, pak může jít o problém se síťovým adaptérem.

## Poznámka

Výstup příkazu `ping` lze přesměrovat do výstupního souboru např pomocí `>` v příkazové řádce nebo `|` v powershellu a testovat tak permanentní dostupnost hostitele na síti.

Nápovědu k příkazu `ping` získáme volbou `ping /?` viz obr. níže:



```
Příkazový řádek

C:\Windows\System32>ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
          [-r count] [-s count] [[-j host-list] | [-k host-list]]
          [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
          [-4] [-6] target_name

Options:
  -t           Ping the specified host until stopped.
               To see statistics and continue - type Control-Break;
               To stop - type Control-C.
  -a           Resolve addresses to hostnames.
  -n count     Number of echo requests to send.
  -l size      Send buffer size.
  -f          Set Don't Fragment flag in packet (IPv4-only).
  -i TTL       Time To Live.
  -v TOS       Type Of Service (IPv4-only. This setting has been deprecated
               and has no effect on the type of service field in the IP
               Header).
  -r count     Record route for count hops (IPv4-only).
  -s count     Timestamp for count hops (IPv4-only).
  -j host-list Loose source route along host-list (IPv4-only).
  -k host-list Strict source route along host-list (IPv4-only).
  -w timeout   Timeout in milliseconds to wait for each reply.
  -R          Use routing header to test reverse route also (IPv6-only).
               Per RFC 5095 the use of this routing header has been
               deprecated. Some systems may drop echo requests if
               this header is used.
  -S srcaddr   Source address to use.
  -c compartment Routing compartment identifier.
  -p           Ping a Hyper-V Network Virtualization provider address.
  -4          Force using IPv4.
  -6          Force using IPv6.
```

## Poznámka

TTL znamená "čas do konce života". Je to hodnota na paketu ICMP, která zabraňuje tomu, aby se tento paket šířil mezi hostiteli tam a zpět donekonečna. Každý směrovač, který se paketu dotkne, snižuje TTL. (Pokud není nakonfigurován tak, aby toto nedělal.) Pokud TTL někdy dosáhne nuly, je paket zahozen. Je to také míra toho, kolik skoků paket urazil. Pokud hodnota TTL začínala například na 128 a vy vidíte hodnotu 28, pak mezi systémem, odkud paket pochází, a konečným cílem bylo 100 skoků (hops).

Každý výrobce hardware nebo software nastavuje na svém zařízení počáteční (defaultní) hodnotu TTL. Např. počáteční hodnota TTL pro Windows 10 je 128, počáteční hodnota TTL pro většinu linuxových jader je 255.

## Příkaz tracert

Příkaz `tracert` (zkratka pro "trace route") je nástroj příkazového řádku ve Windows, který slouží k diagnostice síťových problémů a analýze cesty, kterou pakety cestují z vašeho počítače k cílovému serveru nebo zařízení. `tracert` umožňuje zjistit, přes které směrovače (routery) pakety procházejí, což je užitečné pro identifikaci místa, kde může docházet k problémům se sítí.

## 1. Funkce příkazu `tracert`

- **Sledování Cesty Paketu:** `tracert` ukazuje cestu, kterou paket putuje od vašeho počítače k cílové IP adrese nebo doméně.
- **Identifikace Skoků (Hops):** Každý "hop" představuje jeden síťový prvek, typicky router, kterým paket prochází.
- **Diagnostika Síťových Problémů:** Pomáhá identifikovat, kde v síti dochází k zpožděním nebo ztrátám paketů.
- **Zobrazení Latence:** Ukazuje dobu (v milisekundách), kterou paket potřeboval k dosažení každého skoku.

## 2. Parametry příkazu `tracert`

`tracert [parametry] <cíl>`

- `<cíl>`: Může být IP adresa (např. `8.8.8.8`) nebo doménové jméno (např. `google.com`).

### Parametry:

- `-d`
  - Zabraňuje `tracert` v překládání IP adres na názvy hostitelů.
  - Zrychluje proces sledování, protože není třeba provádět DNS dotazy.

Příklad:

```
tracert -d google.com
```

- `-h <max_hops>`
  - Nastaví maximální počet hops (skoků), které `tracert` použije při sledování cesty.
  - Výchozí Hodnota: 30 hops.
  - Omezuje sledování na určitý počet hops, což může být užitečné pro rychlejší diagnostiku.

Příklad:

```
tracert -h 20 google.com
```

- `-w <timeout>`
  - Popis: Nastaví časový limit (v milisekundách) čekání na odpověď od každého routeru.
  - Výchozí Hodnota: 4000 ms (4 sekundy).
  - Použití: Změna timeoutu může být užitečná v sítích s vysokým zpožděním.

Příklad:

```
tracert -w 2000 google.com
```

- `-4` a `-6`
  - Popis: Specifikují, zda `tracert` bude používat IPv4 (-4) nebo IPv6 (-6) protokol.
  - Použití: Umožňuje sledovat cesty v různých IP protokolech.

Příklad:

```
tracert -4 google.com
```

```
tracert -6 google.com
```

### Příklad 1: Sledování cesty k veřejné IP adrese bez překládání názvů hostitelů

**Úkol:** Sledujte cestu k veřejné IP adrese 8.8.8.8 (Google DNS) bez překládání IP adres na názvy hostitelů.

**Řešení:**

```
tracert -d 8.8.8.8
```

- `-d`: Zabraňuje překladům IP adres na názvy hostitelů, což zrychluje proces sledování.
- 8.8.8.8: Cílová IP adresa, kterou chceme sledovat.

**Příklad výstupu:**

```
Tracing route to 8.8.8.8 over a maximum of 30 hops 1 <1 ms <1 ms <1 ms 192.168.1.1 2 5 ms 4 ms 4 ms 10.0.0.1 3 10 ms 9 ms 10 ms 8.8.8.8
Trace complete.
```

- První hop (192.168.1.1): Pravděpodobně váš domácí router.
- Druhý hop (10.0.0.1): Router poskytovatele internetových služeb (ISP).
- Třetí hop (8.8.8.8): Cílový DNS server Google.

### Příklad 2: Sledování cesty k doméně s omezeným počtem hops a specifikovaným timeoutem

**Úkol:** Sledujte cestu k doméně example.com s maximálně 15 hopy a timeoutem 2000 ms.

**Řešení:**

```
tracert -h 15 -w 2000 example.com
```

- `-h 15`: Omezuje sledování na maximálně 15 hops.
- `-w 2000`: Nastaví timeout na 2000 milisekund (2 sekundy) pro každou odpověď.
- example.com: Cílová doména, kterou chceme sledovat.

**Příklad výstupu:**

```
Tracing route to example.com [93.184.216.34] over a maximum of 15 hops
```

1	<1 ms	<1 ms	<1 ms	192.168.1.1
2	8 ms	7 ms	7 ms	10.0.0.1
3	12 ms	11 ms	12 ms	172.16.0.1
4	20 ms	19 ms	19 ms	93.184.216.34

```
Trace complete.
```

- První hop (192.168.1.1): Váš domácí router.
  - Druhý hop (10.0.0.1): Router ISP.
  - Třetí hop (172.16.0.1): Další router v síti ISP nebo mezi poskytovatelem a cílovým serverem.
  - Čtvrtý hop (93.184.216.34): Cílový server example.com.
- 
- Maximální počet hops: Díky parametru `-h 15` se sledování ukončí, pokud cesta k cíli nepřesáhne 15 hops.



- Timeout: parametr -w 2000 zajišťuje, že `tracert` čeká maximálně 2 sekundy na odpověď od každého routeru. Pokud router neodpoví do této doby, zobrazí se hvězdičky (\*).

### Poznámka: Co jsou "Hops"?

**Hops** (česky často označované jako "skoky") představují jednotlivé kroky, které paket provede při své cestě z jednoho zařízení na druhé v síti. Každý hop odpovídá jednomu routeru nebo jinému síťovému zařízení, které paket přesměruje dále směrem k cílové adrese.

Jak `tracert` Používá "Hops"?

Příkaz `tracert` využívá mechanismus **TTL (Time To Live)**, který omezuje životnost paketů v síti. Každý paket má hodnotu TTL, která se s každým přechodem přes router snižuje o 1. Když TTL dosáhne nuly, paket je zahozen a router pošle zpět zprávu **ICMP Time Exceeded**. `tracert` postupně zvyšuje TTL, aby identifikoval každý hop na cestě k cíli.

### **Postup:**

**TTL=1:** Paket dosáhne prvního routeru (první hop), TTL se sníží na 0, paket je zahozen a router odpoví.

**TTL=2:** Paket dosáhne druhého routeru (druhý hop), TTL se sníží na 0, paket je zahozen a router odpoví.

A tak dále, až do cíle nebo maximálního počtu hops.

## Otázky:

1. Vysvětlete, jak příkaz `tracert` funguje a k čemu slouží ve Windows operačním systému.
2. Jaký je rozdíl mezi příkazy `tracert` a `ping` a v jakých situacích byste použili každý z nich?
3. Popište význam parametrů -d, -h a -w v příkazu `tracert` a uveďte situace, ve kterých je vhodné je použít.
4. Jak lze využít příkaz `tracert` k diagnostice síťových problémů? Uveďte konkrétní příklad situace a jak `tracert` pomůže při jejím řešení.
5. Co znamenají jednotlivé sloupce ve výstupu příkazu `tracert` a jak je správně interpretovat při analýze síťové cesty?
6. Jak ovlivňuje hodnota TTL (Time To Live) počet hops při použití příkazu `tracert` a jak se tento mechanismus využívá k mapování cesty paketů?
7. Vysvětlete, jaký je hlavní účel příkazu `ping` ve Windows a jakým způsobem funguje.
8. Popište význam jednotlivých částí výstupu příkazu `ping` a jak je správně interpretovat při analýze síťové konektivity.
9. Jaké jsou různé parametry příkazu `ping` ve Windows a jaký mají vliv na výsledky testu? Uveďte alespoň dva příklady použití s vysvětlením.

## Odkazy:

Vlastní poznámky

<https://www.howtogeek.com/355664/how-to-use-ping-to-test-your-network/>

<https://subinsb.com/default-device-ttl-values/>

<https://www.pcworld.com/how-to-test-packet-loss-on-windows#wbounce-modal>

<https://cs.wikipedia.org/wiki/ICMP>